# Self Service Password Reset 4.3

## Rest Services Specification

**April 2018**

NetIQ.

# Legal Notice

For information about legal notices, trademarks, disclaimers, warranties, export and other use restrictions, U.S. Government rights, patent policy, and FIPS compliance, see https://www.netiq.com/company/legal/.

**Copyright © 2017 and 2018 NetIQ Corporation. All Rights Reserved.**

# Table of Contents

# About this Book

The *NetIQ Self Service Password Reset REST Services Specification* provides conceptual information and step-by-step guidance for extending SSPR services to external needs.

# Intended Audience

This book provides information for individuals responsible for understanding interfacing concepts and extending SSPR services in a secure, RESTful model.

## Systems Administrator

Deploy Self Service Password Reset services across a various systems on a network, and extend Self Service Password Reset to consume RESTful services offered by other systems on the network.

# Other Information in the Library

The library provides the following information resources in addition to this guide:

## Release Notes

Provides information specific to this release of the Self Service Password Reset product, such as known issues.

## Administration Guide

Provides details configuration tasks specific to this release of Self Service Password Reset.

## Videos

Provide supplemental information about using Self Service Password Reset. For more information, see the Micro Focus Self Service Password Reset Playlist.

# About NetIQ Corporation

We are a global, enterprise software company, with a focus on the three persistent challenges in your environment: change, complexity, and risk—and how we can help you control them.

# Our Viewpoint

## Adapting to change and managing complexity and risk are nothing new

In fact, of all the challenges you face, these are perhaps the most prominent variables that deny you the control you need to securely measure, monitor, and manage your physical, virtual, and cloud computing environments.

## Enabling critical business services, better and faster

We believe that providing as much control as possible to IT organizations is the only way to enable timelier and cost effective delivery of services. Persistent pressures like change and complexity will only continue to increase as organizations continue to change and the technologies needed to manage them become inherently more complex.

# Our Philosophy

## Selling intelligent solutions, not just software

In order to provide reliable control, we first make sure we understand the real-world scenarios in which IT organizations like yours operate — day in and day out. That's the only way we can develop practical, intelligent IT solutions that successfully yield proven, measurable results. And that's so much more rewarding than simply selling software.

## Driving your success is our passion

We place your success at the heart of how we do business. From product inception to deployment, we understand that you need IT solutions that work well and integrate seamlessly with your existing investments; you need ongoing support and training post-deployment; and you need someone that is truly easy to work with — for a change. Ultimately, when you succeed, we all succeed.

# Our Solutions

- Identity & Access Governance
- Access Management
- Security Management
- Systems & Application Management
- Workload Management
- Service Management

# Contacting Sales Support

For questions about products, pricing, and capabilities, contact your local partner. If you cannot contact your partner, contact our Sales Support team.

| | |
|---|---|
| **Worldwide:** | *www.netiq.com/about_netiq/officelocations.asp* |
| **United States and Canada:** | *1-888-323-6768* |
| **Email:** | *info@netiq.com* |
| **Website:** | *www.netiq.com* |

# Contacting Technical Support

For specific product issues, contact our Technical Support team.

| | |
|---|---|
| **Worldwide:** | *www.netiq.com/support/contactinfo.asp* |
| **North and South America:** | *1-713-418-5555* |
| **Europe, Middle East, and Africa:** | *+353 (0) 91-782 677* |
| **Email:** | *support@netiq.com* |
| **Website:** | *www.netiq.com/support* |

# Contacting Documentation Support

Our goal is to provide documentation that meets your needs. The documentation for this product is available on the NetIQ website in HTML and PDF formats on a page that does not require you to log in. If you have suggestions for documentation improvements, click Add Comment at the bottom of any page in the HTML version of the documentation posted at www.netiq.com/documentation. You can also email Documentation-Feedback@netiq.com. We value your input and look forward to hearing from you.

# Contacting the Online User Community

NetIQ Communities, the NetIQ online community, is a collaborative network connecting you to your peers and NetIQ experts. By providing more immediate information, useful links to helpful resources, and access to NetIQ experts, NetIQ Communities helps ensure you are mastering the knowledge you need to realize the full potential of IT investments upon which you rely. For more information, visit http://community.netiq.com.

# SSPR 4.3 REST Services Specification

NetIQ's Self-Service Password Reset (SSPR) product interfaces with other systems using various protocols, including LDAP, and HTTP RESTful services.

This document describes the specific HTTP RESTful services used to interface with other systems, including applications that might be developed "in-house" by corporate developers.

SSPR implements two forms of REST APIs:

## Server REST Services

These services are provided to external *client* applications, allowing *client* applications to make RESTful requests to which SSPR will reply.

## Client REST Services

These services are supplied by external *server* applications, allowing SSPR to make RESTful requests to to which external *server* applications will reply.

# SSPR as a REST Service

Settings → Web Services → REST Services

## Introduction

The SSPR service hosts a set of RESTful web APIs to facilitate 3rd party application (SSPR client) development.

## Authentication

All web services are authenticated using basic access authentication utilizing the standard *Authorization* header.

The username portion of the authentication can either be a fully qualified LDAP DN of the user, or a username string value which the application will search for the user.

Additionally, SSPR must be configured in such away to allow web service calls. Not all functions may be enabled. Some operations which involve a third party (other then the authenticated user) may require additional permissions configured within the application.

## Standard JSON Response

All JSON encoded responses are presented using a standard JSON object.

| **Field** | *Type* | Description |
|---|---|---|
| **error** | *boolean* | `false` if the operation was successful |
| **errorCode** | *number (4-digit)* | Application error code |
| **errorMessage** | *UTF-8 string* | Localized error message string |
| **errorDetail** | *UTF-8 string* | Error Number, Error ID and debugging detail message if any, English only |
| **successMessage** | *UTF-8 string* | Localized success message string |
| **data** | *object* | Requested data |

```
{
"error": true,
"errorCode": 5004,
"errorMessage": "Authentication required.",
"errorDetail": "5004 ERROR_AUTHENTICATION_REQUIRED",
"data": {}
}
```

# Configure SSPR to Service REST Calls

By default, SSPR is not configured to respond to REST requests. This service in enabled in SSPR's *Configuration Editor*:

Settings → Web Services → REST Services → Enable Web Services

# SSPR REST Caller Authentication

SSPR requires that REST callers authenticate using the *Basic Authentication Scheme*, where a base-64 encoded username and password *credential* is supplied using the HTTP *Authorization* header. (See RFC 2617 for details) SSPR REST caller authentication credentials are configured using SSPR's *Configuration Editor*:

Settings → Web Services → REST Services

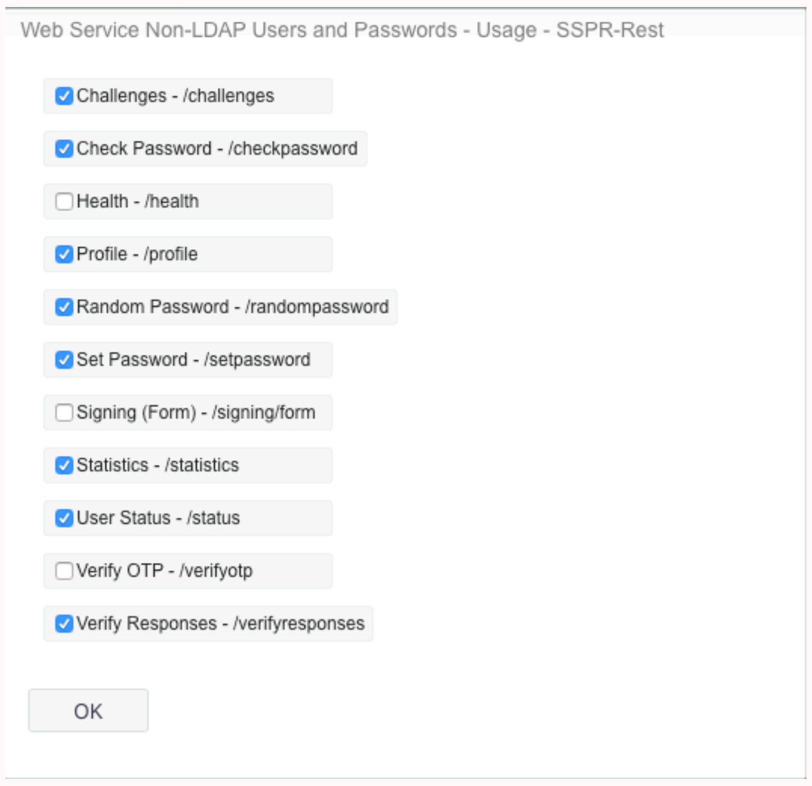
SSPR REST authentication can be configured to use non-LDAP credentials, or LDAP credentials.

## Non-LDAP Authentication Credentials and Permissions

*Non-LDAP* REST credentials are defined solely within SSPR's configuration. Within the SSPR REST context, non-LDAP credentials preempt LDAP credential resolution.

To configure a non-LDAP credential, click *Add Value* in the *Web Service Non-LDAP Users and Passwords* section. Specify a username and a password for the REST credential.

After the user and password have been created, click *Usage* to select the various REST calls that the user may invoke. After saving the SSPR configuration, the REST user account will be able to make the indicated REST calls.



Web Service Non-LDAP Users and Passwords - Usage - SSPR-Rest

☑ Challenges - /challenges

☑ Check Password - /checkpassword

□ Health - /health

☑ Profile - /profile

☑ Random Password - /randompassword

☑ Set Password - /setpassword

□ Signing (Form) - /signing/form

☑ Statistics - /statistics

☑ User Status - /status

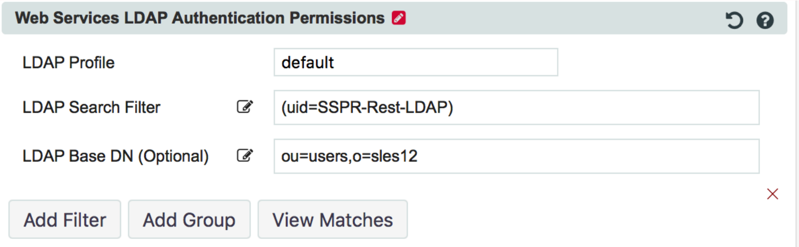
□ Verify OTP - /verifyotp

☑ Verify Responses - /verifyresponses

## LDAP Authentication and Permissions

LDAP credentials for LDAP users and groups are configured using SSPR's *Configuration Editor* in the *Web Services LDAP Authentication Permissions* section of *REST Services*.

For example, to allow a specific LDAP user's credentials to authorize SSPR REST calls, click *Add Filter* and specify an LDAP filter that includes the LDAP user:

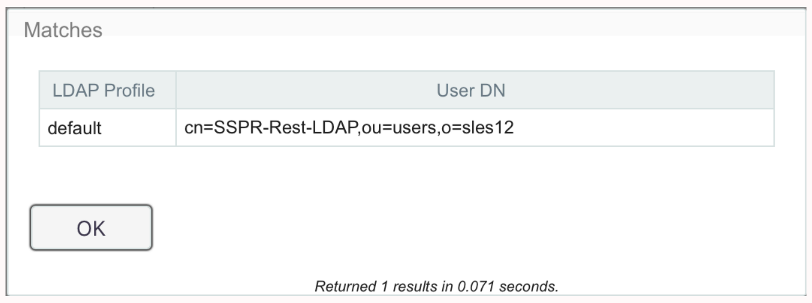

Web Services LDAP Authentication Permissions

LDAP Profile: default

LDAP Search Filter: (uid=SSPR-Rest-LDAP)

LDAP Base DN (Optional): ou=users,o=sles12

The applied filter can be verified by clicking *View Matches*:


Matches

| LDAP Profile | User DN |
|---|---|
| default | cn=SSPR-Rest-LDAP,ou=users,o=sles12 |

Returned 1 results in 0.071 seconds.

### LDAP Third Party Authentication and Permissions

The above section *Web Services LDAP Authentication Permissions* is used to identify LDAP accounts that are allowed to make SSPR REST calls. By default, the users defined in that section cannot target other LDAP accounts using SSPR REST calls. Rather, the REST calls can only target their own LDAP account.

In order for an LDAP REST credential to target other LDAP accounts, the LDAP REST credential must first be defined in the *Web Services LDAP Authentication Permissions*, and then be granted the *Third Party* right explicitly in the *Web Services LDAP Third Party Permissions* section.

For example, to allow a specific LDAP user's credentials to authorize SSPR REST calls that target another LDAP user, click *Add Filter* and specify an LDAP filter that includes the LDAP user (just as was done to define the specific LDAP user in the section above.

## SSPR's LDAP Profile Configured Proxy User

SSPR prefers that a specific *LDAP Proxy User* be configured in each LDAP profile. SSPR uses this LDAP user account to validate LDAP access. Specifically, SSPR implements this user account to validate that SSPR is authorized to modify LDAP user passwords, etc.

LDAP profile configured *proxy users* may not be the target of REST calls. If a REST call attempts to do so, an error is returned:

```
{
  "error": true,
  "errorCode": 7000,
  "errorMessage": "Error_RestInvocationError",
  "errorDetail": "7000 ERROR_REST_INVOCATION_ERROR (rest services can not be invoked against the configure
}
```

# REST Service: /sspr/public/rest/challenges

The default SSPR configuration does not allow reading of user challenge/response data via REST calls. This feature can be enabled in SSPR's *Configuration Editor*:

Settings → Web Services → REST Services → Allow Challenge Service to Read Answers

## GET Method

Retrieve users stored challenges. Location of read responses is determined by the application configuration.

The authenticated REST caller must have the *Challenges* usage privilege in order to successfully invoke this REST interface.

This interface cannot be used to read NMAS stored responses.

### Request Headers

**Authorization** *Basic Authentication Method*

Required. Specifies credentials to authorize the REST caller. This is used by SSPR to validate the caller's rights to perform this action.

**Accept-Language** *Language code*

The request will be processed in the context of the specified language code. For more detail on SSPR locale codes, see Appendix F: Locales (Languages) and Flags.

**Accept** *Document protocol type*

Protocol type values: *application/json*

### Request Query String Parameters

**answers** *true*

Optional. Boolean indicating if answers (in whatever format stored) should be returned in the result. If this parameter is not specified, the value *false* is default.

Requires that SSPR be configured as follows:
Settings→Web Services→REST Services→Allow Challenge Service to Read Answers **= Enabled**

**helpdesk** *true*

Optional. Boolean indicating if helpdesk answers should be returned in the result. If this parameter is not specified, the value *false* is default.

**username** *username or userDN*

Optional. Target Username or userDN of the REST call. If this parameter is not specified, the user specified by the *Authorization* header is assumed.

## Request Content

None

## Response Status Codes

**200 OK** *Successful* The call was successfully serviced by SSPR.

## Response Headers

This list of HTTP response headers is not complete. Headers that are not listed here have little, or no impact for SSPR REST application developers.

**Content-Length** *bytes*

Number of octets (8-bit bytes) found in the *response content*.

**Date** *HTTP-date*

The date and time that the response originated in *HTTP-date* format as defined by RFC 7231 Date/Time Formats.

**Server** *server*

Historically, identifies the server software implemented to respond to the REST request. However, it represents an unnecessary leak of information of information about the platform. Rather than eliminating the header, the response includes this header, but with the innocuous value of *server*.

**Vary** *Accept-Encoding*

Informs cache stores that this response varies based on the value of the *Accept-Encoding* header.

**X-SSPR-Instance** *Name of the SSPR application instance.*

All requests from a specific SSPR server will have this value in common. The value of this header is specified in the SSPR *Configuration Editor*:

Settings→Application→Applications→Instance Name

## Response Content

The content is represented in the json format. The json root object contents include the following child objects:

**error** *Boolean*

Indicates if the REST service experienced an error servicing the request.

**errorCode** *Integer*

If the root *error* object value is *false*, the value of this object will be 0.

If the root *error* object value is *true*, the value of this object will be a numeric SSPR error code. The meaning of the error code can be found in Appendix A: Errors.

Common SSPR *errorCode* Values:

| | | |
|---|---|---|
| **0** | *Success* | Operation completed successfully. |
| **5015** | *ERROR_UNKNOWN* | An error has occurred. If this error occurs repeatedly please contact your help desk. |

**successMessage** *UTF-8 string*

This object is present when the *error* object is *false*. The value of this object is a localized string, generated by SSPR, indicating that the operation has succeeded.

**errorMessage** *UTF-8 string*

This object is present when the *error* object is *true*. The value of this object is a localized string, generated by SSPR, indicating why the operation has failed.

**errorDetail** *UTF-8 string*

This object is present when the *error* object is *true*. The value of this object is a non-localized (English only) string, generated by SSPR, including the *errorCode* and the error *key*.

**data**

This object will be present if the root *error* object value is *false*, and contains the following child objects:

**username** *UTF-8 string*

*user profile* and *userDN* (separated by a pipe character) that is the targeted user of the response.

**policy**

Challenge/response policy associated with the user. See Appendix E: Challange/Response Profile Attributes for field definitions.

**minimumRandoms** *Integer*

This value specifies the number of *random* challenge questions that the user must answer.

When the *challenge/response* services are engaged, challenge questions are presented to the user, to which the user must provide answers. Some of challenge questions are flagged *required*; while the others are flagged as *random*. The user must answer all of the *required* challenge questions.

**challenges** *Object array*

Contains an array of child objects, each child representing one of the user's challenge/responsescredentials.

If the user has not registered any challenge/responses, this field may not be present.

Each child object contains the following objects:

**challengeText** *UTF-8 string*

Text of a challenge question.

**minLength** *Integer*

Text of a challenge question. The minimum length of the user-supplied response string.

**maxLength** *Integer*

The maximum length of the user-supplied response string.

**adminDefined** *Boolean*

Specifies if this challenge question is composed by the SSPR Administrator, of if it is composed by the user.

**required** *Boolean*

The value of this object indicates if the challenge is flagged as *required* or *random*

When the *challenge/response* services are engaged, challenge questions are presented to the user, to which the user must provide answers. Some of challenge questions are flagged *required*; while the others are flagged as *random*. The user must answer all of the *required* challenge questions.

**maxQuestionCharsInAnswer** *Integer*

Maximum number of characters from the challenge question that are permitted in the challenge response.

**enforceWordList** *Boolean*

**true**

The associated challenge response will be rejected if any of the words, in the response, are found on the word-list.

**false**

The associated challenge response will be not be rejected if any of the words, in the response, are found on the word-list.

**answer**

This object contains the following child objects:

**type** *PBKDF2_SHA512*

Encryption method used on stored response.

**answerHash** *128-digit hexadecimal number*

Encrypted response.

**salt** *Random data*

Used to augment the encryption of the response, providing additional random data to defend the *answerHash* against dictionary attacks, etc.

**hashCount** *integer*

The number of hash iterations applied to the *answerHash*.

**caseInsensitive** *Boolean*

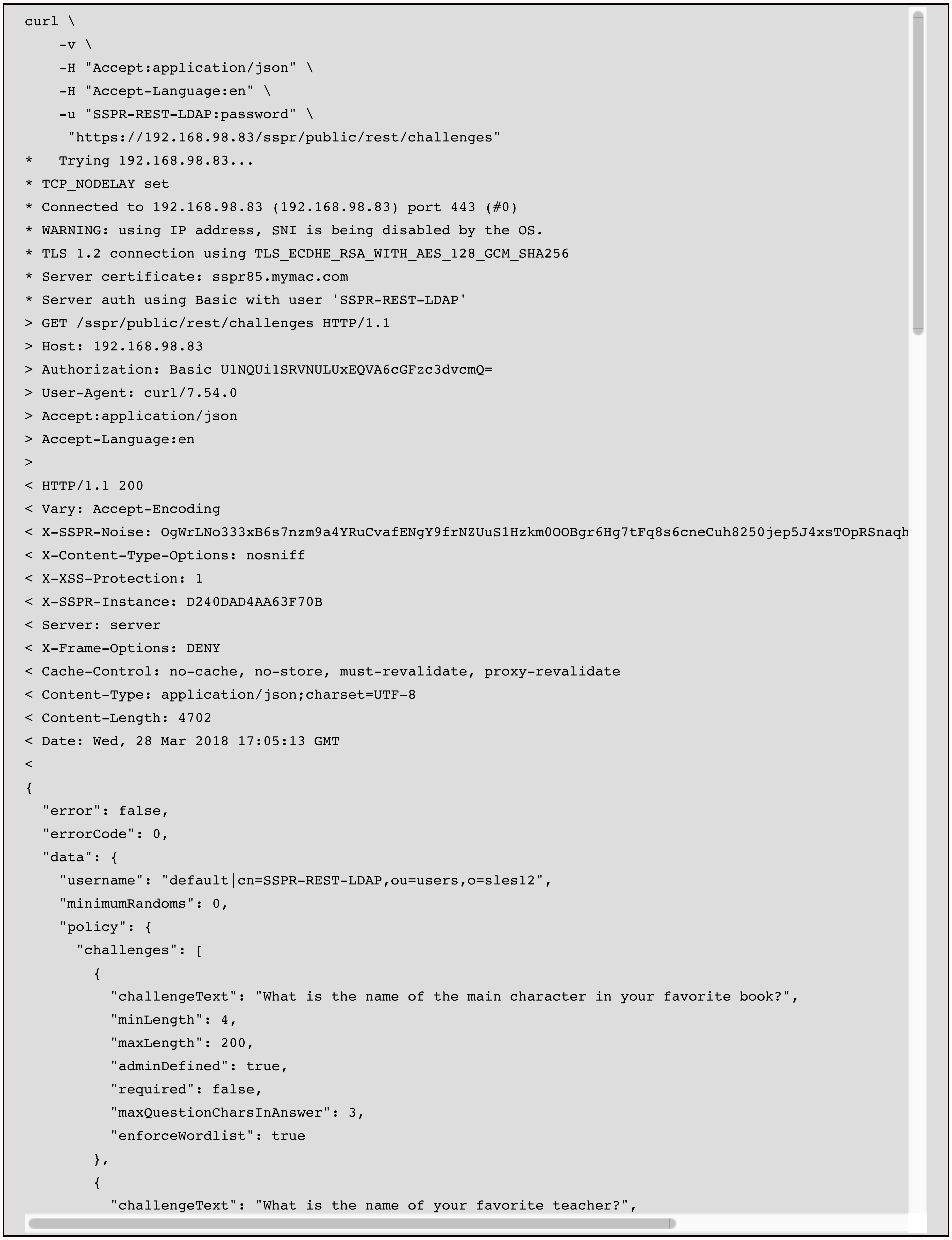
If *true* the response is converted, so that all alphabetical characters are the same case, prior to applying encryption.

## Examples

### Curl Example #1: Parameters in query string.

```
curl \
    -v \
    -H "Accept:application/json" \
    -H "Accept-Language:en" \
    -u "SSPR-REST-EWSS:password" \
     "https://192.168.98.83/sspr/public/rest/challenges?username=sspr-testuser-01&answers=true"
*   Trying 192.168.98.83...
* TCP_NODELAY set
* Connected to 192.168.98.83 (192.168.98.83) port 443 (#0)
* WARNING: using IP address, SNI is being disabled by the OS.
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
* Server certificate: sspr85.mymac.com
* Server auth using Basic with user 'SSPR-REST-EWSS'
> GET /sspr/public/rest/challenges?username=sspr-testuser-01&answers=true HTTP/1.1
> Host: 192.168.98.83
> Authorization: Basic U1NQUi1SRVNULUVXU1M6cGFzc3dvcmQ=
> User-Agent: curl/7.54.0
> Accept:application/json
> Accept-Language:en
>
< HTTP/1.1 200
< Vary: Accept-Encoding
< X-SSPR-Noise: bWGkYu4RV308NBY4rB0bMy5MQPzTNzK3KiLeeO8gAyoqRDN4Jp5nYrJ0a1PeNuvGIxMq4CjZmMKqiOtL6SZvDyow
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1
< X-SSPR-Instance: D240DAD4AA63F70B
< Server: server
< X-Frame-Options: DENY
< Cache-Control: no-cache, no-store, must-revalidate, proxy-revalidate
< Content-Type: application/json;charset=UTF-8
< Content-Length: 4704
< Date: Wed, 28 Mar 2018 15:31:33 GMT
<
{
  "error": false,
  "errorCode": 0,
  "data": {
    "username": "default|cn=sspr-testuser-01,ou=users,o=sles12",
    "minimumRandoms": 0,
    "policy": {
      "challenges": [
        {
          "challengeText": "What is the name of the main character in your favorite book?",
          "minLength": 4,
          "maxLength": 200,
          "adminDefined": true,
          "required": false,
          "maxQuestionCharsInAnswer": 3,
          "enforceWordlist": true
```

**Curl Example #2: No parameters. Target user is the authenticated REST user.**



```
curl \
    -v \
    -H "Accept:application/json" \
    -H "Accept-Language:en" \
    -u "SSPR-REST-LDAP:password" \
     "https://192.168.98.83/sspr/public/rest/challenges"
*   Trying 192.168.98.83...
* TCP_NODELAY set
* Connected to 192.168.98.83 (192.168.98.83) port 443 (#0)
* WARNING: using IP address, SNI is being disabled by the OS.
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
* Server certificate: sspr85.mymac.com
* Server auth using Basic with user 'SSPR-REST-LDAP'
> GET /sspr/public/rest/challenges HTTP/1.1
> Host: 192.168.98.83
> Authorization: Basic U1NQUi1SRVNULUxEQVA6cGFzc3dvcmQ=
> User-Agent: curl/7.54.0
> Accept:application/json
> Accept-Language:en
>
< HTTP/1.1 200
< Vary: Accept-Encoding
< X-SSPR-Noise: OgWrLNo333xB6s7nzm9a4YRuCvafENgY9frNZUuS1Hzkm0OOBgr6Hg7tFq8s6cneCuh8250jep5J4xsTOpRSnaqh
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1
< X-SSPR-Instance: D240DAD4AA63F70B
< Server: server
< X-Frame-Options: DENY
< Cache-Control: no-cache, no-store, must-revalidate, proxy-revalidate
< Content-Type: application/json;charset=UTF-8
< Content-Length: 4702
< Date: Wed, 28 Mar 2018 17:05:13 GMT
<
{
  "error": false,
  "errorCode": 0,
  "data": {
    "username": "default|cn=SSPR-REST-LDAP,ou=users,o=sles12",
    "minimumRandoms": 0,
    "policy": {
      "challenges": [
        {
          "challengeText": "What is the name of the main character in your favorite book?",
          "minLength": 4,
          "maxLength": 200,
          "adminDefined": true,
          "required": false,
          "maxQuestionCharsInAnswer": 3,
          "enforceWordlist": true
        },
        {
          "challengeText": "What is the name of your favorite teacher?",
```

# POST Method

Set users stored challenge/response set.

Parameters may be specified as query string arguments, or as application/json formatted request content.

## Request Headers

**Authorization** *Basic Authentication Method*

Required. Specifies credentials to authorize the REST caller. This is used by SSPR to validate the caller's rights to perform this action.

**Accept-Language** *Language code*

The request will be processed in the context of the specified language code. For more detail on SSPR locale codes, see Appendix F: Locales (Languages) and Flags.

**Accept** *Document protocol type*

Protocol type values: *application/json*

**Content-Type** *Document protocol type*

Protocol type values: *application/json*

## Request Query String Parameters

**answers** *true*

Optional. Boolean indicating if answers (in whatever format stored) should be returned in the result. If this parameter is not specified, the value *false* is default.

Requires that SSPR be configured as follows:

Settings→Web Services→REST Services→Allow Challenge Service to Read Answers **= Enabled**

**helpdesk** *true*

Optional. Boolean indicating if helpdesk answers should be returned in the result. If this parameter is not specified, the value *false* is default.

**username** *username or userDN*

Optional. Target Username or userDN of the REST call. If this parameter is not specified, the user specified by the *Authorization* header is assumed.

**minimumRandoms** *Integer*

This value specifies the number of *random* challenge questions that the user must answer.

When the *challenge/response* services are engaged, challenge questions are presented to the user, to which the user must provide answers. Some of challenge questions are flagged *required*; while the others are flagged as *random*. The user must answer all of the *required* challenge questions.

## Request Content

The content is represented in the json format. The json root object contents may include the following child objects:

**username** *UTF-8 string*

*user profile* and *userDN* (separated by a pipe character) that is the targeted user of the response.

**challenges** *Object array*

Contains an array of child objects, each child representing one of the user's challenge/responsescredentials.

Each child object contains the following objects:

**challengeText** *UTF-8 string*

Text of a challenge question.

**minLength** *Integer*

Text of a challenge question. The minimum length of the user-supplied response string.

**maxLength** *Integer*

The maximum length of the user-supplied response string.

**adminDefined** *Boolean*

**true**

The value of the *challengeText* attribute, or challenge question, is predefined by the SSPR system administrator.

**false**

The value of the *challengeText* attribute, or challenge question, is supplied by the user.

**required** *Boolean*

The value of this object indicates if the challenge is flagged as *required* or *random*.

When the *challenge/response* services are engaged, challenge questions are presented to the user, to which the user must provide answers. Some of challenge questions are flagged *required*; while the others are flagged as *random*. The user must answer all of the *required* challenge questions.

**answer**

This object contains the following child objects:

**answerText** *UTF-8 string*

Text of the response to the *challengeText*.

## Response Status Codes

**200 OK** *Successful* The call was successfully serviced by SSPR.

## Response Headers

This list of HTTP response headers is not complete. Headers that are not listed here have little, or no impact for SSPR REST application developers.

**Content-Type** *application/json;charset=UTF-8*

Indicates the media type of the resource.

**Content-Length** *bytes*

Number of octets (8-bit bytes) found in the *response content*.

**Date** *HTTP-date*

The date and time that the response originated in *HTTP-date* format as defined by RFC 7231 Date/Time Formats.

**Server** *server*

Historically, identifies the server software implemented to respond to the REST request. However, it represents an unnecessary leak of information of information about the platform. Rather than eliminating the header, the response includes this header, but with the innocuous value of *server*.

**Vary** *Accept-Encoding*

Informs cache stores that this response varies based on the value of the *Accept-Encoding* header.

**X-SSPR-Instance** *Name of the SSPR application instance.*

All requests from a specific SSPR server will have this value in common. The value of this header is specified in the SSPR *Configuration Editor*:

Settings→Application→Applications→Instance Name

## Response Content

The content is represented in the json format. The json root object contents may include the following child objects:

**error** *Boolean*

Indicates if the REST service experienced an error servicing the request.

**errorCode** *Integer*

If the root *error* object value is *false*, the value of this object will be 0.

If the root *error* object value is *true*, the value of this object will be a numeric SSPR error code. The meaning of the error code can be found in Appendix A: Errors.

Common SSPR *errorCode* Values:

**0** *Success* Operation completed successfully.

**5015** *ERROR_UNKNOWN* An error has occurred. If this error occurs repeatedly please contact your help desk.

**successMessage** *UTF-8 string*

This object is present when the *error* object is *false*. The value of this object is a localized string, generated by SSPR, indicating that the operation has succeeded.

**errorMessage** *UTF-8 string*

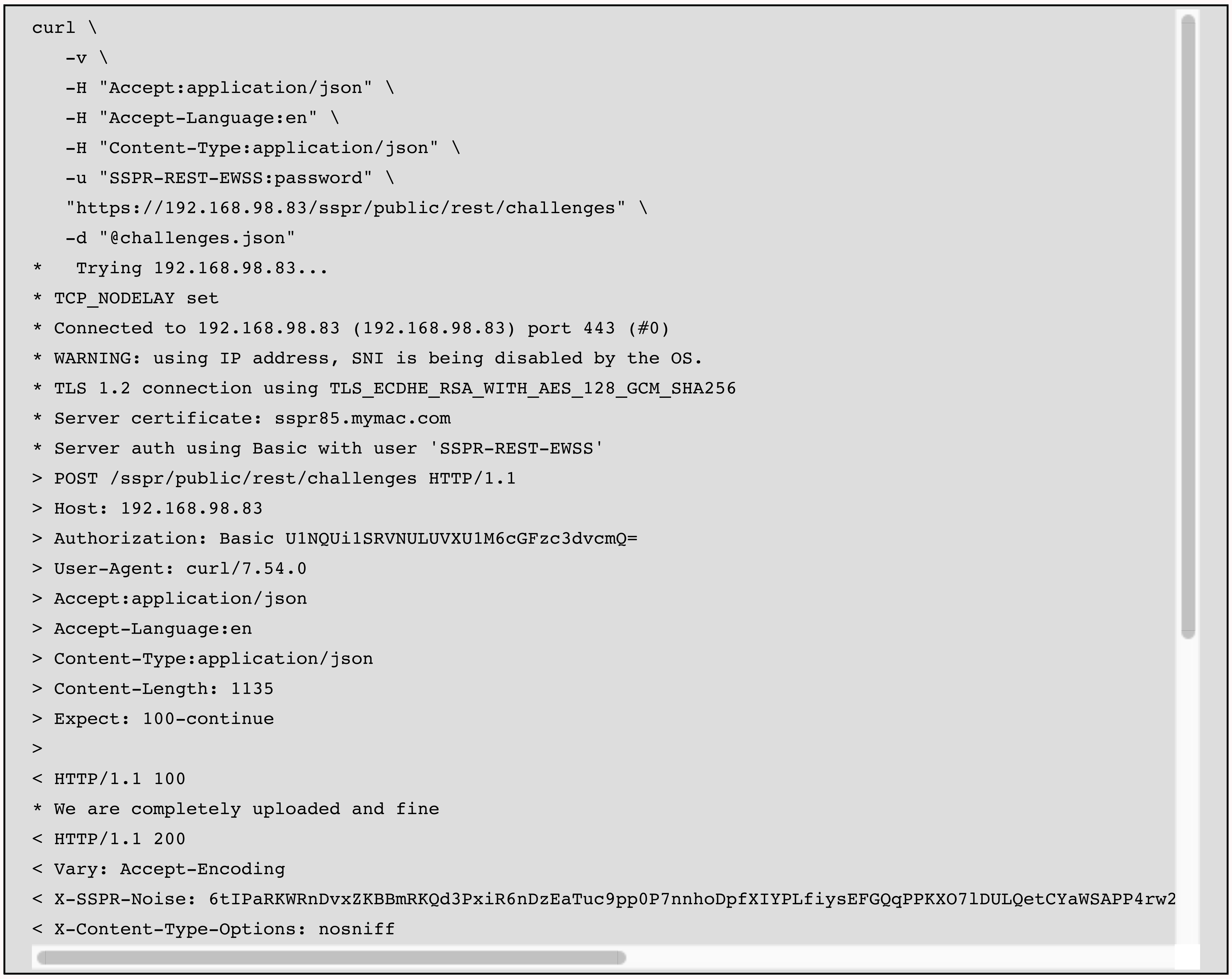
This object is present when the *error* object is *true*. The value of this object is a localized string, generated by SSPR, indicating why the operation has failed.

**errorDetail** *UTF-8 string*

This object is present when the *error* object is *true*. The value of this object is a non-localized (English only) string, generated by SSPR, including the *errorCode* and the error *key*.

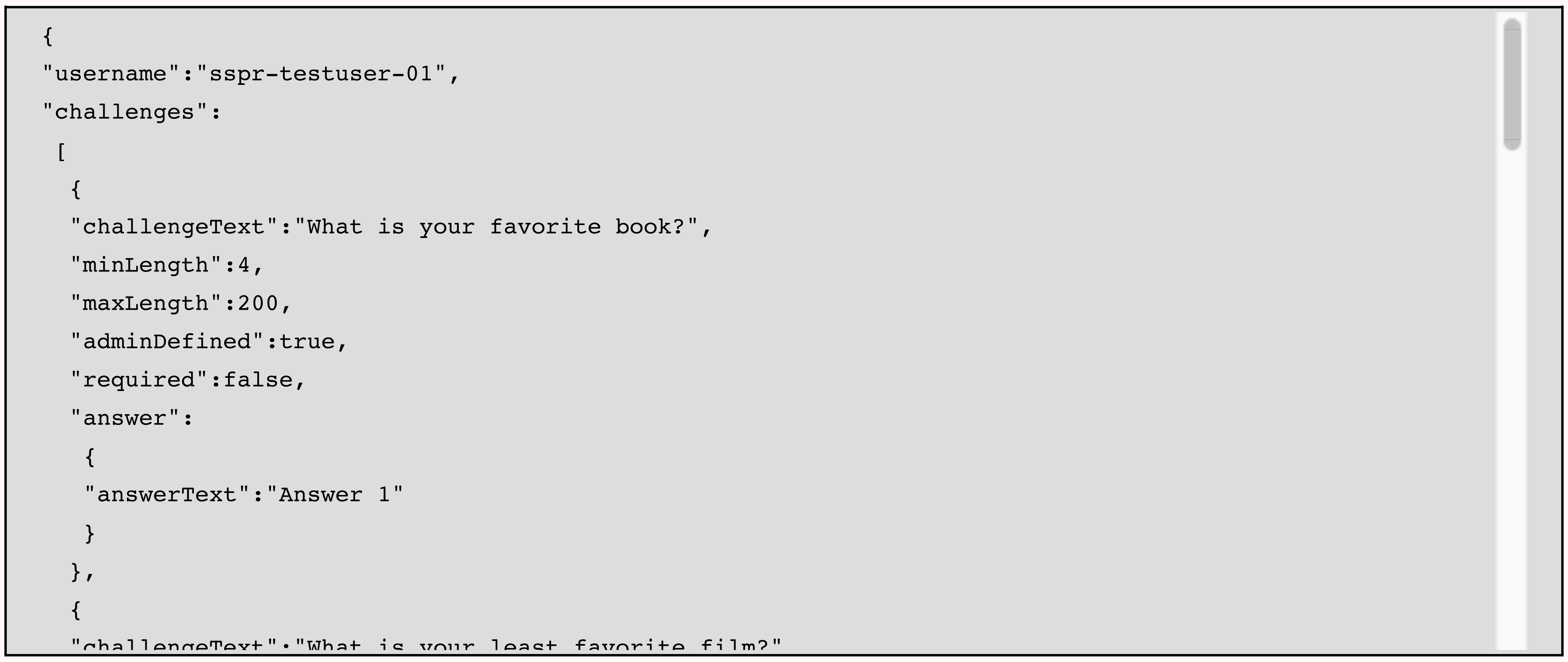
## Examples

### Curl Example #1: All parameters passed via json content.



```
curl \
   -v \
   -H "Accept:application/json" \
   -H "Accept-Language:en" \
   -H "Content-Type:application/json" \
   -u "SSPR-REST-EWSS:password" \
   "https://192.168.98.83/sspr/public/rest/challenges" \
   -d "@challenges.json"
*   Trying 192.168.98.83...
* TCP_NODELAY set
* Connected to 192.168.98.83 (192.168.98.83) port 443 (#0)
* WARNING: using IP address, SNI is being disabled by the OS.
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
* Server certificate: sspr85.mymac.com
* Server auth using Basic with user 'SSPR-REST-EWSS'
> POST /sspr/public/rest/challenges HTTP/1.1
> Host: 192.168.98.83
> Authorization: Basic U1NQUi1SRVNULUVXU1M6cGFzc3dvcmQ=
> User-Agent: curl/7.54.0
> Accept:application/json
> Accept-Language:en
> Content-Type:application/json
> Content-Length: 1135
> Expect: 100-continue
>
< HTTP/1.1 100
* We are completely uploaded and fine
< HTTP/1.1 200
< Vary: Accept-Encoding
< X-SSPR-Noise: 6tIPaRKWRnDvxZKBBmRKQd3PxiR6nDzEaTuc9pp0P7nnhoDpfXIYPLfiysEFGQqPPKXO7lDULQetCYaWSAPP4rw2
< X-Content-Type-Options: nosniff
```

File: challenges.json



```
{
"username":"sspr-testuser-01",
"challenges":
 [
  {
  "challengeText":"What is your favorite book?",
  "minLength":4,
  "maxLength":200,
  "adminDefined":true,
  "required":false,
  "answer":
   {
   "answerText":"Answer 1"
   }
  },
  {
  "challengeText":"What is your least favorite film?"
```

### Curl Example #2: username parameter passed on query string, all others in json content.

* Known Issue: 1087169

```
curl \
   -v \
   -H "Accept:application/json" \
   -H "Accept-Language:en" \
   -H "Content-Type:application/json" \
   -u "SSPR-REST-EWSS:password" \
   "https://192.168.98.83/sspr/public/rest/challenges?username=sspr-testuser-01" \
   -d "@challenges-01.json"
*   Trying 192.168.98.83...
* TCP_NODELAY set
* Connected to 192.168.98.83 (192.168.98.83) port 443 (#0)
* WARNING: using IP address, SNI is being disabled by the OS.
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
* Server certificate: sspr85.mymac.com
* Server auth using Basic with user 'SSPR-REST-EWSS'
> POST /sspr/public/rest/challenges?username=sspr-testuser-01 HTTP/1.1
> Host: 192.168.98.83
> Authorization: Basic U1NQUi1SRVNULUVXU1M6cGFzc3dvcmQ=
> User-Agent: curl/7.54.0
> Accept:application/json
> Accept-Language:en
> Content-Type:application/json
> Content-Length: 1105
> Expect: 100-continue
>
< HTTP/1.1 100
* We are completely uploaded and fine
< HTTP/1.1 200
< Vary: Accept-Encoding
< X-SSPR-Noise: uynVOD34m7cHPVo6NJKcjI1sQApJBteFB0T
< X-Content-Type-Options: nosniff
```

File: challenges-01.json

```
{
"challenges":
 [
  {
  "challengeText":"What is your favorite book?",
  "minLength":4,
  "maxLength":200,
  "adminDefined":true,
  "required":false,
  "answer":
   {
   "answerText":"Answer 1"
   }
  },
  {
  "challengeText":"What is your least favorite film?",
  "minLength":4
```

# DELETE Method

Clear users saved responses.

## Request Headers

**Authorization** *Basic Authentication Method*

Required. Specifies credentials to authorize the REST caller. This is used by SSPR to validate the caller's rights to perform this action.

**Accept-Language** *Language code*

The request will be processed in the context of the specified language code. For more detail on SSPR locale codes, see Appendix F: Locales (Languages) and Flags.

**Accept** *Document protocol type*

Protocol type values: *application/json*

**Content-Type** *Document protocol type*

This header is required even when the HTTP request has no content body. Supported values:

*application/json*
application/x-www-form-urlencoded * Known Issue: 1084033

## Request Query String Parameters

**username** *username or userDN*

Optional. Target Username or userDN of the REST call. If this parameter is not specified, the user specified by the *Authorization* header is assumed.

## Request Content

None

## Response Status Codes

**200 OK** *Successful* The call was successfully serviced by SSPR.

## Response Headers

This list of HTTP response headers is not complete. Headers that are not listed here have little, or no impact for SSPR REST application developers.

**Content-Type** *Indicates the type of the data in the response content body.*

application/json;charset=UTF-8

**Content-Length** *bytes*

Number of octets (8-bit bytes) found in the *response content*.

**Date** *HTTP-date*

The date and time that the response originated in *HTTP-date* format as defined by RFC 7231 Date/Time Formats.

**Server** *server*

Historically, identifies the server software implemented to respond to the REST request. However, it represents an unnecessary leak of information of information about the platform. Rather than eliminating the header, the response includes this header, but with the innocuous value of *server*.

**Vary** *Accept-Encoding*

Informs cache stores that this response varies based on the value of the *Accept-Encoding* header.

**X-SSPR-Instance** *Name of the SSPR application instance.*

All requests from a specific SSPR server will have this value in common. The value of this header is specified in the SSPR *Configuration Editor*:

Settings→Application→Applications→Instance Name

## Response Content

The content is represented in the json format. The json root object contents may include the following child objects:

**error** *Boolean*

Indicates if the REST service experienced an error servicing the request.

**errorCode** *Integer*

If the root *error* object value is *false*, the value of this object will be 0.

If the root *error* object value is *true*, the value of this object will be a numeric SSPR error code. The meaning of the error code can be found in Appendix A: Errors.

Common SSPR *errorCode* Values:

| | | |
|---|---|---|
| **0** | *Success* | Operation completed successfully. |
| **5015** | *ERROR_UNKNOWN* | An error has occurred. If this error occurs repeatedly please contact your help desk. |
| **5016** | *ERROR_CANT_MATCH_USER* | An ldap user for username value 'xxx' was not found. |

**successMessage** *UTF-8 string*

This object is present when the *error* object is *false*. The value of this object is a localized string, generated by SSPR, indicating that the operation has succeeded.

**errorMessage** *UTF-8 string*

This object is present when the *error* object is *true*. The value of this object is a localized string, generated by SSPR, indicating why the operation has failed.

**errorDetail** *UTF-8 string*

This object is present when the *error* object is *true*. The value of this object is a non-localized (English only) string, generated by SSPR, including the *errorCode* and the error *key*.

## Examples

### Curl Example #1: username parameter specified in the query string.

```
curl \
   -v \
   -X DELETE \
   -H "Accept-Language: en" \
   -H "Accept: application/json" \
   -H "Content-Type: application/json" \
   -u "SSPR-REST-EWSS:password" \
   "https://192.168.98.83/sspr/public/rest/challenges?username=sspr-testuser-01"
*   Trying 192.168.98.83...
* TCP_NODELAY set
* Connected to 192.168.98.83 (192.168.98.83) port 443 (#0)
* WARNING: using IP address, SNI is being disabled by the OS.
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
* Server certificate: sspr85.mymac.com
* Server auth using Basic with user 'SSPR-REST-EWSS'
> DELETE /sspr/public/rest/challenges?username=sspr-testuser-01 HTTP/1.1
> Host: 192.168.98.83
> Authorization: Basic U1NQUi1SRVNULUVXU1M6cGFzc3dvcmQ=
> User-Agent: curl/7.54.0
> Accept-Language: en
> Accept: application/json
> Content-Type: application/json
>
< HTTP/1.1 200
< Vary: Accept-Encoding
< X-SSPR-Noise: 1BMgJuY7kgeV4QSinJfybsjIvbccR3qww9cFRFqAK2VRx8pCnxnwyd1NKaJwPyzeTtVS9qqbktQs2vTIbwcI7FzI3m
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1
< X-SSPR-Instance: D240DAD4AA63F70B
< Server: server
< X-Frame-Options: DENY
< Cache-Control: no-cache, no-store, must-revalidate, proxy-revalidate
< Content-Type: application/json;charset=UTF-8
< Content-Length: 109
< Date: Wed, 28 Mar 2018 17:27:59 GMT
<
{
  "error": false,
  "errorCode": 0,
  "successMessage": "The operation has been successfully completed."
}
* Connection #0 to host 192.168.98.83 left intact
```

### Curl Example #2: username parameter specified in body content in json format.

```
curl \
    -v \
    -X DELETE \
    -H "Accept-Language: en" \
    -H "Accept: application/json" \
    -H "Content-Type: application/json" \
    -u "SSPR-REST-EWSS:password" \
    "https://192.168.98.83/sspr/public/rest/challenges" \
    -d "@challenges-del.json"
*   Trying 192.168.98.83...
* TCP_NODELAY set
* Connected to 192.168.98.83 (192.168.98.83) port 443 (#0)
* WARNING: using IP address, SNI is being disabled by the OS.
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
* Server certificate: sspr85.mymac.com
* Server auth using Basic with user 'SSPR-REST-EWSS'
> DELETE /sspr/public/rest/challenges HTTP/1.1
> Host: 192.168.98.83
> Authorization: Basic U1NQUi1SRVNULUVXU1M6cGFzc3dvcmQ=
> User-Agent: curl/7.54.0
> Accept-Language: en
> Accept: application/json
> Content-Type: application/json
> Content-Length: 31
>
* upload completely sent off: 31 out of 31 bytes
< HTTP/1.1 200
< Vary: Accept-Encoding
< X-SSPR-Noise: rras8qNlthSYvfZwef5LF9MsLcR7AqBQ0ehekuYJaBy5SfSrVQ5ATLyHtTcC
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1
< X-SSPR-Instance: D240DAD4AA63F70B
< Server: server
< X-Frame-Options: DENY
< Cache-Control: no-cache, no-store, must-revalidate, proxy-revalidate
< Content-Type: application/json;charset=UTF-8
< Content-Length: 109
< Date: Wed, 28 Mar 2018 17:32:09 GMT
<
{
  "error": false,
  "errorCode": 0,
  "successMessage": "The operation has been successfully completed."
}
* Connection #0 to host 192.168.98.83 left intact
```

### challenges-del.json:

```
{
"username":"sspr-testuser-01"
}
```

### Curl Example #3: No parameters. Target user is the authenticated REST user.

* Known Issue: 1087344

```
curl \
    -v \
    -X DELETE \
    -H "Accept-Language: en" \
    -H "Accept: application/json" \
    -H "Content-Type: application/json" \
    -u "SSPR-REST-LDAP:password" \
    "https://192.168.98.83/sspr/public/rest/challenges"
*   Trying 192.168.98.83...
* TCP_NODELAY set
* Connected to 192.168.98.83 (192.168.98.83) port 443 (#0)
* WARNING: using IP address, SNI is being disabled by the OS.
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
* Server certificate: sspr85.mymac.com
* Server auth using Basic with user 'SSPR-REST-LDAP'
> DELETE /sspr/public/rest/challenges HTTP/1.1
> Host: 192.168.98.83
> Authorization: Basic U1NQUi1SRVNULUxEQVA6cGFzc3dvcmQ=
> User-Agent: curl/7.54.0
> Accept-Language: en
> Accept: application/json
> Content-Type: application/json
>
< HTTP/1.1 200
< Vary: Accept-Encoding
< X-SSPR-Noise: LycqXbyswLzlrJ7c9TcpYUper305HSbUPZfjfIHkBPSFY7Dm3Y2jzRauDrIm9LxFtQJKyrawZ5tzWZBd0ZJvq6LL5I
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1
< X-SSPR-Instance: D240DAD4AA63F70B
< Server: server
< X-Frame-Options: DENY
< Cache-Control: no-cache, no-store, must-revalidate, proxy-revalidate
< Content-Type: application/json;charset=UTF-8
< Content-Length: 201
< Date: Wed, 28 Mar 2018 17:37:34 GMT
<
{
  "error": true,
  "errorCode": 5013,
  "errorMessage": "A required parameter is missing.",
  "errorDetail": "5013 ERROR_MISSING_PARAMETER (username parameter is not specified) fields: [username]"
}
* Connection #0 to host 192.168.98.83 left intact
```

# REST Service: /sspr/public/rest/checkpassword

## POST Method

Check a password value(s) against user policy.

Parameters may be specified as *application/json* or *application/x-www-form-urlencoded* formatted request content, or as query string arguments.

### Request Headers

**Authorization** *Basic Authentication Method*

Required. Specifies credentials to authorize the REST caller. This is used by SSPR to validate the caller's rights to perform this action.

**Accept-Language** *Language code*

The request will be processed in the context of the specified language code. For more detail on SSPR locale codes, see Appendix F: Locales (Languages) and Flags.

**Accept** *Document protocol type*

Protocol type values: *application/json*

**Content-Type** *Document protocol type*

***NOTE:*** *This header is required, even when all parameters are specified in the query string and there is no content following the HTTP headers section.*

Protocol type values:
*application/json*
*application/x-www-form-urlencoded*

### Request Query String Parameters

**password1** *password value*

**password2** *password value confirmation*

**username** *username or userDN*

Optional. Target Username or userDN of the REST call. If this parameter is not specified, the user specified by the *Authorization* header is assumed.

### Request Content

**password1** *password value*

**password2** *password value confirmation*

**username** *username or userDN*

Optional. Target Username or userDN of the REST call. If this parameter is not specified, the user specified by the *Authorization* header is assumed.

## Response Status Codes

**200 OK** *Successful* The call was successfully serviced by SSPR.

## Response Headers

This list of HTTP response headers is not complete. Headers that are not listed here have little, or no impact for SSPR REST application developers.

**Content-Length** *bytes*

Number of octets (8-bit bytes) found in the *response content*.

**Content-Type** *Indicates the type of the data in the response content body.*

application/json;charset=UTF-8

**Date** *HTTP-date*

The date and time that the response originated in *HTTP-date* format as defined by RFC 7231 Date/Time Formats.

**Server** *server*

Historically, identifies the server software implemented to respond to the REST request. However, it represents an unnecessary leak of information of information about the platform. Rather than eliminating the header, the response includes this header, but with the innocuous value of *server*.

**Vary** *Accept-Encoding*

Informs cache stores that this response varies based on the value of the *Accept-Encoding* header.

**X-SSPR-Instance** *Name of the SSPR application instance.*

All requests from a specific SSPR server will have this value in common. The value of this header is specified in the SSPR *Configuration Editor*:

Settings→Application→Applications→Instance Name

## Response Content

The content is represented in the json format. The json root object contents include the following child objects:

**error** *Boolean*

Indicates if the REST service experienced an error servicing the request.

**errorCode** *Integer*

If the root *error* object value is *false*, the value of this object will be 0.

If the root *error* object value is *true*, the value of this object will be a numeric SSPR error code. The meaning of the error code can be found in Appendix A: Errors.

Common SSPR *errorCode* Values:

| | | |
|---|---|---|
| **0** | *Success* | Operation completed successfully. |
| **5015** | *ERROR_UNKNOWN* | An error has occurred. If this error occurs repeatedly please contact your help desk. |

**successMessage** *UTF-8 string*

This object is present when the *error* object is *false*. The value of this object is a localized string, generated by SSPR, indicating that the operation has succeeded.

**errorMessage** *UTF-8 string*

This object is present when the *error* object is *true*. The value of this object is a localized string, generated by SSPR, indicating why the operation has failed.

**errorDetail** *UTF-8 string*

This object is present when the *error* object is *true*. The value of this object is a non-localized (English only) string, generated by SSPR, including the *errorCode* and the error *key*.

**data**

This object will be present if the root *error* object value is *false*, and contains the following child objects:

**version** *integer*

Indicates the version of the parent *data* object. For SSPR 4.3, this value will always be *2*.

**strength** *integer*

Indicates the strength of the password. SSPR judges the password strengths on a strength on a scale of 0 to 100 irrespective of other password policy settings. *Good* is 45 or better, while 70 or better is considered *strong*.

**match** *MATCH*

Indicates that the request parameters *password1* and *password2* match (as strings).

*NO_MATCH*

Indicates that the request parameters *password1* and *password2* do not match (as strings).

**message** *UTF-8 string*

Message indicating success or failure of the (check password) operation. On failure, the value is a hint on the cause of the failure.

**passed** *boolean*

Indicates whether or not the request parameter *password1* passed the user password profile requirements.

**errorCode** *integer*

The value of this object indicates the propriety of the target password. The meaning of the value can be found in Appendix A: Errors.

Common SSPR *errorCode* Values:

**0** *Sanctioned*

The target password meets the minimum required standards.

**4007** *PASSWORD_TOO_SHORT*

New password is too short.

## Examples

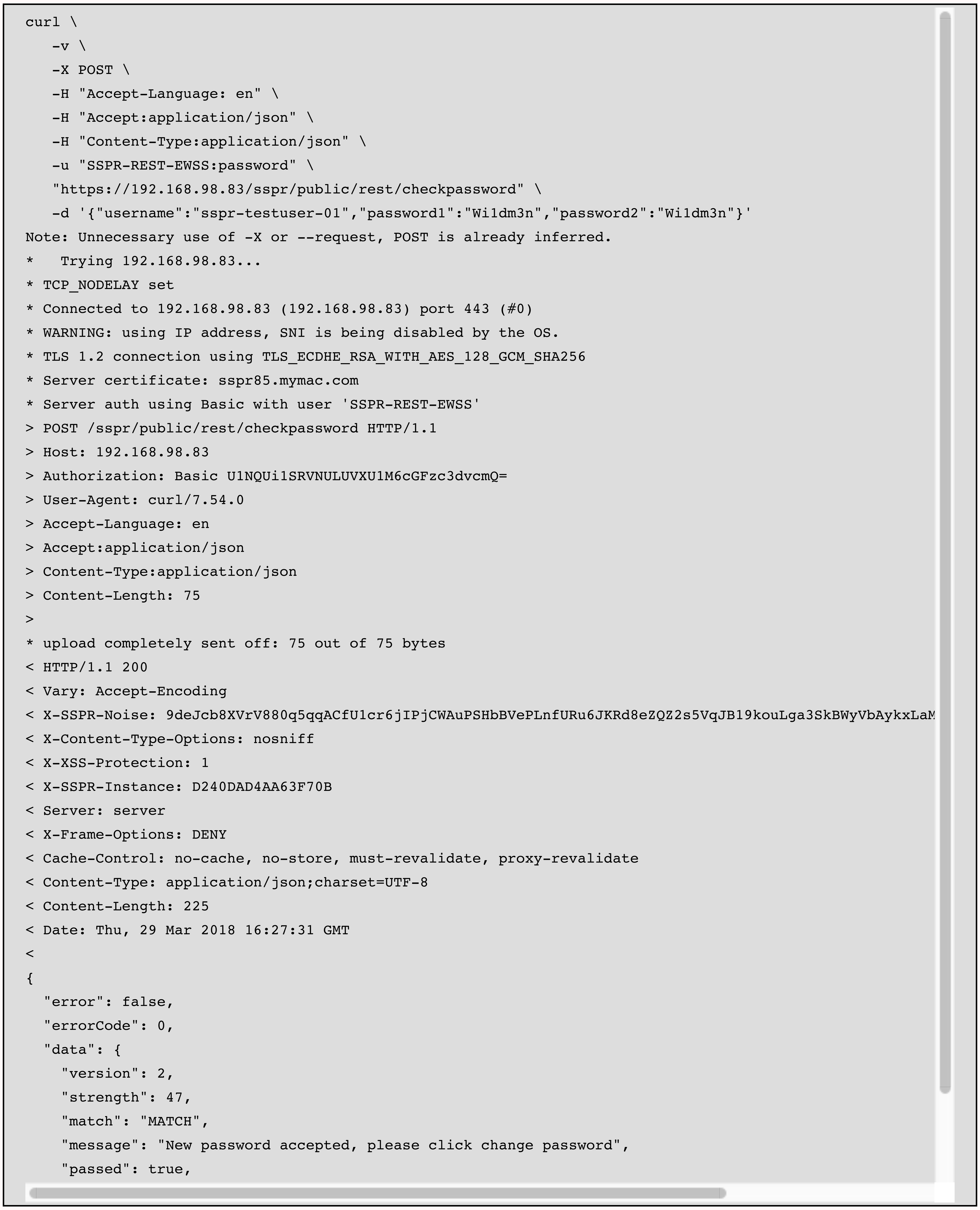
### Curl Example #1

The password and username parameters in content as json data.

```
curl \
   -v \
   -X POST \
   -H "Accept-Language: en" \
   -H "Accept: application/json" \
   -H "Content-Type: application/json" \
   -u "SSPR-REST-EWSS:password" \
   "https://192.168.98.83/sspr/public/rest/checkpassword" \
   -d '{"username":"sspr-testuser-01","password1":"Wi1dm3n","password2":"Wi1dm3n"}'
*   Trying 192.168.98.83...
* TCP_NODELAY set
* Connected to 192.168.98.83 (192.168.98.83) port 443 (#0)
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
* Server certificate: sspr85.mymac.com
* Server auth using Basic with user 'SSPR-REST-EWSS'
> POST /sspr/public/rest/checkpassword HTTP/1.1
> Host: 192.168.98.83
> Authorization: Basic U1NQUi1SRVNULUVXU1M6cGFzc3dvcmQ=
> User-Agent: curl/7.54.0
> Accept-Language: en
> Accept: application/json
> Content-Type: application/json
> Content-Length: 75
>
* upload completely sent off: 75 out of 75 bytes
< HTTP/1.1 200
< Vary: Accept-Encoding
< X-SSPR-Noise: WVNmne3tJqQkLLBI6PKBRyFTx6qhiIjgZVzkZTW1wj0vKX480wIaZBQWyZBVtOhBIQ5aEojzYkjdQR9cW6Pn9SfZ
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1
< X-SSPR-Instance: D240DAD4AA63F70B
< Server: server
< X-Frame-Options: DENY
< Cache-Control: no-cache, no-store, must-revalidate, proxy-revalidate
< Content-Type: application/json;charset=UTF-8
< Content-Length: 225
< Date: Wed, 28 Mar 2018 19:04:07 GMT
<
{
  "error": false,
  "errorCode": 0,
  "data": {
    "version": 2,
    "strength": 47,
    "match": "MATCH",
    "message": "New password accepted, please click change password",
    "passed": true,
    "errorCode": 0
  }
```

### Curl Example #2

The password and username parameters in content as form data.



```
curl \
   -v \
   -X POST \
   -H "Accept-Language: en" \
   -H "Accept:application/json" \
   -H "Content-Type:application/json" \
   -u "SSPR-REST-EWSS:password" \
   "https://192.168.98.83/sspr/public/rest/checkpassword" \
   -d '{"username":"sspr-testuser-01","password1":"Wi1dm3n","password2":"Wi1dm3n"}'
Note: Unnecessary use of -X or --request, POST is already inferred.
*   Trying 192.168.98.83...
* TCP_NODELAY set
* Connected to 192.168.98.83 (192.168.98.83) port 443 (#0)
* WARNING: using IP address, SNI is being disabled by the OS.
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
* Server certificate: sspr85.mymac.com
* Server auth using Basic with user 'SSPR-REST-EWSS'
> POST /sspr/public/rest/checkpassword HTTP/1.1
> Host: 192.168.98.83
> Authorization: Basic U1NQUi1SRVNULUVXU1M6cGFzc3dvcmQ=
> User-Agent: curl/7.54.0
> Accept-Language: en
> Accept:application/json
> Content-Type:application/json
> Content-Length: 75
>
* upload completely sent off: 75 out of 75 bytes
< HTTP/1.1 200
< Vary: Accept-Encoding
< X-SSPR-Noise: 9deJcb8XVrV880q5qqACfU1cr6jIPjCWAuPSHbBVePLnfURu6JKRd8eZQZ2s5VqJB19kouLga3SkBWyVbAykxLaM
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1
< X-SSPR-Instance: D240DAD4AA63F70B
< Server: server
< X-Frame-Options: DENY
< Cache-Control: no-cache, no-store, must-revalidate, proxy-revalidate
< Content-Type: application/json;charset=UTF-8
< Content-Length: 225
< Date: Thu, 29 Mar 2018 16:27:31 GMT
<
{
  "error": false,
  "errorCode": 0,
  "data": {
    "version": 2,
    "strength": 47,
    "match": "MATCH",
    "message": "New password accepted, please click change password",
    "passed": true,
```

### Curl Example #3

Parameters in the query string.

```
curl \
   -v \
   -X POST \
   -H "Accept-Language: en" \
   -H "Accept:application/json" \
   -H "Content-Type:application/x-www-form-urlencoded" \
   -u "SSPR-REST-EWSS:password" \
   "https://192.168.98.83/sspr/public/rest/checkpassword?username=sspr-testuser-01&password1=alowBuff&pass
* TCP_NODELAY set
* Connected to 192.168.98.83 (192.168.98.83) port 443 (#0)
* WARNING: using IP address, SNI is being disabled by the OS.
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
* Server certificate: sspr85.mymac.com
* Server auth using Basic with user 'SSPR-REST-EWSS'
> POST /sspr/public/rest/checkpassword?username=sspr-testuser-01&password1=alowBuff&password2=alowBuff HTT
> Host: 192.168.98.83
> Authorization: Basic U1NQUi1SRVNULUVXU1M6cGFzc3dvcmQ=
> User-Agent: curl/7.54.0
> Accept-Language: en
> Accept:application/json
> Content-Type:application/x-www-form-urlencoded
>
< HTTP/1.1 200
< Vary: Accept-Encoding
< X-SSPR-Noise: Va0rCUgcbMIfw6djv9Imk6rcG1Ca9sKJXbk1OhYCPNbqDhfb9jeLnBSv2RyJa
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1
< X-SSPR-Instance: D240DAD4AA63F70B
< Server: server
< X-Frame-Options: DENY
< Cache-Control: no-cache, no-store, must-revalidate, proxy-revalidate
< Content-Type: application/json;charset=UTF-8
< Content-Length: 225
< Date: Thu, 29 Mar 2018 16:32:55 GMT
<
{
  "error": false,
  "errorCode": 0,
  "data": {
    "version": 2,
    "strength": 21,
    "match": "MATCH",
    "message": "New password accepted, please click change password",
    "passed": true,
    "errorCode": 0
  }
}
* Connection #0 to host 192.168.98.83 left intact
```

# REST Service: /sspr/public/rest/health

**NOTE:** The default SSPR configuration does not enable the health service for REST calls. This feature can be enabled in SSPR's *Configuration Editor*:

Settings → Web Services → REST Services → Enable Public Health and Statistics Web Services

## GET Method

Read SSPR's health metrics.

### Request Headers

**Authorization** *Basic Authentication Method*

Required. Specifies credentials to authorize the REST caller. This is used by SSPR to validate the caller's rights to perform this action.

**Accept-Language** *Language code*

The request will be processed in the context of the specified language code. For more detail on SSPR locale codes, see Appendix F: Locales (Languages) and Flags.

**Accept** *Document protocol type*

Protocol type values:
*application/json*
*text/plain*

### Request Query String Parameters

**refreshImmediate** *true*

Optional. Indicates if the server should refresh the health status before calling this service. If this parameter is not specified, a default value of *false* is used. Specifying *true* for this parameter requires that the REST client (caller) have administrative rights.

### Request Content

None

### Response Status Codes

**200 OK** *Successful* The call was successfully serviced by SSPR.

## Response Headers

This list of HTTP response headers is not complete. Headers that are not listed here have little, or no impact for SSPR REST application developers.

**Content-Length** *bytes*

Number of octets (8-bit bytes) found in the *response content*.

**Date** *HTTP-date*

The date and time that the response originated in *HTTP-date* format as defined by RFC 7231 Date/Time Formats.

**Server** *server*

Historically, identifies the server software implemented to respond to the REST request. However, it represents an unnecessary leak of information of information about the platform. Rather than eliminating the header, the response includes this header, but with the innocuous value of *server*.

**Vary** *Accept-Encoding*

Informs cache stores that this response varies based on the value of the *Accept-Encoding* header.

**X-SSPR-Instance** *Name of the SSPR application instance.*

All requests from a specific SSPR server will have this value in common. The value of this header is specified in the SSPR *Configuration Editor*:

Settings→Application→Applications→Instance Name

## Response Content

The response content is dictated by the value of the request *Accept* header.

For **Accept: text/plain**, the response content will be a one-word string that indicates the most severe health status occurring in any of SSPR's *health monitored* components. Values include (from least to most severe): *DEBUG*, *INFO*, *GOOD*, *CONFIG*, *CAUTION* & *WARN*.

For **Accept: application/json**, the content will be a json formatted string. The json root object contents include the following child objects:

**error** *Boolean*

Indicates if the REST service experienced an error servicing the request.

**errorCode** *Integer*

If the root *error* object value is *false*, the value of this object will be 0.

If the root *error* object value is *true*, the value of this object will be a numeric SSPR error code. The meaning of the error code can be found in Appendix A: Errors.

Common SSPR *errorCode* Values:

| | | |
|---|---|---|
| **0** | *Success* | Operation completed successfully. |
| **5015** | *ERROR_UNKNOWN* | An error has occurred. If this error occurs repeatedly please contact your help desk. |

**successMessage** *UTF-8 string*

This object is present when the *error* object is *false*. The value of this object is a localized string, generated by SSPR, indicating that the operation has succeeded.

**errorMessage** *UTF-8 string*

This object is present when the *error* object is *true*. The value of this object is a localized string, generated by SSPR, indicating why the operation has failed.

**errorDetail** *UTF-8 string*

This object is present when the *error* object is *true*. The value of this object is a non-localized (English only) string, generated by SSPR, including the *errorCode* and the error *key*.

**data**

This object will be present if the root *error* object value is *false*, and contains the following child objects:

**timestamp** *ISO 8601, Combined date and time, Coordinated Universal Time*

Indicates when the parent data object was compiled.

**overall** *string*

Indicates the most severe health status occurring in any of SSPR's *health monitored* components. Values include (from least to most severe): *DEBUG*, *INFO*, *GOOD*, *CONFIG*, *CAUTION*, *WARN*.

**records** *Object array*

Contains an array of child objects, each child representing one component of the SSPR application's health.

Each child object contains the following objects:

**status** *string*

Health status this component of the SSPR application. Values include (from least to most severe): *DEBUG*, *INFO*, *GOOD*, *CONFIG*, *CAUTION*, *WARN*.

**topic** *string*

Name of this health component of the SSPR application. Values include: *Appliance*, *Application*, *Configuration*, *LDAP*, *Email*, *Integrity*, *TokenService*, *Platform*, *LocalDB*, *SMS*, *Database* & *Audit*.

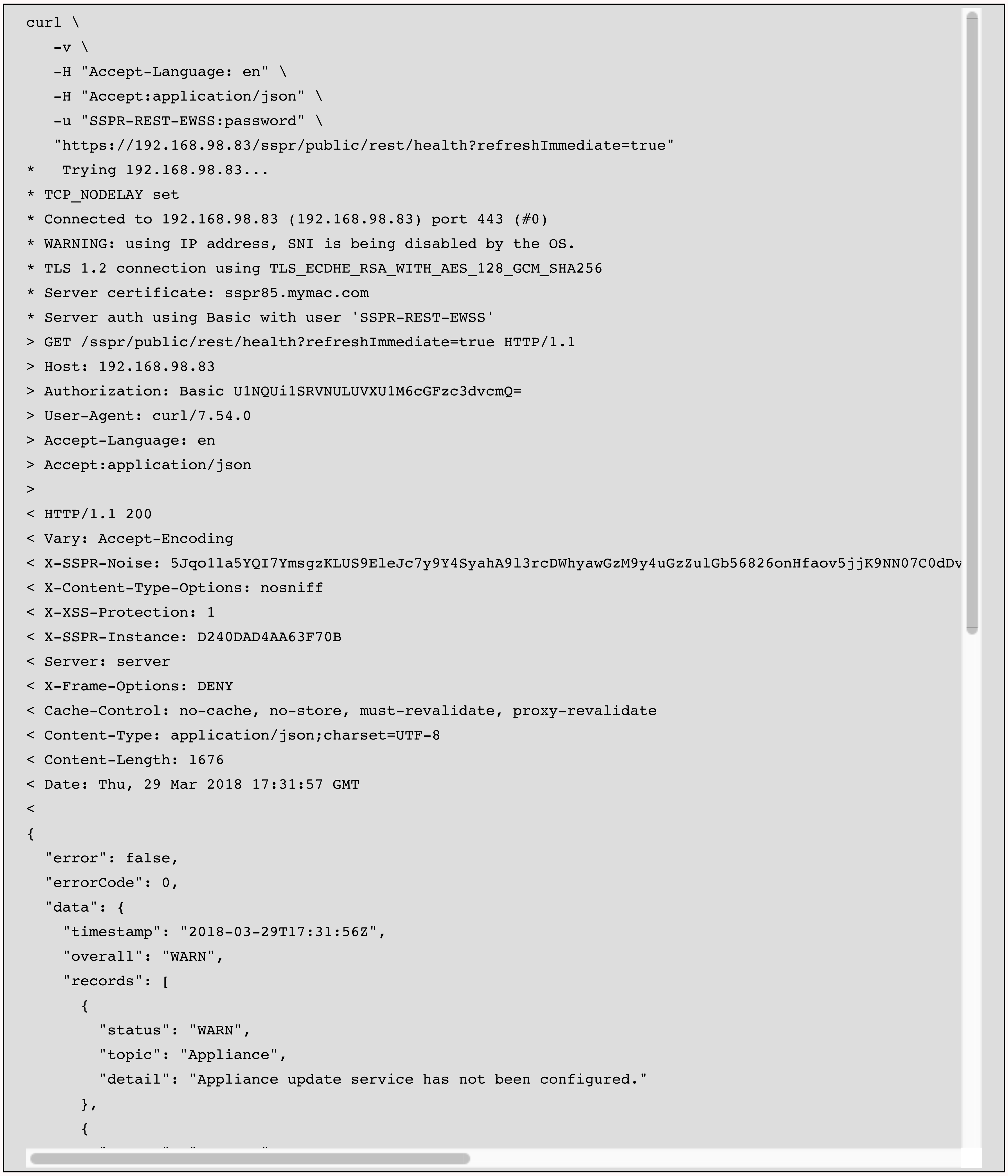
**detail** *string*

Explanation detail of this *status* condition.

## Examples

### Curl Example #1

The refreshImmidiate parameter as a query string, returning results in the content body as json formatted data.



```
curl \
   -v \
   -H "Accept-Language: en" \
   -H "Accept:application/json" \
   -u "SSPR-REST-EWSS:password" \
   "https://192.168.98.83/sspr/public/rest/health?refreshImmediate=true"
*   Trying 192.168.98.83...
* TCP_NODELAY set
* Connected to 192.168.98.83 (192.168.98.83) port 443 (#0)
* WARNING: using IP address, SNI is being disabled by the OS.
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
* Server certificate: sspr85.mymac.com
* Server auth using Basic with user 'SSPR-REST-EWSS'
> GET /sspr/public/rest/health?refreshImmediate=true HTTP/1.1
> Host: 192.168.98.83
> Authorization: Basic U1NQUi1SRVNULUVXU1M6cGFzc3dvcmQ=
> User-Agent: curl/7.54.0
> Accept-Language: en
> Accept:application/json
>
< HTTP/1.1 200
< Vary: Accept-Encoding
< X-SSPR-Noise: 5Jqo1la5YQI7YmsgzKLUS9EleJc7y9Y4SyahA9l3rcDWhyawGzM9y4uGzZulGb56826onHfaov5jjK9NN07C0dDv
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1
< X-SSPR-Instance: D240DAD4AA63F70B
< Server: server
< X-Frame-Options: DENY
< Cache-Control: no-cache, no-store, must-revalidate, proxy-revalidate
< Content-Type: application/json;charset=UTF-8
< Content-Length: 1676
< Date: Thu, 29 Mar 2018 17:31:57 GMT
<
{
  "error": false,
  "errorCode": 0,
  "data": {
    "timestamp": "2018-03-29T17:31:56Z",
    "overall": "WARN",
    "records": [
      {
        "status": "WARN",
        "topic": "Appliance",
        "detail": "Appliance update service has not been configured."
      },
      {
```

Curl Example #2

The refreshImmidiate parameter as a query string, returning results in the content body as a text string.

```
curl \
   -v \
   -H "Accept-Language: en" \
   -H "Accept:text/plain" \
   -u "SSPR-REST-EWSS:password" \
   "https://192.168.98.83/sspr/public/rest/health?refreshImmediate=true"
*   Trying 192.168.98.83...
* TCP_NODELAY set
* Connected to 192.168.98.83 (192.168.98.83) port 443 (#0)
* WARNING: using IP address, SNI is being disabled by the OS.
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
* Server certificate: sspr85.mymac.com
* Server auth using Basic with user 'SSPR-REST-EWSS'
> GET /sspr/public/rest/health?refreshImmediate=true HTTP/1.1
> Host: 192.168.98.83
> Authorization: Basic U1NQUi1SRVNULUVXU1M6cGFzc3dvcmQ=
> User-Agent: curl/7.54.0
> Accept-Language: en
> Accept:text/plain
>
< HTTP/1.1 200
< Vary: Accept-Encoding
< X-SSPR-Noise: GrqOmQZSriYFg14BcB53x5CmSybJq2SVZ5Klz8ST0an7WygPE0himDW9eVZCFtB0U9NxyZZXDou3Jj7fKqUGzCV3kg
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1
< X-SSPR-Instance: D240DAD4AA63F70B
< Server: server
< X-Frame-Options: DENY
< Cache-Control: no-cache, no-store, must-revalidate, proxy-revalidate
< Content-Type: text/plain;charset=UTF-8
< Content-Length: 5
< Date: Thu, 29 Mar 2018 19:18:03 GMT
<
WARN
* Connection #0 to host 192.168.98.83 left intact
```

# REST Service: /sspr/public/rest/profile

**NOTE:** Requires that SSPR be configured with the profile module (for both *GET* and *POST* methods):

Modules → Authenticated → Update Profile → Update Profile Settings = Enabled (True)

## GET Method

Retrieve user profile data.

### Request Headers

**Authorization** *Basic Authentication Method*

Required. Specifies credentials to authorize the REST caller. This is used by SSPR to validate the caller's rights to perform this action.

**Accept-Language** *Language code*

The request will be processed in the context of the specified language code. For more detail on SSPR locale codes, see Appendix F: Locales (Languages) and Flags.

**Accept** *Document protocol type*

Protocol type values:
*application/json*

### Request Query String Parameters

**username** *username or userDN*

Required. Target Username or userDN of the REST call.

### Request Content

None

### Response Status Codes

**200 OK** *Successful* The call was successfully serviced by SSPR.

### Response Headers

This list of HTTP response headers is not complete. Headers that are not listed here have little, or no impact for SSPR REST application developers.

**Content-Length** *bytes*

Number of octets (8-bit bytes) found in the *response content*.

**Date** *HTTP-date*

The date and time that the response originated in *HTTP-date* format as defined by RFC 7231 Date/Time Formats.

**Server** *server*

Historically, identifies the server software implemented to respond to the REST request. However, it represents an unnecessary leak of information of information about the platform. Rather than eliminating the header, the response includes this header, but with the innocuous value of *server*.

**Vary** *Accept-Encoding*

Informs cache stores that this response varies based on the value of the *Accept-Encoding* header.

**X-SSPR-Instance** *Name of the SSPR application instance.*

All requests from a specific SSPR server will have this value in common. The value of this header is specified in the SSPR *Configuration Editor*:

Settings→Application→Applications→Instance Name

## Response Content

The response content is a json formatted string. The json root object contents include the following child objects:

**error** *Boolean*

Indicates if the REST service experienced an error servicing the request.

**errorCode** *Integer*

If the root *error* object value is *false*, the value of this object will be 0.

If the root *error* object value is *true*, the value of this object will be a numeric SSPR error code. The meaning of the error code can be found in Appendix A: Errors.

Common SSPR *errorCode* Values:

| | | |
|---|---|---|
| **0** | *Success* | Operation completed successfully. |
| **5015** | *ERROR_UNKNOWN* | An error has occurred. If this error occurs repeatedly please contact your help desk. |

**successMessage** *UTF-8 string*

This object is present when the *error* object is *false*. The value of this object is a localized string, generated by SSPR, indicating that the operation has succeeded.

**errorMessage** *UTF-8 string*

This object is present when the *error* object is *true*. The value of this object is a localized string, generated by SSPR, indicating why the operation has failed.

**errorDetail** *UTF-8 string*

This object is present when the *error* object is *true*. The value of this object is a non-localized (English only) string, generated by SSPR, including the *errorCode* and the error *key*.

**data**

This object will be present if the root *error* object value is *false*, and contains the following child objects:

**profile** *object*

This object contains user profile objects. These objects will vary depending on the objects defined in the user profile. The following child objects represent example user profile objects that might be encountered:

**telephoneNumber** *string*

The target user's telephone number.

**mail** *string*

The target user's email address.

**title** *string*

The target user's title.

**formDefinition** *Array object*

Contains an array of child objects, one for each profile object. each child representing of the object found in the *profile* object (above).

Each child object contains the objects that describe the limitations of the target *profile* (object in the section above):

**name** *string*

Name of the target object (found in the *profile* section above).

**minimumLength** *integer*

The minimum acceptable string length of the target object's value.

**maximumLength** *integer*

The maximum acceptable string length of the target object's value.

**type** *string*

The format type of the target object's value. Format types include:

| | |
|---|---|
| **checkbox** | A binary choice. |
| **email** | Email address. |
| **hidden** | Not displayed. |
| **number** | Numeric value. |
| **password** | Protected password value. |
| **random** | random value. |
| **select** | Item selected from a defined list. |
| **tel** | Telephone number. |
| **text** | Free text. |
| **url** | A Uniform Resource Locator (URL). Colloquially termed a *web address*. |

**required** *boolean*

Indicates that this value may not be empty.

**confirmationRequired** *boolean*

Requires the value to be entered, by the user, twice; and that both entered values are the same value.

**readonly** *boolean*

The value is displayed to the user, but cannot be modified by the user.

**unique** *boolean*

Indicates that the value is maintained as unique by the value's source (*LDAP directory*, or *Remote Form Data Service*).

**multivalue** *boolean*

Indicates that the target object value may contain multiple sub-values.

**labels** *Array object*

Lists an array of localized displayable label child-objects for the target object, where each child-object has a name (containing the SSPR locale code) and a UTF-8 string of the displayable label for the indicated locale.

For more detail on SSPR locale codes, see Appendix F: Locales (Languages) and Flags.

**regexErrors** *Array object*

Lists an array of localized displayable regexErrors child-objects for the target object, where each child-object has a name (representing SSPR locale code) and a UTF-8 string of the displayable regexError for the indicated locale.

For more detail on SSPR locale codes, see Appendix F: Locales (Languages) and Flags.

**description** *Array object*

Lists an array of localized displayable description child-objects for the target object, where each child-object has a name (representing the SSPR locale code) and a UTF-8 string of the displayable description for the indicated locale.

For more detail on SSPR locale codes, see Appendix F: Locales (Languages) and Flags.

**selectOptions** *Array object*

If the target object has a *type* of *select*, this object will contain the list of options from which the user makes a selection.

## Examples

Curl Example #1

The target username parameter specified in the query string.

```
curl \
   -v \
   -H "Accept-Language: en" \
   -H "Accept: application/json" \
   -u "SSPR-REST-EWSS:password" \
   "https://192.168.98.83/sspr/public/rest/profile?username=sspr-testuser-01"
*   Trying 192.168.98.83...
* TCP_NODELAY set
* Connected to 192.168.98.83 (192.168.98.83) port 443 (#0)
* WARNING: using IP address, SNI is being disabled by the OS.
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
* Server certificate: sspr85.mymac.com
* Server auth using Basic with user 'SSPR-REST-EWSS'
> GET /sspr/public/rest/profile?username=sspr-testuser-01 HTTP/1.1
> Host: 192.168.98.83
> Authorization: Basic U1NQUi1SRVNULUVXU1M6cGFzc3dvcmQ=
> User-Agent: curl/7.54.0
> Accept-Language: en
> Accept: application/json
>
< HTTP/1.1 200
< Vary: Accept-Encoding
< X-SSPR-Noise: C2xVfKLQaglcc90qNeU6IHxCsLsSoW5uYpB3MIlZlpq647P0LSSy8EUcrSBAKA416gdaXqYRYxuTr5WIp9NfjY48
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1
< X-SSPR-Instance: D240DAD4AA63F70B
< Server: server
< X-Frame-Options: DENY
< Cache-Control: no-cache, no-store, must-revalidate, proxy-revalidate
< Content-Type: application/json;charset=UTF-8
< Content-Length: 1610
< Date: Thu, 29 Mar 2018 19:42:43 GMT
<
{
  "error": false,
  "errorCode": 0,
  "data": {
    "profile": {
      "telephoneNumber": "1-234-567-8901",
      "mail": "genious@novell.com",
      "title": "Genious"
    },
    "formDefinition": [
      {
        "name": "mail",
        "minimumLength": 1,
        "maximumLength": 64,
        "type": "email",
        "required": true,
```

## Examples

Curl Example #2

No target username parameter specified.

In this case, the target user is the LDAP authenticated REST user *SSPR-REST-LDAP*.

```
curl \
   -v \
   -H "Accept-Language: en" \
   -H "Accept: application/json" \
   -u "SSPR-REST-LDAP:password" \
   "https://192.168.98.83/sspr/public/rest/profile"
*   Trying 192.168.98.83...
* TCP_NODELAY set
* Connected to 192.168.98.83 (192.168.98.83) port 443 (#0)
* WARNING: using IP address, SNI is being disabled by the OS.
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
* Server certificate: sspr85.mymac.com
* Server auth using Basic with user 'SSPR-REST-LDAP'
> GET /sspr/public/rest/profile HTTP/1.1
> Host: 192.168.98.83
> Authorization: Basic U1NQUi1SRVNULUxEQVA6cGFzc3dvcmQ=
> User-Agent: curl/7.54.0
> Accept-Language: en
> Accept: application/json
>
< HTTP/1.1 200
< Vary: Accept-Encoding
< X-SSPR-Noise: oZvKsprw23MNZVBEM6DEj4IrTeibNxaW5AkiUhYxkbfaiwFAgk
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1
< X-SSPR-Instance: D240DAD4AA63F70B
< Server: server
< X-Frame-Options: DENY
< Cache-Control: no-cache, no-store, must-revalidate, proxy-revalidate
< Content-Type: application/json;charset=UTF-8
< Content-Length: 1634
< Date: Mon, 02 Apr 2018 21:00:15 GMT
<
{
  "error": false,
  "errorCode": 0,
  "data": {
    "profile": {
      "telephoneNumber": "2-222-222-2222",
      "mail": "SSPR-Rest-LDAP@microfocus.com",
      "title": "User: SSPR-Rest-LDAP"
    },
    "formDefinition": [
      {
        "name": "mail",
        "minimumLength": 1,
        "maximumLength": 64,
        "type": "email",
        "required": true
```

# POST Method

Set user profile data.

## Request Headers

**Authorization** *Basic Authentication Method*

Required. Specifies credentials to authorize the REST caller. This is used by SSPR to validate the caller's rights to perform this action.

**Accept-Language** *Language code*

The request will be processed in the context of the specified language code. For more detail on SSPR locale codes, see Appendix F: Locales (Languages) and Flags.

**Accept** *Document protocol type*

Protocol type values:
*application/json*

**Content-Type** *Document protocol type*

Protocol type values:
*application/json*

## Request Query String Parameters

**username** *username or userDN*

Json object representing the target Username or userDN of the REST call.

* Known Issue: 1087773

## Request Content

**username** *username or userDN*

Required. Json object representing the target Username or userDN of the REST call.

* Known Issue: 1087776

**profile** *username or userDN*

Required. Json container object for profile fields and values. The contained named objects represent those found in the user profile:

Modules → Public → New User Registration → New User Profiles → *default*

```
"profile":
 {
 "title":"Genious",
 "description":"Genious User",
 "mail":"genious@novell.com"
 }
```

## Response Status Codes

**200 OK** *Successful* The call was successfully serviced by SSPR.

## Response Headers

This list of HTTP response headers is not complete. Headers that are not listed here have little, or no impact for SSPR REST application developers.

**Content-Type** *application/json;charset=UTF-8*

Indicates the type of the data in the response content body.

**Content-Length** *bytes*

Number of octets (8-bit bytes) found in the *response content*.

**Date** *HTTP-date*

The date and time that the response originated in *HTTP-date* format as defined by RFC 7231 Date/Time Formats.

**Server** *server*

Historically, identifies the server software implemented to respond to the REST request. However, it represents an unnecessary leak of information of information about the platform. Rather than eliminating the header, the response includes this header, but with the innocuous value of *server*.

**Vary** *Accept-Encoding*

Informs cache stores that this response varies based on the value of the *Accept-Encoding* header.

**X-SSPR-Instance** *Name of the SSPR application instance.*

All requests from a specific SSPR server will have this value in common. The value of this header is specified in the SSPR *Configuration Editor*:

Settings→Application→Applications→Instance Name

## Response Content

The response content is a json formatted string. The json root object contents include the following child objects:

**error** *Boolean*

Indicates if the REST service experienced an error servicing the request.

**errorCode** *Integer*

If the root *error* object value is *false*, the value of this object will be 0.

If the root *error* object value is *true*, the value of this object will be a numeric SSPR error code. The meaning of the error code can be found in Appendix A: Errors.

Common SSPR *errorCode* Values:

| | | |
|---|---|---|
| **0** | *Success* | Operation completed successfully. |
| **5015** | *ERROR_UNKNOWN* | An error has occurred. If this error occurs repeatedly please contact your help desk. |

**successMessage** *UTF-8 string*

This object is present when the *error* object is *false*. The value of this object is a localized string, generated by SSPR, indicating that the operation has succeeded.

**errorMessage** *UTF-8 string*

This object is present when the *error* object is *true*. The value of this object is a localized string, generated by SSPR, indicating why the operation has failed.

**errorDetail** *UTF-8 string*

This object is present when the *error* object is *true*. The value of this object is a non-localized (English only) string, generated by SSPR, including the *errorCode* and the error *key*.

## Examples

### Curl Example #1

All parameters passed in request content as json data.

```
curl \
   -v \
   -H "Accept-Language: en" \
   -H "Accept:application/json" \
   -H "Content-Type:application/json" \
   -u "SSPR-REST-EWSS:password" \
   "https://192.168.98.83/sspr/public/rest/profile" \
   -d '{"username":"sspr-testuser-01","profile":{"title":"Genious","description":"Genious User","mail":"ge
* TCP_NODELAY set
* Connected to 192.168.98.83 (192.168.98.83) port 443 (#0)
* WARNING: using IP address, SNI is being disabled by the OS.
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
* Server certificate: sspr85.mymac.com
* Server auth using Basic with user 'SSPR-REST-EWSS'
> POST /sspr/public/rest/profile HTTP/1.1
> Host: 192.168.98.83
> Authorization: Basic U1NQUi1SRVNULUVXU1M6cGFzc3dvcmQ=
> User-Agent: curl/7.54.0
> Accept-Language: en
> Accept:application/json
> Content-Type:application/json
> Content-Length: 118
>
* upload completely sent off: 118 out of 118 bytes
< HTTP/1.1 200
< Vary: Accept-Encoding
< X-SSPR-Noise: HSZqhLAOVDUaZC4TI3QE26tkz9BnKFg
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1
< X-SSPR-Instance: D240DAD4AA63F70B
< Server: server
< X-Frame-Options: DENY
< Cache-Control: no-cache, no-store, must-revalidate, proxy-revalidate
< Content-Type: application/json;charset=UTF-8
< Content-Length: 115
< Date: Mon, 02 Apr 2018 16:34:43 GMT
<
{
  "error": false,
  "errorCode": 0,
  "successMessage": "Your user information has been successfully updated."
}
* Connection #0 to host 192.168.98.83 left intact
```

## Examples

### Curl Example #2

The username parameter passed as a query string, profile data in request content as json data.

* Known Issue: 1087773

```
curl \
   -v \
   -H "Accept-Language: en" \
   -H "Accept:application/json" \
   -H "Content-Type:application/json" \
   -u "SSPR-REST-EWSS:password" \
   "https://192.168.98.83/sspr/public/rest/profile?username=sspr-testuser-01" \
   -d '{"profile":{"title":"Genious","description":"Genious User","mail":"genious@novell.com"}}'
*   Trying 192.168.98.83...
* TCP_NODELAY set
* Connected to 192.168.98.83 (192.168.98.83) port 443 (#0)
* WARNING: using IP address, SNI is being disabled by the OS.
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
* Server certificate: sspr85.mymac.com
* Server auth using Basic with user 'SSPR-REST-EWSS'
> POST /sspr/public/rest/profile?username=sspr-testuser-01 HTTP/1.1
> Host: 192.168.98.83
> Authorization: Basic U1NQUi1SRVNULUVXU1M6cGFzc3dvcmQ=
> User-Agent: curl/7.54.0
> Accept-Language: en
> Accept:application/json
> Content-Type:application/json
> Content-Length: 88
>
* upload completely sent off: 88 out of 88 bytes
< HTTP/1.1 200
< Vary: Accept-Encoding
< X-SSPR-Noise: COzwyz5VIfDOaIzDaPu7TKd6XbNLrSsEFTQ2FmIjeXNSpaHFHFe33vqQhOoiBaEBbmwfXBwMNu0tMG6bskx
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1
< X-SSPR-Instance: D240DAD4AA63F70B
< Server: server
< X-Frame-Options: DENY
< Cache-Control: no-cache, no-store, must-revalidate, proxy-revalidate
< Content-Type: application/json;charset=UTF-8
< Content-Length: 351
< Date: Mon, 02 Apr 2018 17:27:31 GMT
<
{
  "error": true,
  "errorCode": 5015,
  "errorMessage": "An error has occurred.  If this error occurs repeatedly please contact your help desk."
  "errorDetail": "5015 ERROR_UNKNOWN (unexpected error building json response: 7000 ERROR_REST_INVOCATION_
}
* Connection #0 to host 192.168.98.83 left intact
```

## Examples

### Curl Example #3

No username parameter (making the authenticated LDAP rest user the target), profile data in request content as json data.

* Known Issue: 1087776

```
curl \
    -v \
    -H "Accept-Language: en" \
    -H "Accept:application/json" \
    -H "Content-Type:application/json" \
    -u "SSPR-REST-LDAP:password" \
    "https://192.168.98.83/sspr/public/rest/profile" \
    -d '{"profile":{"employeeId":31415926,"description":"Genious User","mail":"genious@novell.com"}}'
*   Trying 192.168.98.83...
* TCP_NODELAY set
* Connected to 192.168.98.83 (192.168.98.83) port 443 (#0)
* WARNING: using IP address, SNI is being disabled by the OS.
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
* Server certificate: sspr85.mymac.com
* Server auth using Basic with user 'SSPR-REST-LDAP'
> POST /sspr/public/rest/profile HTTP/1.1
> Host: 192.168.98.83
> Authorization: Basic U1NQUi1SRVNULUxEQVA6cGFzc3dvcmQ=
> User-Agent: curl/7.54.0
> Accept-Language: en
> Accept:application/json
> Content-Type:application/json
> Content-Length: 92
>
* upload completely sent off: 92 out of 92 bytes
< HTTP/1.1 200
< Vary: Accept-Encoding
< X-SSPR-Noise: LaSh43fbW2Ox1AwRpMygVmADffD6KyNneEUnQD63rhlpsdTQ4pdYoMHf2BN8L4xuKCbkeHlhfXxejI9iQ0J77vyE4i
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1
< X-SSPR-Instance: D240DAD4AA63F70B
< Server: server
< X-Frame-Options: DENY
< Cache-Control: no-cache, no-store, must-revalidate, proxy-revalidate
< Content-Type: application/json;charset=UTF-8
< Content-Length: 423
< Date: Mon, 02 Apr 2018 17:57:42 GMT
<
{
  "error": true,
  "errorCode": 5015,
  "errorMessage": "An error has occurred.  If this error occurs repeatedly please contact your help desk."
  "errorDetail": "5015 ERROR_UNKNOWN (unexpected error building json response: 5015 ERROR_UNKNOWN (error s
}
* Connection #0 to host 192.168.98.83 left intact
```

# REST Service: /sspr/public/rest/randompassword

## GET Method

Generate (and return) a single random password value based on a specific user's password policy.

### Request Headers

**Authorization** *Basic Authentication Method*

Required. Specifies credentials to authorize the REST caller. This is used by SSPR to validate the caller's rights to perform this action.

**Accept-Language** *Language code*

The request will be processed in the context of the specified language code. For more detail on SSPR locale codes, see Appendix F: Locales (Languages) and Flags.

**Accept** *Document protocol type*

Protocol type values:
*application/json*

### Request Query String Parameters

**username** *username or userDN*

Optional. Target Username or userDN of the REST call. If this parameter is not specified, the user specified by the *Authorization* header is assumed.

**strength** *percentage (0-100)*

Optional. A percentage number (0-100) specifying the minimum strength of the generated password.

**minLength** *number*

Optional. A number specifying the minimum length of the generated password.

**chars** *list of characters*

Optional. A list of characters to use for generating the password.

### Request Content

None

### Response Status Codes

**200 OK** *Successful* The call was successfully serviced by SSPR.

## Response Headers

This list of HTTP response headers is not complete. Headers that are not listed here have little, or no impact for SSPR REST application developers.

**Content-Type** *application/json;charset=UTF-8*

Indicates the type of the data in the response content body.

**Content-Length** *bytes*

Number of octets (8-bit bytes) found in the *response content*.

**Date** *HTTP-date*

The date and time that the response originated in *HTTP-date* format as defined by RFC 7231 Date/Time Formats.

**Server** *server*

Historically, identifies the server software implemented to respond to the REST request. However, it represents an unnecessary leak of information of information about the platform. Rather than eliminating the header, the response includes this header, but with the innocuous value of *server*.

**Vary** *Accept-Encoding*

Informs cache stores that this response varies based on the value of the *Accept-Encoding* header.

**X-SSPR-Instance** *Name of the SSPR application instance.*

All requests from a specific SSPR server will have this value in common. The value of this header is specified in the SSPR *Configuration Editor*:

Settings→Application→Applications→Instance Name

## Response Content

The entire response content consists of the generated password value string.

## Examples

### Curl Example #1

No parameters.

In this case, the user password policy will be that of the SSPR's LDAP *test-user* policy.

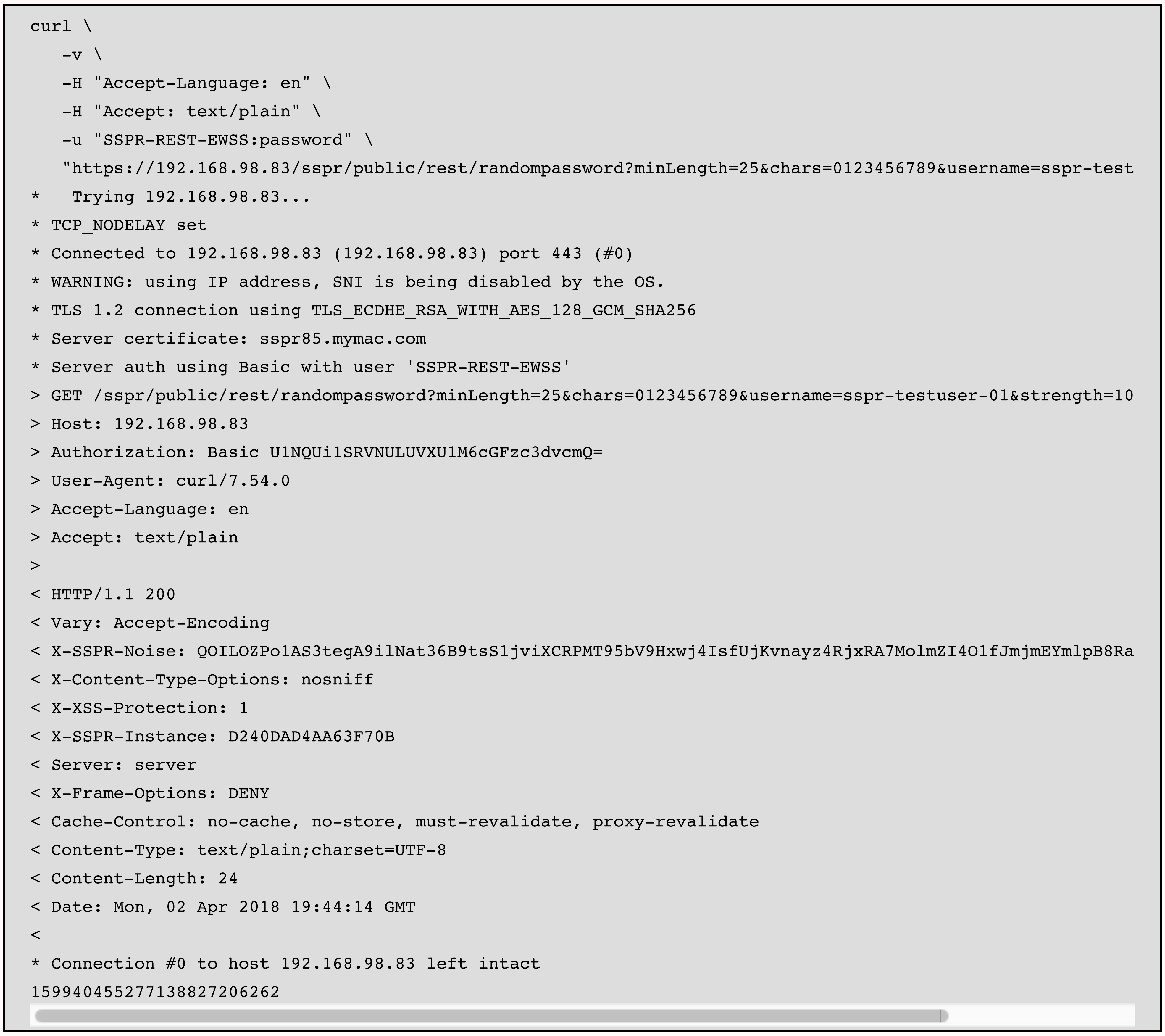
```
curl \
   -v \
   -H "Accept-Language: en" \
   -H "Accept:text/plain" \
   -u "SSPR-REST-EWSS:password" \
   "https://192.168.98.83/sspr/public/rest/randompassword"
*   Trying 192.168.98.83...
* TCP_NODELAY set
* Connected to 192.168.98.83 (192.168.98.83) port 443 (#0)
* WARNING: using IP address, SNI is being disabled by the OS.
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
* Server certificate: sspr85.mymac.com
* Server auth using Basic with user 'SSPR-REST-EWSS'
> GET /sspr/public/rest/randompassword HTTP/1.1
> Host: 192.168.98.83
> Authorization: Basic U1NQUi1SRVNULUVXU1M6cGFzc3dvcmQ=
> User-Agent: curl/7.54.0
> Accept-Language: en
> Accept:text/plain
>
< HTTP/1.1 200
< Vary: Accept-Encoding
< X-SSPR-Noise: 5oCIegfrSntdoEHGkLfwr0u6Rgzuilcx9fXgjTmhndik3GTqxOXyFPQc2CUvGJGr8xK0S9H4Q6oFrHAyVsN73deMdD
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1
< X-SSPR-Instance: D240DAD4AA63F70B
< Server: server
< X-Frame-Options: DENY
< Cache-Control: no-cache, no-store, must-revalidate, proxy-revalidate
< Content-Type: text/plain;charset=UTF-8
< Content-Length: 12
< Date: Mon, 02 Apr 2018 19:24:19 GMT
<
* Connection #0 to host 192.168.98.83 left intact
alMbeHAnians
```

The resulting generated the password: **alMbeHAnians**

Curl Example #2

With parameters as a query string:

```
Minimum length...: 25
character Set....: 0123456789
Password Strength: 100
All other password policy attributes
   based on user.: sspr-testuser-01
```



```
curl \
   -v \
   -H "Accept-Language: en" \
   -H "Accept: text/plain" \
   -u "SSPR-REST-EWSS:password" \
   "https://192.168.98.83/sspr/public/rest/randompassword?minLength=25&chars=0123456789&username=sspr-test
*   Trying 192.168.98.83...
* TCP_NODELAY set
* Connected to 192.168.98.83 (192.168.98.83) port 443 (#0)
* WARNING: using IP address, SNI is being disabled by the OS.
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
* Server certificate: sspr85.mymac.com
* Server auth using Basic with user 'SSPR-REST-EWSS'
> GET /sspr/public/rest/randompassword?minLength=25&chars=0123456789&username=sspr-testuser-01&strength=10
> Host: 192.168.98.83
> Authorization: Basic U1NQUi1SRVNULUVXU1M6cGFzc3dvcmQ=
> User-Agent: curl/7.54.0
> Accept-Language: en
> Accept: text/plain
>
< HTTP/1.1 200
< Vary: Accept-Encoding
< X-SSPR-Noise: QOILOZPo1AS3tegA9ilNat36B9tsS1jviXCRPMT95bV9Hxwj4IsfUjKvnayz4RjxRA7MolmZI4O1fJmjmEYmlpB8Ra
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1
< X-SSPR-Instance: D240DAD4AA63F70B
< Server: server
< X-Frame-Options: DENY
< Cache-Control: no-cache, no-store, must-revalidate, proxy-revalidate
< Content-Type: text/plain;charset=UTF-8
< Content-Length: 24
< Date: Mon, 02 Apr 2018 19:44:14 GMT
<
* Connection #0 to host 192.168.98.83 left intact
159940455277138827206262
```

The resulting generated the password: **159940455277138827206262**

# POST Method

Generate a single random password value.

## Request Headers

**Authorization** *Basic Authentication Method*

Required. Specifies credentials to authorize the REST caller. This is used by SSPR to validate the caller's rights to perform this action.

**Accept-Language** *Language code*

The request will be processed in the context of the specified language code. For more detail on SSPR locale codes, see Appendix F: Locales (Languages) and Flags.

**Accept** *Document protocol type*

Protocol type values:
*application/json*

**Content-Type** *Document protocol type*

***NOTE:*** *This header is required, even when all parameters are specified in the query string and there is no content following the HTTP headers section.*

Protocol type values:
*application/json*

## Request Query String Parameters

**username** *username or userDN*

Optional. Target Username or userDN of the REST call. If this parameter is not specified, the user specified by the *Authorization* header is assumed.

**strength** *Integer (0-100)*

Optional. An integer (0-100) specifying the minimum strength of the generated password (0=weak - 100=strong).

**minLength** *number*

Optional. A number specifying the minimum length of the generated password.

**chars** *list of characters*

Optional. A list of characters to use for generating the password.

## Request Content

**username** *username or userDN*

Optional. Target Username or userDN of the REST call. If this parameter is not specified, the user specified by the *Authorization* header is assumed.

**strength** *Integer (0-100)*

Optional. An integer (0-100) specifying the minimum strength of the generated password (0=weak - 100=strong).

**minLength** *number*

Optional. A number specifying the minimum length of the generated password.

**chars** *list of characters*

Optional. A list of characters to use for generating the password.

## Response Status Codes

**200 OK** *Successful* The call was successfully serviced by SSPR.

## Response Headers

This list of HTTP response headers is not complete. Headers that are not listed here have little, or no impact for SSPR REST application developers.

**Content-Type** *application/json;charset=UTF-8*

Indicates the type of the data in the response content body.

**Content-Length** *bytes*

Number of octets (8-bit bytes) found in the *response content*.

**Date** *HTTP-date*

The date and time that the response originated in *HTTP-date* format as defined by RFC 7231 Date/Time Formats.

**Server** *server*

Historically, identifies the server software implemented to respond to the REST request. However, it represents an unnecessary leak of information of information about the platform. Rather than eliminating the header, the response includes this header, but with the innocuous value of *server*.

**Vary** *Accept-Encoding*

Informs cache stores that this response varies based on the value of the *Accept-Encoding* header.

**X-SSPR-Instance** *Name of the SSPR application instance.*

All requests from a specific SSPR server will have this value in common. The value of this header is specified in the SSPR *Configuration Editor*:

Settings→Application→Applications→Instance Name

## Response Content

**error** *Boolean*

Indicates if the REST service experienced an error servicing the request.

**errorCode** *Integer*

If the root *error* object value is *false*, the value of this object will be 0.

If the root *error* object value is *true*, the value of this object will be a numeric SSPR error code. The meaning of the error code can be found in Appendix A: Errors.

Common SSPR *errorCode* Values:

| | | |
|---|---|---|
| **0** | *Success* | Operation completed successfully. |
| **5015** | *ERROR_UNKNOWN* | An error has occurred. If this error occurs repeatedly please contact your help desk. |

**successMessage** *UTF-8 string*

This object is present when the *error* object is *false*. The value of this object is a localized string, generated by SSPR, indicating that the operation has succeeded.

**errorMessage** *UTF-8 string*

This object is present when the *error* object is *true*. The value of this object is a localized string, generated by SSPR, indicating why the operation has failed.

**errorDetail** *UTF-8 string*

This object is present when the *error* object is *true*. The value of this object is a non-localized (English only) string, generated by SSPR, including the *errorCode* and the error *key*.

**data** *object*

This object will be present if the root *error* object value is *false*, and contains the following child object:

**password** *UTF-8 string*

The generated password value string.

## Examples

### Curl Example #1

Parameters as reply content in json format.

```
curl \
   -v \
   -H "Accept-Language: en" \
   -H "Accept:application/json" \
   -H "Content-Type:application/json" \
   -u "SSPR-REST-EWSS:password" \
   "https://192.168.98.83/sspr/public/rest/randompassword" \
   -d'{"username":"sspr-testuser-01","strength":100,"minLength":25,"chars":"1234567890"}'
*   Trying 192.168.98.83...
* TCP_NODELAY set
* Connected to 192.168.98.83 (192.168.98.83) port 443 (#0)
* WARNING: using IP address, SNI is being disabled by the OS.
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
* Server certificate: sspr85.mymac.com
* Server auth using Basic with user 'SSPR-REST-EWSS'
> POST /sspr/public/rest/randompassword HTTP/1.1
> Host: 192.168.98.83
> Authorization: Basic U1NQUi1SRVNULUVXU1M6cGFzc3dvcmQ=
> User-Agent: curl/7.54.0
> Accept-Language: en
> Accept:application/json
> Content-Type:application/json
> Content-Length: 82
>
* upload completely sent off: 82 out of 82 bytes
< HTTP/1.1 200
< Vary: Accept-Encoding
< X-SSPR-Noise: 8g7h16dwYzNaf7LwuoEXKb9AMpADvoQiv1jaeE7
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1
< X-SSPR-Instance: D240DAD4AA63F70B
< Server: server
< X-Frame-Options: DENY
< Cache-Control: no-cache, no-store, must-revalidate, proxy-revalidate
< Content-Type: application/json;charset=UTF-8
< Content-Length: 99
< Date: Mon, 02 Apr 2018 21:56:59 GMT
<
{
  "error": false,
  "errorCode": 0,
  "data": {
    "password": "9701986003318973810499 91"
  }
}
* Connection #0 to host 192.168.98.83 left intact
```

The resulting generated the password: **9701986003318973810499 91**

## Curl Example #2

Parameters as a query string.

* Known Issue: 1087797

```
curl \
   -v \
   -X POST \
   -H "Accept-Language: en" \
   -H "Accept:application/json" \
   -H "Content-Type:application/x-www-form-urlencoded" \
   -u "SSPR-REST-EWSS:password" \
   "https://192.168.98.83/sspr/public/rest/randompassword?username=sspr-testuser-01&strength=100&minLength
*   Trying 192.168.98.83...
* TCP_NODELAY set
* Connected to 192.168.98.83 (192.168.98.83) port 443 (#0)
* WARNING: using IP address, SNI is being disabled by the OS.
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
* Server certificate: sspr85.mymac.com
* Server auth using Basic with user 'SSPR-REST-EWSS'
> POST /sspr/public/rest/randompassword?username=sspr-testuser-01&strength=100&minLength=25&chars=12345678
> Host: 192.168.98.83
> Authorization: Basic U1NQUi1SRVNULUVXU1M6cGFzc3dvcmQ=
> User-Agent: curl/7.54.0
> Accept-Language: en
> Accept:application/json
> Content-Type:application/x-www-form-urlencoded
>
< HTTP/1.1 200
< Vary: Accept-Encoding
< X-SSPR-Noise: VrIhwZJsIZpuLLXwgqcZp345I48Is9S9aJWWVtBeE32uGMejLDQ
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1
< X-SSPR-Instance: D240DAD4AA63F70B
< Server: server
< X-Frame-Options: DENY
< Cache-Control: no-cache, no-store, must-revalidate, proxy-revalidate
< Content-Type: application/json;charset=UTF-8
< Content-Length: 100
< Date: Mon, 02 Apr 2018 22:27:12 GMT
<
{
  "error": false,
  "errorCode": 0,
  "data": {
    "password": "2575677368354)11503432425"
  }
}
* Connection #0 to host 192.168.98.83 left intact
```

The resulting generated the password: **2575677368354)11503432425**

Curl Example #3

No parameters.

In this case, the user password policy will be that of the SSPR's LDAP *test-user* policy.

```
curl \
   -v \
   -X POST \
   -H "Accept-Language: en" \
   -H "Accept:application/json" \
   -H "Content-Type:application/x-www-form-urlencoded" \
   -u "SSPR-REST-EWSS:password" \
   "https://192.168.98.83/sspr/public/rest/randompassword"
*   Trying 192.168.98.83...
* TCP_NODELAY set
* Connected to 192.168.98.83 (192.168.98.83) port 443 (#0)
* WARNING: using IP address, SNI is being disabled by the OS.
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
* Server certificate: sspr85.mymac.com
* Server auth using Basic with user 'SSPR-REST-EWSS'
> POST /sspr/public/rest/randompassword HTTP/1.1
> Host: 192.168.98.83
> Authorization: Basic U1NQUi1SRVNULUVXU1M6cGFzc3dvcmQ=
> User-Agent: curl/7.54.0
> Accept-Language: en
> Accept:application/json
> Content-Type:application/x-www-form-urlencoded
>
< HTTP/1.1 200
< Vary: Accept-Encoding
< X-SSPR-Noise: wjzxE5UFukDWPvMilVgUeH9cD0PnrdFDObY1DI5RNet96LbxzzsvfofSnM4sqVm1FFP0ORjtqI97I9JPJ2B
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1
< X-SSPR-Instance: D240DAD4AA63F70B
< Server: server
< X-Frame-Options: DENY
< Cache-Control: no-cache, no-store, must-revalidate, proxy-revalidate
< Content-Type: application/json;charset=UTF-8
< Content-Length: 86
< Date: Mon, 02 Apr 2018 22:45:19 GMT
<
{
  "error": false,
  "errorCode": 0,
  "data": {
    "password": "WaluRihUght"
  }
}
* Connection #0 to host 192.168.98.83 left intact
```

The resulting generated the password: **WaluRihUght**

# REST Service: /sspr/public/rest/setpassword

## POST Method

Set a single password value.

Parameters may be specified as query string arguments, or as application/json or *application/x-www.form-urlencoded* formatted request content.

### Request Headers

**Authorization** *Basic Authentication Method*

Required. Specifies credentials to authorize the REST caller. This is used by SSPR to validate the caller's rights to perform this action.

**Accept-Language** *Language code*

The request will be processed in the context of the specified language code. For more detail on SSPR locale codes, see Appendix F: Locales (Languages) and Flags.

**Accept** *Document protocol type*

Protocol type values:
*application/json*

**Content-Length** *bytes*

Number of octets (8-bit bytes) found in the *response content*.

**Content-Type** *Document protocol type*

***NOTE:*** *This header is required, even when all parameters are specified in the query string and there is no content following the HTTP headers section.*

Protocol type values:
*application/json*
*application/x-www-form-urlencoded*

### Request Query String Parameters

**username** *username or userDN*

Optional. Target Username or userDN of the REST call. If this parameter is not specified, the user specified by the *Authorization* header is assumed.

**password** *New password*

Required when the *random* parameter is absent, or set to *false*. New password string.

**random** *boolean*

Optional. Generate a random password. When *true*, the *password* parameter is not be specified.

## Request Content

**username** *username or userDN*

Optional. Target Username or userDN of the REST call. If this parameter is not specified, the user specified by the *Authorization* header is assumed.

**password** *New password*

Required when the *random* parameter is absent, or set to *false*. New password string.

**random** *boolean*

Optional. Generate a random password. When *true*, the *password* parameter is not be specified.

## Response Status Codes

**200 OK** *Successful* The call was successfully serviced by SSPR.

## Response Headers

This list of HTTP response headers is not complete. Headers that are not listed here have little, or no impact for SSPR REST application developers.

**Content-Type** *application/json;charset=UTF-8*

Indicates the type of the data in the response content body.

**Content-Length** *bytes*

Number of octets (8-bit bytes) found in the *response content*.

**Date** *HTTP-date*

The date and time that the response originated in *HTTP-date* format as defined by RFC 7231 Date/Time Formats.

**Server** *server*

Historically, identifies the server software implemented to respond to the REST request. However, it represents an unnecessary leak of information of information about the platform. Rather than eliminating the header, the response includes this header, but with the innocuous value of *server*.

**Vary** *Accept-Encoding*

Informs cache stores that this response varies based on the value of the *Accept-Encoding* header.

**X-SSPR-Instance** *Name of the SSPR application instance.*

All requests from a specific SSPR server will have this value in common. The value of this header is specified in the SSPR *Configuration Editor*:

Settings→Application→Applications→Instance Name

## Response Content

**error** *Boolean*

Indicates if the REST service experienced an error servicing the request.

**errorCode** *Integer*

If the root *error* object value is *false*, the value of this object will be 0.

If the root *error* object value is *true*, the value of this object will be a numeric SSPR error code. The meaning of the error code can be found in Appendix A: Errors.

Common SSPR *errorCode* Values:

| | | |
|---|---|---|
| **0** | *Success* | Operation completed successfully. |
| **5015** | *ERROR_UNKNOWN* | An error has occurred. If this error occurs repeatedly please contact your help desk. |

**successMessage** *UTF-8 string*

This object is present when the *error* object is *false*. The value of this object is a localized string, generated by SSPR, indicating that the operation has succeeded.

**errorMessage** *UTF-8 string*

This object is present when the *error* object is *true*. The value of this object is a localized string, generated by SSPR, indicating why the operation has failed.

**errorDetail** *UTF-8 string*

This object is present when the *error* object is *true*. The value of this object is a non-localized (English only) string, generated by SSPR, including the *errorCode* and the error *key*.

**data** *object*

This object will be present if the root *error* object value is *false*, and contains the following child object:

**username** *UTF-8 string*

Specifies the name of the SSPR password policy applied, and the target user DN, separated by a pipe ('|') character:

*(policy name)|(user DN)*

Example: `default|cn=sspr-testuser-01,ou=users,o=sles12`

In the case of this example, the SSPR password policy can be found in the SSPR *Configuration Editor*:

Policies → Password Policies → default

**random** *Boolean*

Indicates how the request parameter *random* was specified.

## Examples

### Curl Example #1

With parameters as a content in json format.

```
curl \
   -v \
   -H "Accept-Language: en" \
   -H "Accept:application/json" \
   -H "Content-Type:application/json" \
   -u "SSPR-REST-EWSS:password" \
   "https://192.168.98.83/sspr/public/rest/setpassword" \
   -d '{"username":"sspr-testuser-01","password":"herbie"}'
*   Trying 192.168.98.83...
* TCP_NODELAY set
* Connected to 192.168.98.83 (192.168.98.83) port 443 (#0)
* WARNING: using IP address, SNI is being disabled by the OS.
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
* Server certificate: sspr85.mymac.com
* Server auth using Basic with user 'SSPR-REST-EWSS'
> POST /sspr/public/rest/setpassword HTTP/1.1
> Host: 192.168.98.83
> Authorization: Basic U1NQUi1SRVNULUVXU1M6cGFzc3dvcmQ=
> User-Agent: curl/7.54.0
> Accept-Language: en
> Accept:application/json
> Content-Type:application/json
> Content-Length: 51
>
* upload completely sent off: 51 out of 51 bytes
< HTTP/1.1 200
< Vary: Accept-Encoding
< X-SSPR-Noise: 3gcXltihOveTwfsanjm0cMReYaAjboQxfQx4FlKnE18q2PXlmlAKLXi5PFgw060FrwgUEQeAAuRaEd
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1
< X-SSPR-Instance: D240DAD4AA63F70B
< Server: server
< X-Frame-Options: DENY
< Cache-Control: no-cache, no-store, must-revalidate, proxy-revalidate
< Content-Type: application/json;charset=UTF-8
< Content-Length: 208
< Date: Tue, 03 Apr 2018 15:22:32 GMT
<
{
  "error": false,
  "errorCode": 0,
  "successMessage": "The password has been changed successfully.",
  "data": {
    "username": "default|cn=sspr-testuser-01,ou=users,o=sles12",
    "random": false
  }
}
* Connection #0 to host 192.168.98.83 left intact
```

### Curl Example #2

With parameters as a content in *x-www-form-urlencoded* format.

```
curl \
   -v \
   -H "Accept-Language: en" \
   -H "Accept:application/json" \
   -H "Content-Type:application/x-www-form-urlencoded" \
   -u "SSPR-REST-EWSS:password" \
   "https://192.168.98.83/sspr/public/rest/setpassword" \
   -d 'username=sspr-testuser-01&password=herbie'
*   Trying 192.168.98.83...
* TCP_NODELAY set
* Connected to 192.168.98.83 (192.168.98.83) port 443 (#0)
* WARNING: using IP address, SNI is being disabled by the OS.
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
* Server certificate: sspr85.mymac.com
* Server auth using Basic with user 'SSPR-REST-EWSS'
> POST /sspr/public/rest/setpassword HTTP/1.1
> Host: 192.168.98.83
> Authorization: Basic U1NQUi1SRVNULUVXU1M6cGFzc3dvcmQ=
> User-Agent: curl/7.54.0
> Accept-Language: en
> Accept:application/json
> Content-Type:application/x-www-form-urlencoded
> Content-Length: 41
>
* upload completely sent off: 41 out of 41 bytes
< HTTP/1.1 200
< Vary: Accept-Encoding
< X-SSPR-Noise: euZCcSGcKJwOu6CcHMHuXJjuPANWxbSlKYNe3oTkBjXFPPfXX9bmoDExUzuUYBD4IKC6LFG4aiqCDCRir5ZtwfDaO9
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1
< X-SSPR-Instance: D240DAD4AA63F70B
< Server: server
< X-Frame-Options: DENY
< Cache-Control: no-cache, no-store, must-revalidate, proxy-revalidate
< Content-Type: application/json;charset=UTF-8
< Content-Length: 208
< Date: Tue, 03 Apr 2018 16:12:32 GMT
<
{
  "error": false,
  "errorCode": 0,
  "successMessage": "The password has been changed successfully.",
  "data": {
    "username": "default|cn=sspr-testuser-01,ou=users,o=sles12",
    "random": false
  }
}
* Connection #0 to host 192.168.98.83 left intact
```

Curl Example #3

Parameters as a query string.

```
curl \
   -v \
   -X POST \
   -H "Accept-Language: en" \
   -H "Accept:application/json" \
   -H "Content-Type:application/x-www-form-urlencoded" \
   -u "SSPR-REST-EWSS:password" \
   "https://192.168.98.83/sspr/public/rest/setpassword?username=sspr-testuser-01&password=herbie"
*   Trying 192.168.98.83...
* TCP_NODELAY set
* Connected to 192.168.98.83 (192.168.98.83) port 443 (#0)
* WARNING: using IP address, SNI is being disabled by the OS.
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
* Server certificate: sspr85.mymac.com
* Server auth using Basic with user 'SSPR-REST-EWSS'
> POST /sspr/public/rest/setpassword?username=sspr-testuser-01&password=herbie HTTP/1.1
> Host: 192.168.98.83
> Authorization: Basic U1NQUi1SRVNULUVXU1M6cGFzc3dvcmQ=
> User-Agent: curl/7.54.0
> Accept-Language: en
> Accept:application/json
> Content-Type:application/x-www-form-urlencoded
>
< HTTP/1.1 200
< Vary: Accept-Encoding
< X-SSPR-Noise: naeD1Oev9Uhm1CCwroDhYeZLCEcDw8XNZdXwm4aFM10JWP79BuSQdr03ZdeCF3rFeOGMSn6ob2I7VByK7RghHoIkUw
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1
< X-SSPR-Instance: D240DAD4AA63F70B
< Server: server
< X-Frame-Options: DENY
< Cache-Control: no-cache, no-store, must-revalidate, proxy-revalidate
< Content-Type: application/json;charset=UTF-8
< Content-Length: 208
< Date: Tue, 03 Apr 2018 16:17:51 GMT
<
{
  "error": false,
  "errorCode": 0,
  "successMessage": "The password has been changed successfully.",
  "data": {
    "username": "default|cn=sspr-testuser-01,ou=users,o=sles12",
    "random": false
  }
}
* Connection #0 to host 192.168.98.83 left intact
```

### Curl Example #4

Set a random password, parameters as a query string.

```
curl \
   -v \
   -X POST \
   -H "Accept-Language: en" \
   -H "Accept:application/json" \
   -H "Content-Type:application/x-www-form-urlencoded" \
   -u "SSPR-REST-EWSS:password" \
   "https://192.168.98.83/sspr/public/rest/setpassword?username=sspr-testuser-01&random=true"
*   Trying 192.168.98.83...
* TCP_NODELAY set
* Connected to 192.168.98.83 (192.168.98.83) port 443 (#0)
* WARNING: using IP address, SNI is being disabled by the OS.
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
* Server certificate: sspr85.mymac.com
* Server auth using Basic with user 'SSPR-REST-EWSS'
> POST /sspr/public/rest/setpassword?username=sspr-testuser-01&random=true HTTP/1.1
> Host: 192.168.98.83
> Authorization: Basic U1NQUi1SRVNULUVXU1M6cGFzc3dvcmQ=
> User-Agent: curl/7.54.0
> Accept-Language: en
> Accept:application/json
> Content-Type:application/x-www-form-urlencoded
>
< HTTP/1.1 200
< Vary: Accept-Encoding
< X-SSPR-Noise: kw37RXeXFzN6V8809LwSu1thrpoeK77S1V4HJD8fbxj47HDfc9dh3VuqPpAd
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1
< X-SSPR-Instance: D240DAD4AA63F70B
< Server: server
< X-Frame-Options: DENY
< Cache-Control: no-cache, no-store, must-revalidate, proxy-revalidate
< Content-Type: application/json;charset=UTF-8
< Content-Length: 207
< Date: Tue, 03 Apr 2018 16:23:41 GMT
<
{
  "error": false,
  "errorCode": 0,
  "successMessage": "The password has been changed successfully.",
  "data": {
    "username": "default|cn=sspr-testuser-01,ou=users,o=sles12",
    "random": true
  }
}
* Connection #0 to host 192.168.98.83 left intact
```

# REST Service: /sspr/public/rest/signing/form

## POST Method

Pre-sign, and encrypt, form data for injection into an SSPR user form request. SSPR user form requests do not permit a remote application to POST data directly to them through a browser.

A remote application can use this REST API to pre-sign form data, and can then POST this pre-signed form data to SSPR in place of actual form data content.

The data returned from this REST API represents the form data submitted in the request, in an SSPR signed and encrypted format.

It is expected that the data returned will then accompany an SSPR request, such as the creation of a new user. This is done by attaching the data returned from this REST API as the value of a *signedForm* parameter in the *query string* of the SSPR request, instead of actual form data.

The data returned expires after a period of time.

### Request Headers

**Authorization** *Basic Authentication Method*

Required. Specifies credentials to authorize the REST caller. This is used by SSPR to validate the caller's rights to perform this action.

Use a named secret *username:secret* value in SSPR's *Configuration Editor*:

Settings → Web Services → REST Services → Web Service Non-LDAP Users and Passwords

**Accept-Language** *Language code*

The request will be processed in the context of the specified language code. For more detail on SSPR locale codes, see Appendix F: Locales (Languages) and Flags.

**Accept** *Document protocol type*

Protocol type values:
*application/json*

**Content-Type** *Document protocol type*

***NOTE:*** *This header is required, even when all parameters are specified in the query string and there is no content following the HTTP headers section.*

Protocol type values:
*application/json*
*application/x-www-form-urlencoded* * Known Issue: 1087993

### Request Query String Parameters

None

## Request Content

Contains *application/json* or *application/x-www-form-urlencoded* form data to be signed. Examples:

**application/json:**

{"givenName":"John","sn":"Doe","mail":"john.doe@novell.com"}.

**application/x-www-form-urlencoded:**

givenName=John&sn=Doe&mail=john%2Edoe%40novell%2Ecom

## Response Status Codes

**200 OK** *Successful* The call was successfully serviced by SSPR.

## Response Headers

This list of HTTP response headers is not complete. Headers that are not listed here have little, or no impact for SSPR REST application developers.

**Content-Type** *application/json;charset=UTF-8*

Indicates the type of the data in the response content body.

**Content-Length** *bytes*

Number of octets (8-bit bytes) found in the *response content*.

**Date** *HTTP-date*

The date and time that the response originated in *HTTP-date* format as defined by RFC 7231 Date/Time Formats.

**Server** *server*

Historically, identifies the server software implemented to respond to the REST request. However, it represents an unnecessary leak of information of information about the platform. Rather than eliminating the header, the response includes this header, but with the innocuous value of *server*.

**Vary** *Accept-Encoding*

Informs cache stores that this response varies based on the value of the *Accept-Encoding* header.

**X-SSPR-Instance** *Name of the SSPR application instance.*

All requests from a specific SSPR server will have this value in common. The value of this header is specified in the SSPR *Configuration Editor*:

Settings→Application→Applications→Instance Name

## Response Content

**error** *Boolean*

Indicates if the REST service experienced an error servicing the request.

**errorCode** *Integer*

If the root *error* object value is *false*, the value of this object will be 0.

If the root *error* object value is *true*, the value of this object will be a numeric SSPR error code. The meaning of the error code can be found in Appendix A: Errors.

Common SSPR *errorCode* Values:

| | | |
|---|---|---|
| **0** | *Success* | Operation completed successfully. |
| **5015** | *ERROR_UNKNOWN* | An error has occurred. If this error occurs repeatedly please contact your help desk. |

**successMessage** *UTF-8 string*

This object is present when the *error* object is *false*. The value of this object is a localized string, generated by SSPR, indicating that the operation has succeeded.

**errorMessage** *UTF-8 string*

This object is present when the *error* object is *true*. The value of this object is a localized string, generated by SSPR, indicating why the operation has failed.

**errorDetail** *UTF-8 string*

This object is present when the *error* object is *true*. The value of this object is a non-localized (English only) string, generated by SSPR, including the *errorCode* and the error *key*.

**data** *UTF-8 string*

This object will be present if the root *error* object value is *false*.

The associated string value represents SSPR *signed* and *encrypted* request data content.

It is expected that this string will then accompany an SSPR request, such as the creation of a new user, using the *signedForm* value in the query string of the request.

The returned string expires after a period of time.

## Examples

### Curl Example #1

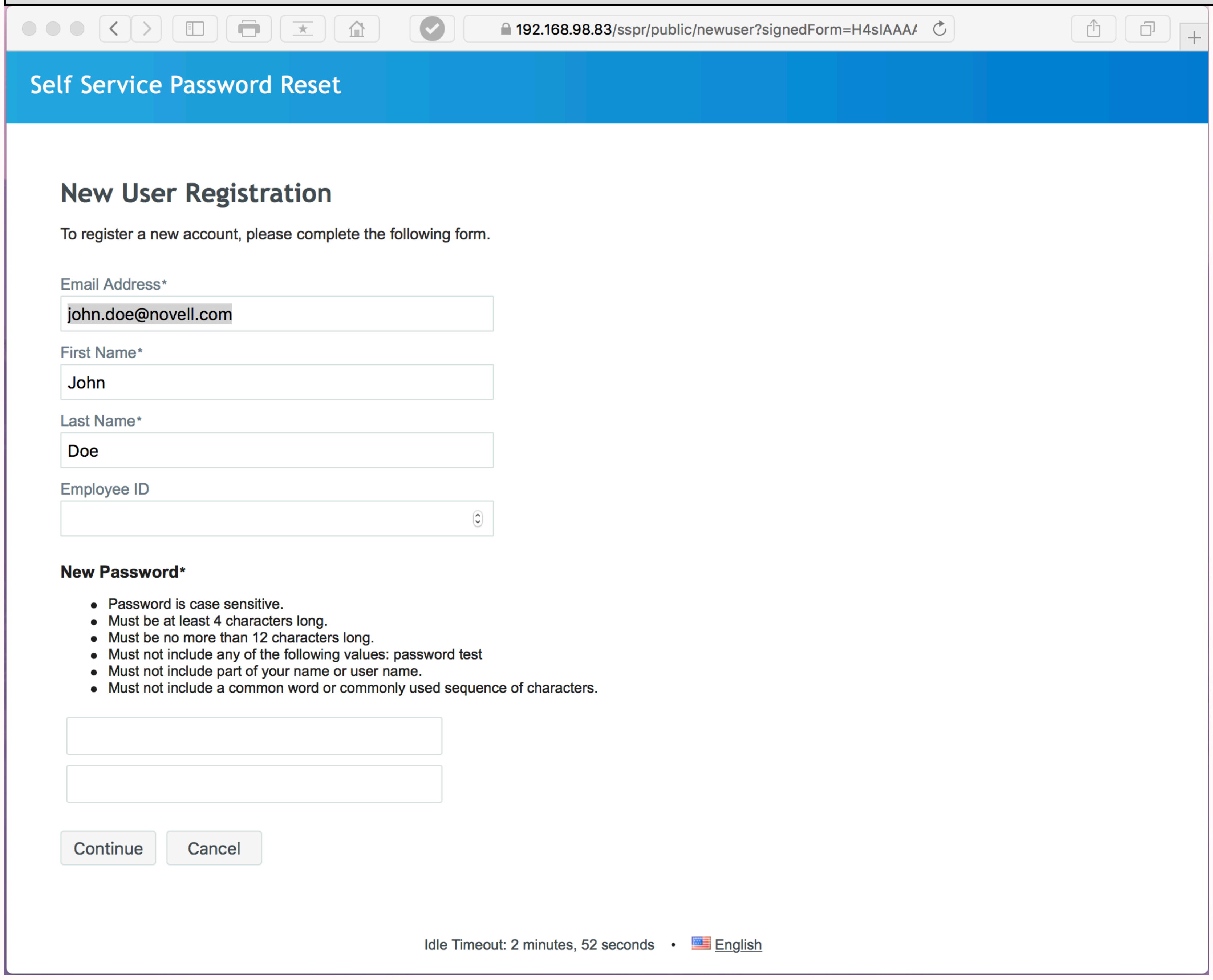
Form parameters passed as request content in json format.

```
curl \
    -v \
    -X POST \
    -H "Accept-Language: en" \
    -H "Accept:application/json" \
    -H "Content-Type:application/json" \
    -u "SSPR-REST-EWSS:password" \
    "https://192.168.98.83/sspr/public/rest/signing/form" \
    -d '{"givenName":"John","sn":"Doe","mail":"john.doe@novell.com"}'
Note: Unnecessary use of -X or --request, POST is already inferred.
*   Trying 192.168.98.83...
* TCP_NODELAY set
* Connected to 192.168.98.83 (192.168.98.83) port 443 (#0)
* WARNING: using IP address, SNI is being disabled by the OS.
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
* Server certificate: sspr85.mymac.com
* Server auth using Basic with user 'SSPR-REST-EWSS'
> POST /sspr/public/rest/signing/form HTTP/1.1
> Host: 192.168.98.83
> Authorization: Basic U1NQUi1SRVNULUVXU1M6cGFzc3dvcmQ=
> User-Agent: curl/7.54.0
> Accept-Language: en
> Accept:application/json
> Content-Type:application/json
> Content-Length: 60
>
* upload completely sent off: 60 out of 60 bytes
< HTTP/1.1 200
< Vary: Accept-Encoding
< X-SSPR-Noise: Ig5vcbqPIWwMI0mHfegW9SwIcVzNpp0M
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1
< X-SSPR-Instance: D240DAD4AA63F70B
< Server: server
< X-Frame-Options: DENY
< Cache-Control: no-cache, no-store, must-revalidate, proxy-revalidate
< Content-Type: application/json;charset=UTF-8
< Content-Length: 285
< Date: Wed, 04 Apr 2018 21:57:05 GMT
<
{
  "error": false,
  "errorCode": 0,
  "data": "H4sIAAAAAAAAAAGVAGr_UFdNLkdDTTEQVB6SueLSFGD0rCV1UgH9MHCrhsIY9xkp1M591zltQkqvKwg0zafaFShEjQ9jy4v
}
* Connection #0 to host 192.168.98.83 left intact
```

Subsequent use of the returned data to create a new user can be demonstrated by entering the the data from the response into a browser.

Example browser location URL:

```
https://192.168.98.83/sspr/public/newuser?signedForm=H4sIAAAAAAAAAAGVAGr_UFdNLkdDTTEQVB6SueLSFGD0rCV1UgH9M
```



192.168.98.83/sspr/public/newuser?signedForm=H4sIAAA
Self Service Password Reset

## New User Registration

To register a new account, please complete the following form.

| | |
|---|---|
| Email Address* | john.doe@novell.com |
| First Name* | John |
| Last Name* | Doe |
| Employee ID | |

**New Password***

- Password is case sensitive.
- Must be at least 4 characters long.
- Must be no more than 12 characters long.
- Must not include any of the following values: password test
- Must not include part of your name or user name.
- Must not include a common word or commonly used sequence of characters.

### Curl Example #2

Form parameters passed as request content in *application/x-www-form-urlencoded* format.

* Known Issue: 1087993

```
curl \
    -v \
    -X POST \
    -H "Accept-Language: en" \
    -H "Accept: application/json" \
    -H "Content-Type: application/x-www-form-urlencoded" \
    -u "SSPR-REST-EWSS:password" \
    "https://192.168.98.83/sspr/public/rest/signing/form" \
    -d 'givenName=John&sn=Doe&mail=john%2Edoe%40novell%2Ecom'
Note: Unnecessary use of -X or --request, POST is already inferred.
*   Trying 192.168.98.83...
* TCP_NODELAY set
* Connected to 192.168.98.83 (192.168.98.83) port 443 (#0)
* WARNING: using IP address, SNI is being disabled by the OS.
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
* Server certificate: sspr85.mymac.com
* Server auth using Basic with user 'SSPR-REST-EWSS'
> POST /sspr/public/rest/signing/form HTTP/1.1
> Host: 192.168.98.83
> Authorization: Basic U1NQUi1SRVNULUVXU1M6cGFzc3dvcmQ=
> User-Agent: curl/7.54.0
> Accept-Language: en
> Accept: application/json
> Content-Type: application/x-www-form-urlencoded
> Content-Length: 52
>
* upload completely sent off: 52 out of 52 bytes
< HTTP/1.1 200
< Vary: Accept-Encoding
< X-SSPR-Noise: nVav0BXAp4ytV0R4Xl
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1
< X-SSPR-Instance: D240DAD4AA63F70B
< Server: server
< X-Frame-Options: DENY
< Cache-Control: no-cache, no-store, must-revalidate, proxy-revalidate
< Content-Type: application/json;charset=UTF-8
< Content-Length: 188
< Date: Tue, 03 Apr 2018 22:35:45 GMT
<
{
  "error": true,
  "errorCode": 5015,
  "errorMessage": "An error has occurred.  If this error occurs repeatedly please contact your help desk."
  "errorDetail": "5015 ERROR_UNKNOWN"
}
* Connection #0 to host 192.168.98.83 left intact
```

# REST Service: /sspr/public/rest/statistics

## GET Method

Read SSPR application statistics.

### Request Headers

**Authorization** *Basic Authentication Method*

Required. Specifies credentials to authorize the REST caller. This is used by SSPR to validate the caller's rights to perform this action. Use a named secret *username:secret* value in SSPR's *Configuration Editor*:

Settings → Web Services → REST Services → Web Service Non-LDAP Users and Passwords

**Accept-Language** *Language code*

The request will be processed in the context of the specified language code. For more detail on SSPR locale codes, see Appendix F: Locales (Languages) and Flags.

**Accept** *Document protocol type*

Protocol type values:
*application/json*
*/* (Content will be returned in the default *application/json* format.)

### Request Query String Parameters

**statKey** *Event Statistic Key* * Known Issue: 1089267

Key of statistic to retrieve. (See Appendix B: Event Statistics).

**days** *number of days to report.*

Optional. Number of days to include in the response *data.nameData* element.

If this parameter is not specified, all available days will be returned in the response *data.nameData* element.

### Request Content

None

### Response Status Codes

**200 OK** *Successful* The call was successfully serviced by SSPR.

## Response Headers

This list of HTTP response headers is not complete. Headers that are not listed here have little, or no impact for SSPR REST application developers.

**Content-Type** *application/json;charset=UTF-8*

Indicates the type of the data in the response content body.

**Content-Length** *bytes*

Number of octets (8-bit bytes) found in the *response content*.

**Date** *HTTP-date*

The date and time that the response originated in *HTTP-date* format as defined by RFC 7231 Date/Time Formats.

**Server** *server*

Historically, identifies the server software implemented to respond to the REST request. However, it represents an unnecessary leak of information of information about the platform. Rather than eliminating the header, the response includes this header, but with the innocuous value of *server*.

**Vary** *Accept-Encoding*

Informs cache stores that this response varies based on the value of the *Accept-Encoding* header.

**X-SSPR-Instance** *Name of the SSPR application instance.*

All requests from a specific SSPR server will have this value in common. The value of this header is specified in the SSPR *Configuration Editor*:

Settings→Application→Applications→Instance Name

## Response Content

**error** *Boolean*

Indicates if the REST service experienced an error servicing the request.

**errorCode** *Integer*

If the root *error* object value is *false*, the value of this object will be 0.

If the root *error* object value is *true*, the value of this object will be a numeric SSPR error code. The meaning of the error code can be found in Appendix A: Errors.

Common SSPR *errorCode* Values:

| | | |
|---|---|---|
| **0** | *Success* | Operation completed successfully. |
| **5015** | *ERROR_UNKNOWN* | An error has occurred. If this error occurs repeatedly please contact your help desk. |

**successMessage** *UTF-8 string*

This object is present when the *error* object is *false*. The value of this object is a localized string, generated by SSPR, indicating that the operation has succeeded.

**errorMessage** *UTF-8 string*

This object is present when the *error* object is *true*. The value of this object is a localized string, generated by SSPR, indicating why the operation has failed.

**errorDetail** *UTF-8 string*

This object is present when the *error* object is *true*. The value of this object is a non-localized (English only) string, generated by SSPR, including the *errorCode* and the error *key*.

**data** *object*

This object will be present if the root *error* object value is *false*.

Contains child objects, each child representing a component of the SSPR application's statistics. Child objects are include:

**EPS** *object*

Contains child objects, each child representing a component of the SSPR application's *Events Per Second* statistics. Child objects are listed in Appendix C: Events Per Second Statistics.

**nameData** *array object*

This object is present when a valid *statKey* query string parameter value is specified.

Each element of this array object holds one day of the requested statistic specified by the *statKey* parameter. * Known Issue: 1089267

By default, all days are returned, but this can be reduced by specifying the *days* parameter.

**keyData** *object*

This object is present when no query string parameters are specified.

This object lists each SSPR event statistic and with the sum total of those events that have occurred since the beginning of the current *Coordinated Universal Time* (UTC) 24-hour day.

Child objects are listed in Appendix B: Event Statistics.

## Examples

### Curl Example #1

Parameters in query string.

```
curl \
    -v \
    -H "Accept:application/json" \
    -H "Accept-Language:en" \
    -u "SSPR-REST-EWSS:password" \
    "https://192.168.98.83/sspr/public/rest/statistics?days=14&statName=PASSWORD_CHANGES"
*   Trying 192.168.98.83...
* TCP_NODELAY set
* Connected to 192.168.98.83 (192.168.98.83) port 443 (#0)
* WARNING: using IP address, SNI is being disabled by the OS.
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
* Server certificate: sspr85.mymac.com
* Server auth using Basic with user 'SSPR-REST-EWSS'
> GET /sspr/public/rest/statistics?days=14&statName=PASSWORD_CHANGES HTTP/1.1
> Host: 192.168.98.83
> Authorization: Basic U1NQUi1SRVNULUVXU1M6cGFzc3dvcmQ=
> User-Agent: curl/7.54.0
> Accept:application/json
> Accept-Language:en
>
< HTTP/1.1 200
< Vary: Accept-Encoding
< X-SSPR-Noise: zOvEwS3Z2iAuKDsCaWeP0206bV6YZVvHeEQxw
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1
< X-SSPR-Instance: D240DAD4AA63F70B
< Server: server
< X-Frame-Options: DENY
< Cache-Control: no-cache, no-store, must-revalidate, proxy-revalidate
< Content-Type: application/json;charset=UTF-8
< Content-Length: 1358
< Date: Thu, 05 Apr 2018 18:43:33 GMT
<
{
  "error": false,
  "errorCode": 0,
  "data": {
    "EPS": {
      "AUTHENTICATION_DAY": "0.001",
      "AUTHENTICATION_HOUR": "0.001",
      "AUTHENTICATION_MINUTE": "0.001",
      "DB_READS_DAY": "0.000",
      "DB_READS_HOUR": "0.000",
      "DB_READS_MINUTE": "0.000",
      "DB_WRITES_DAY": "0.000",
      "DB_WRITES_HOUR": "0.000",
      "DB_WRITES_MINUTE": "0.000",
```

### Curl Example #2

No parameters.

* Known Issue: 1089394

```
curl \
    -v \
    -H "Accept:application/json" \
    -H "Accept-Language:en" \
    -u "SSPR-REST-EWSS:password" \
    "https://192.168.98.83/sspr/public/rest/statistics"
*   Trying 192.168.98.83...
* TCP_NODELAY set
* Connected to 192.168.98.83 (192.168.98.83) port 443 (#0)
* WARNING: using IP address, SNI is being disabled by the OS.
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
* Server certificate: sspr.example.com
* Server auth using Basic with user 'SSPR-REST-EWSS'
> GET /sspr/public/rest/statistics HTTP/1.1
> Host: 192.168.98.83
> Authorization: Basic U1NQUi1SRVNULUVXU1M6cGFzc3dvcmQ=
> User-Agent: curl/7.54.0
> Accept:application/json
> Accept-Language:en
>
< HTTP/1.1 200
< Vary: Accept-Encoding
< X-SSPR-Noise: rnq5Eirrc6CGCA5QpkaSeuzi76M6RBJKv5FdmE7YFVXJxzZyDtaLV56Jk37YAnJbMxXEBE6oNWrX5kOZHJoPXhNS
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1
< X-SSPR-Instance: 894B3362E2E382A5
< Server: server
< X-Frame-Options: DENY
< Cache-Control: no-cache, no-store, must-revalidate, proxy-revalidate
< Content-Type: application/json;charset=UTF-8
< Content-Length: 3641
< Date: Thu, 12 Apr 2018 15:31:08 GMT
<
{
  "error": false,
  "errorCode": 0,
  "data": {
    "EPS": {
      "AUTHENTICATION_DAY": "0.000",
      "AUTHENTICATION_HOUR": "0.000",
      "AUTHENTICATION_MINUTE": "0.000",
      "DB_READS_DAY": "0.000",
      "DB_READS_HOUR": "0.000",
      "DB_READS_MINUTE": "0.000",
      "DB_WRITES_DAY": "0.000",
      "DB_WRITES_HOUR": "0.000",
      "DB_WRITES_MINUTE": "0.000",
      "INTRUDER_ATTEMPTS_DAY": "0.000",
      "INTRUDER_ATTEMPTS_HOUR": "0.000",
```

# REST Service: /sspr/public/rest/status

## GET Method

Read user status data.

### Request Headers

**Authorization** *Basic Authentication Method*

Required. Specifies credentials to authorize the REST caller. This is used by SSPR to validate the caller's rights to perform this action.

Use a named secret *username:secret* value in SSPR's *Configuration Editor*:

Settings → Web Services → REST Services → Web Service Non-LDAP Users and Passwords

**Accept-Language** *Language code*

The request will be processed in the context of the specified language code. For more detail on SSPR locale codes, see Appendix F: Locales (Languages) and Flags.

**Accept** *Document protocol type*

Protocol type values:
*application/json*

### Request Query String Parameters

**username** *username or userDN*

Optional. Target Username or userDN of the REST call. If this parameter is not specified, the user specified by the *Authorization* header is assumed.

### Request Content

None

### Response Status Codes

**200 OK** *Successful* The call was successfully serviced by SSPR.

## Response Headers

This list of HTTP response headers is not complete. Headers that are not listed here have little, or no impact for SSPR REST application developers.

**Content-Type** *application/json;charset=UTF-8*

Indicates the type of the data in the response content body.

**Content-Length** *bytes*

Number of octets (8-bit bytes) found in the *response content*.

**Date** *HTTP-date*

The date and time that the response originated in *HTTP-date* format as defined by RFC 7231 Date/Time Formats.

**Server** *server*

Historically, identifies the server software implemented to respond to the REST request. However, it represents an unnecessary leak of information of information about the platform. Rather than eliminating the header, the response includes this header, but with the innocuous value of *server*.

**Vary** *Accept-Encoding*

Informs cache stores that this response varies based on the value of the *Accept-Encoding* header.

**X-SSPR-Instance** *Name of the SSPR application instance.*

All requests from a specific SSPR server will have this value in common. The value of this header is specified in the SSPR *Configuration Editor*:

Settings→Application→Applications→Instance Name

## Response Content

**error** *Boolean*

Indicates if the REST service experienced an error servicing the request.

**errorCode** *Integer*

If the root *error* object value is *false*, the value of this object will be 0.

If the root *error* object value is *true*, the value of this object will be a numeric SSPR error code. The meaning of the error code can be found in Appendix A: Errors.

Common SSPR *errorCode* Values:

| | | |
|---|---|---|
| **0** | *Success* | Operation completed successfully. |
| **5015** | *ERROR_UNKNOWN* | An error has occurred. If this error occurs repeatedly please contact your help desk. |

**successMessage** *UTF-8 string*

This object is present when the *error* object is *false*. The value of this object is a localized string, generated by SSPR, indicating that the operation has succeeded.

**errorMessage** *UTF-8 string*

This object is present when the *error* object is *true*. The value of this object is a localized string, generated by SSPR, indicating why the operation has failed.

**errorDetail** *UTF-8 string*

This object is present when the *error* object is *true*. The value of this object is a non-localized (English only) string, generated by SSPR, including the *errorCode* and the error *key*.

**data** *object*

This object will be present if the root *error* object value is *false*.

Contains child objects, each child representing a component of the SSPR user's status. Child objects may include those found in Appendix H: User Information, but may also be absent if the specific child object does not have an assigned value.

**passwordPolicy** *object*

This object will be present if the root *error* object value is *false*.

Contains child objects, each child representing an attribute of the SSPR password policy associated with the target user. The attributes are defined in Appendix G: Password Policy Attributes.

**passwordRules** *object*

When users change their password using SSPR, a *change password* screen is presented to the user. On this screen, SSPR lists several *password rules* that the user must consider when selecting a new password.

The *password rules* are internally calculated by SSPR, and are localized according to the *Accept-Language* header of the REST request.

The SSPR generated *password rules* are listed in the reply of this REST response to allow external developers to implement their own *change password* user interface, and allow that interface to list the same *password rules* as SSPR's *change password* screen.

The list consists of an array of localized UTF-8 strings. Example:

```
"passwordRules":
 [
 "Password is case sensitive.",
 "Must be at least 4 characters long.",
 "Must be no more than 12 characters long.",
 "Must not include any of the following values: password test",
 "Must not include part of your name or user name.",
 "Must not include a common word or commonly used sequence of characters."
 ]
```

## Examples

### Curl Example #1

Parameter *username* in query string.

```
curl \
    -v \
    -H "Accept:application/json" \
    -H "Accept-Language:en" \
    -u "SSPR-REST-EWSS:password" \
    "https://192.168.98.83/sspr/public/rest/status?username=sspr-testuser-01"
*   Trying 192.168.98.83...
* TCP_NODELAY set
* Connected to 192.168.98.83 (192.168.98.83) port 443 (#0)
* WARNING: using IP address, SNI is being disabled by the OS.
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
* Server certificate: sspr85.mymac.com
* Server auth using Basic with user 'SSPR-REST-EWSS'
> GET /sspr/public/rest/status?username=sspr-testuser-01 HTTP/1.1
> Host: 192.168.98.83
> Authorization: Basic U1NQUi1SRVNULUVXU1M6cGFzc3dvcmQ=
> User-Agent: curl/7.54.0
> Accept:application/json
> Accept-Language:en
>
< HTTP/1.1 200
< Vary: Accept-Encoding
< X-SSPR-Noise: NdoClvSQueYJcTs39rzdoPY5VN4CpZRh
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1
< X-SSPR-Instance: D240DAD4AA63F70B
< Server: server
< X-Frame-Options: DENY
< Cache-Control: no-cache, no-store, must-revalidate, proxy-revalidate
< Content-Type: application/json;charset=UTF-8
< Content-Length: 2394
< Date: Tue, 10 Apr 2018 20:40:18 GMT
<
{
  "error": false,
  "errorCode": 0,
  "data": {
    "userDN": "cn=sspr-testuser-01,ou=users,o=sles12",
    "ldapProfile": "default",
    "userID": "sspr-testuser-01",
    "userEmailAddress": "genious@novell.com",
    "passwordLastModifiedTime": "2018-04-03T16:35:42Z",
    "lastLoginTime": "2018-03-22T16:49:58Z",
    "requiresNewPassword": false,
    "requiresResponseConfig": true,
    "requiresUpdateProfile": false,
    "requiresOtpConfig": false,
```

# REST Service: /sspr/public/rest/verifyotp

## POST Method

Validate a *One-Time Password* against a user's stored secret.

Parameters may be specified as query string arguments, or as *application/json* formatted request content.

### Request Headers

**Authorization** *Basic Authentication Method*

Required. Specifies credentials to authorize the REST caller. This is used by SSPR to validate the caller's rights to perform this action. Use a named secret *username:secret* value in SSPR's *Configuration Editor*:

Settings → Web Services → REST Services → Web Service Non-LDAP Users and Passwords

**Accept** *Document protocol type*

Protocol type values:
*application/json*

**Content-Type** *Document protocol type*

***NOTE:*** *This header is required, even when all parameters are specified in the query string and there is no content following the HTTP headers section.*

Protocol type values:
*application/json*

### Request Query String Parameters

**username** *username or ldap DN*

Optional. Target Username or userDN of the REST call. If this parameter is not specified, the user specified by the *Authorization* header is assumed.

**token** *password string*

The *One-Time Password* to be verified.

### Request Content

**username** *username or ldap DN*

Optional. Target Username or userDN of the REST call. If this parameter is not specified, the user specified by the *Authorization* header is assumed.

**token** *password string*

The *One-Time Password* to be verified.

## Response Status Codes

**200 OK** *Successful* The call was successfully serviced by SSPR.

## Response Headers

This list of HTTP response headers is not complete. Headers that are not listed here have little, or no impact for SSPR REST application developers.

**Content-Type** *application/json;charset=UTF-8*

Indicates the type of the data in the response content body.

**Content-Length** *bytes*

Number of octets (8-bit bytes) found in the *response content*.

**Date** *HTTP-date*

The date and time that the response originated in *HTTP-date* format as defined by RFC 7231 Date/Time Formats.

**Server** *server*

Historically, identifies the server software implemented to respond to the REST request. However, it represents an unnecessary leak of information of information about the platform. Rather than eliminating the header, the response includes this header, but with the innocuous value of *server*.

**Vary** *Accept-Encoding*

Informs cache stores that this response varies based on the value of the *Accept-Encoding* header.

**X-SSPR-Instance** *Name of the SSPR application instance.*

All requests from a specific SSPR server will have this value in common. The value of this header is specified in the SSPR *Configuration Editor*:

Settings→Application→Applications→Instance Name

## Response Content

**error** *Boolean*

Indicates if the REST service experienced an error servicing the request.

**errorCode** *Integer*

If the root *error* object value is *false*, the value of this object will be 0.

If the root *error* object value is *true*, the value of this object will be a numeric SSPR error code. The meaning of the error code can be found in Appendix A: Errors.

Common SSPR *errorCode* Values:

| | | |
|---|---|---|
| **0** | *Success* | Operation completed successfully. |
| **5015** | *ERROR_UNKNOWN* | An error has occurred. If this error occurs repeatedly please contact your help desk. |

**successMessage** *UTF-8 string*

This object is present when the *error* object is *false*. The value of this object is a localized string, generated by SSPR, indicating that the operation has succeeded.

**errorMessage** *UTF-8 string*

This object is present when the *error* object is *true*. The value of this object is a localized string, generated by SSPR, indicating why the operation has failed.

**errorDetail** *UTF-8 string*

This object is present when the *error* object is *true*. The value of this object is a non-localized (English only) string, generated by SSPR, including the *errorCode* and the error *key*.

**data** *boolean*

This object is present when the *error* object is *false*. The value of this object indicates that the *One-Time Password*, supplied by the *token* parameter, passed validation.

## Examples

### Curl Example #1

Parameters *username* and *token* in content as json data.

```
curl \
    -v \
    -H "Accept:application/json" \
    -H "Accept-Language:en" \
    -H "Content-Type:application/json" \
    -u "SSPR-REST-EWSS:password" \
    "https://192.168.98.83/sspr/public/rest/verifyotp" \
    -d'{"username":"sspr-testuser-01","token":123456}'
*   Trying 192.168.98.83...
* TCP_NODELAY set
* Connected to 192.168.98.83 (192.168.98.83) port 443 (#0)
* WARNING: using IP address, SNI is being disabled by the OS.
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
* Server certificate: sspr.example.com
* Server auth using Basic with user 'SSPR-REST-EWSS'
> POST /sspr/public/rest/verifyotp HTTP/1.1
> Host: 192.168.98.83
> Authorization: Basic U1NQUi1SRVNULUVXU1M6cGFzc3dvcmQ=
> User-Agent: curl/7.54.0
> Accept:application/json
> Accept-Language:en
> Content-Type:application/json
> Content-Length: 46
>
* upload completely sent off: 46 out of 46 bytes
< HTTP/1.1 200
< Vary: Accept-Encoding
< X-SSPR-Noise: lxVVKGD5mBM6O1HgHX5ijpUvg7k7g5Xqa8aj0os4brhT33ARrjb2hBK15iBzICVGYmbW4TVFnnIuaVqq9iudT
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1
< X-SSPR-Instance: 894B3362E2E382A5
< Server: server
< X-Frame-Options: DENY
< Cache-Control: no-cache, no-store, must-revalidate, proxy-revalidate
< Content-Type: application/json;charset=UTF-8
< Content-Length: 126
< Date: Thu, 12 Apr 2018 22:21:50 GMT
<
{
  "error": false,
  "errorCode": 0,
  "successMessage": "The operation has been successfully completed.",
  "data": false
}
* Connection #0 to host 192.168.98.83 left intact
```

### Curl Example #2

Parameters *username* and *token* in query string.

```
curl \
    -v \
    -X POST \
    -H "Accept: application/json" \
    -H "Accept-Language: en" \
    -H "Content-Type: application/json" \
    -u "SSPR-REST-EWSS:password" \
    "https://192.168.98.83/sspr/public/rest/verifyotp?username=sspr-testuser-01&token=123456"
*   Trying 192.168.98.83...
* TCP_NODELAY set
* Connected to 192.168.98.83 (192.168.98.83) port 443 (#0)
* WARNING: using IP address, SNI is being disabled by the OS.
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
* Server certificate: sspr.example.com
* Server auth using Basic with user 'SSPR-REST-EWSS'
> POST /sspr/public/rest/verifyotp?username=sspr-testuser-01&token=123456 HTTP/1.1
> Host: 192.168.98.83
> Authorization: Basic U1NQUi1SRVNULUVXU1M6cGFzc3dvcmQ=
> User-Agent: curl/7.54.0
> Accept: application/json
> Accept-Language: en
> Content-Type: application/json
>
< HTTP/1.1 200
< Vary: Accept-Encoding
< X-SSPR-Noise: uFyiKK33WHHw4nHfa6N153G8l11hLwdLfANrGz42CdTgglkm6xn27XwZ2jsgLBviLHR8SuamfWLxCTgmq0q
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1
< X-SSPR-Instance: 894B3362E2E382A5
< Server: server
< X-Frame-Options: DENY
< Cache-Control: no-cache, no-store, must-revalidate, proxy-revalidate
< Content-Type: application/json;charset=UTF-8
< Content-Length: 126
< Date: Thu, 12 Apr 2018 22:50:53 GMT
<
{
  "error": false,
  "errorCode": 0,
  "successMessage": "The operation has been successfully completed.",
  "data": false
}
* Connection #0 to host 192.168.98.83 left intact
```

# REST Service: /sspr/public/rest/verifyresponses

## POST Method

Validate supplied challenge response answers against a user's stored responses.

*Note this service will not work properly if the user's responses are stored only in the NMAS repository.*

The username parameter may be specified as query a string argument, or as part of the *application/json* formatted request content.

### Request Headers

**Authorization** *Basic Authentication Method*

Required. Specifies credentials to authorize the REST caller. This is used by SSPR to validate the caller's rights to perform this action.

**Accept-Language** *Language code*

The request will be processed in the context of the specified language code. For more detail on SSPR locale codes, see Appendix F: Locales (Languages) and Flags.

**Accept** *Document protocol type*

Protocol type values:
*application/json*

**Content-Type** *Document protocol type*

Protocol type values:
*application/json*

### Request Query String Parameters

**username** *username or userDN*

Optional. Target Username or userDN of the REST call. If this parameter is not specified, the user specified by the *Authorization* header is assumed.

### Request Content

**username** *username or userDN*

Optional. Target Username or userDN of the REST call. If this parameter is not specified, the user specified by the *Authorization* header is assumed.

**challenges** *array object*

Array of objects containing challenge/response questions and answers to be verified.

Retrieve challenge objects using the challenges service to discover the proper object formatting. The question object data must match precisely the question object received from the challenges service so that the answer can be applied to the correct corresponding question. This includes each parameter of the question object.

**challengeText** *UTF-8 string*

Text of the challenge/response question, to which an *answer* will be supplied to verify.

**minLength** *Integer*

The minimum length of the user-supplied response string.

**maxLength** *Integer*

The maximum length of the user-supplied response string.

**adminDefined** *boolean*

Indicates if value of the *challengeText* attribute, the challenge question, was supplied by the system administrator, or if it was supplied by the user.

**required** *boolean*

Indicates if this challenge is *required*, or *random*.

Users must correctly answer all *required* challenges to be validated.

Random challenges are placed in a pool; where the user may be required to correctly answer a specified number of them in order to be validated.

**answer** *object*

**answer** *string*

Text of the challenge/response answer to verify.

## Response Status Codes

**200 OK** *Successful* The call was successfully serviced by SSPR.

## Response Headers

This list of HTTP response headers is not complete. Headers that are not listed here have little, or no impact for SSPR REST application developers.

**Content-Type** *application/json;charset=UTF-8*

Indicates the type of the data in the response content body.

**Content-Length** *bytes*

Number of octets (8-bit bytes) found in the *response content*.

**Date** *HTTP-date*

The date and time that the response originated in *HTTP-date* format as defined by RFC 7231 Date/Time Formats.

**Server** *server*

Historically, identifies the server software implemented to respond to the REST request. However, it represents an unnecessary leak of information of information about the platform. Rather than eliminating the header, the response includes this header, but with the innocuous value of *server*.

**Vary** *Accept-Encoding*

Informs cache stores that this response varies based on the value of the *Accept-Encoding* header.

**X-SSPR-Instance** *Name of the SSPR application instance.*

All requests from a specific SSPR server will have this value in common. The value of this header is specified in the SSPR *Configuration Editor*:

Settings→Application→Applications→Instance Name

## Response Content

**error** *Boolean*

Indicates if the REST service experienced an error servicing the request.

**errorCode** *Integer*

If the root *error* object value is *false*, the value of this object will be 0.

If the root *error* object value is *true*, the value of this object will be a numeric SSPR error code. The meaning of the error code can be found in Appendix A: Errors.

Common SSPR *errorCode* Values:

| | | |
|---|---|---|
| **0** | *Success* | Operation completed successfully. |
| **5015** | *ERROR_UNKNOWN* | An error has occurred. If this error occurs repeatedly please contact your help desk.<br>May be caused when the target user has not registered challenge/response data. * Known Issue: 1084214 |

**successMessage** *UTF-8 string*

This object is present when the *error* object is *false*. The value of this object is a localized string, generated by SSPR, indicating that the operation has succeeded.

**errorMessage** *UTF-8 string*

This object is present when the *error* object is *true*. The value of this object is a localized string, generated by SSPR, indicating why the operation has failed.

**errorDetail** *UTF-8 string*

This object is present when the *error* object is *true*. The value of this object is a non-localized (English only) string, generated by SSPR, including the *errorCode* and the error *key*.

**data** *boolean*

This object is present when the *error* object is *false*. The value of this object indicates if the supplied *challenges* passed validation.

## Examples

### Curl Example #1

Parameters *username* and *list of challenges* in content as json data.

```
curl \
    -v \
    -H "Accept: application/json" \
    -H "Accept-Language: en" \
    -H "Content-Type: application/json" \
    -u "SSPR-REST-EWSS:password" \
    "https://192.168.98.83/sspr/public/rest/verifyresponses" \
    -d'@verifyresponses_01.json'
*   Trying 192.168.98.83...
* TCP_NODELAY set
* Connected to 192.168.98.83 (192.168.98.83) port 443 (#0)
* WARNING: using IP address, SNI is being disabled by the OS.
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
* Server certificate: sspr85.mymac.com
* Server auth using Basic with user 'SSPR-REST-EWSS'
> POST /sspr/public/rest/verifyresponses HTTP/1.1
> Host: 192.168.98.83
> Authorization: Basic U1NQUi1SRVNULUVXU1M6cGFzc3dvcmQ=
> User-Agent: curl/7.54.0
> Accept: application/json
> Accept-Language: en
> Content-Type: application/json
> Content-Length: 394
>
* upload completely sent off: 394 out of 394 bytes
```

File: verifyresponses_01.json

```
{
"username":"sspr-testuser-01",
"challenges":
 [
  {
  "challengeText":"Who is your favorite author?",
  "minLength":4,
  "maxLength":200,
  "adminDefined":true,
  "required":false,
  "answer":
   {
   "answerText":"wiyfa"
   }
  },
  {
  "challengeText":"What is your favorite food?"
```

Curl Example #2

The *username* specified in query string, and *list of challenges* specified in json file.

```
curl \
    -v \
    -H "Accept:application/json" \
    -H "Accept-Language:en" \
    -H "Content-Type:application/json" \
    -u "SSPR-REST-EWSS:password" \
    "https://192.168.98.83/sspr/public/rest/verifyresponses?username=sspr-testuser-01" \
    -d'@verifyresponses_02.json'
*   Trying 192.168.98.83...
* TCP_NODELAY set
* Connected to 192.168.98.83 (192.168.98.83) port 443 (#0)
* WARNING: using IP address, SNI is being disabled by the OS.
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
* Server certificate: sspr85.mymac.com
* Server auth using Basic with user 'SSPR-REST-EWSS'
> POST /sspr/public/rest/verifyresponses?username=sspr-testuser-01 HTTP/1.1
> Host: 192.168.98.83
> Authorization: Basic U1NQUi1SRVNULUVXU1M6cGFzc3dvcmQ=
> User-Agent: curl/7.54.0
> Accept:application/json
> Accept-Language:en
> Content-Type:application/json
> Content-Length: 364
>
```

File: verifyresponses_02.json

```
{
"challenges":
 [
  {
  "challengeText":"Who is your favorite author?",
  "minLength":4,
  "maxLength":200,
  "adminDefined":true,
  "required":false,
  "answer":
   {
   "answerText":"wiyfa"
   }
  },
  {
  "challengeText":"What is your favorite food?",
  "minLength":4
```

# SSPR as a REST Client

Settings → Web Services → REST Clients

## Introduction

The SSPR application can be configured to make REST calls to external hosts. External hosts (supplied by the SSPR customer) can be programmed (by the SSPR customer) supply RESTful interfaces that serve configured SSPR requests.

# REST Client: Remote Form Data Service

SSPR can facilitate a non-LDAP data source for validation and storage of form (user account) data. This feature might be implemented to store sensitive data in a corporate database, rather than the LDAP directory. Examples might include US Social Security numbers, Credit card numbers, etc.

To implement this feature, an external HTTP/REST server must be supplied which is capable of intercepting the REST client requests from SSPR. This external REST service is not supplied by with SSPR. Rather, it is engineered in-house by SSPR customers.

Currently, the *Remote Form Data Service* feature is only implemented in association with SSPR's *New User Registration* form.

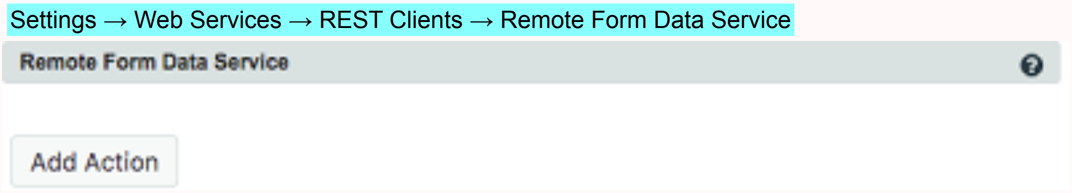
When properly configured, SSPR will issue REST API calls to a customer-supplied host to validate each field, as well as when the final (accepted) form is submitted.

The specific REST calls include:

- Validation form fields *while the user is filling out the form*.
- Request to store/write the form data.

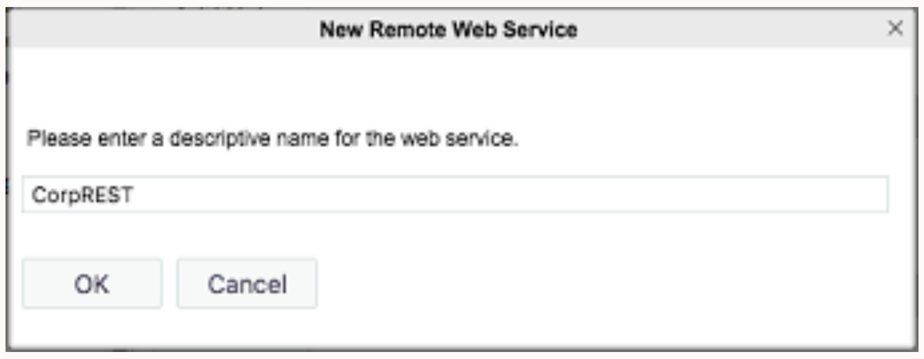
## Configure Self Service Password Reset for Remote Form Data Service

Use the SSPR *configuration editor* to configure of SSPR to interact with an external HTTP/REST *form data* server:


Settings → Web Services → REST Clients → Remote Form Data Service

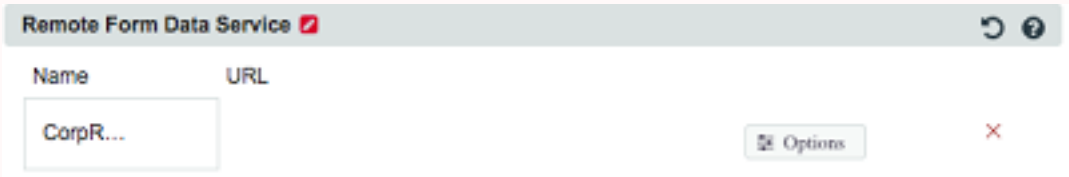
Click *Add Action* to specify an external HTTP/REST server.

In this example, a new web service configuration named *CorpREST* is specified.

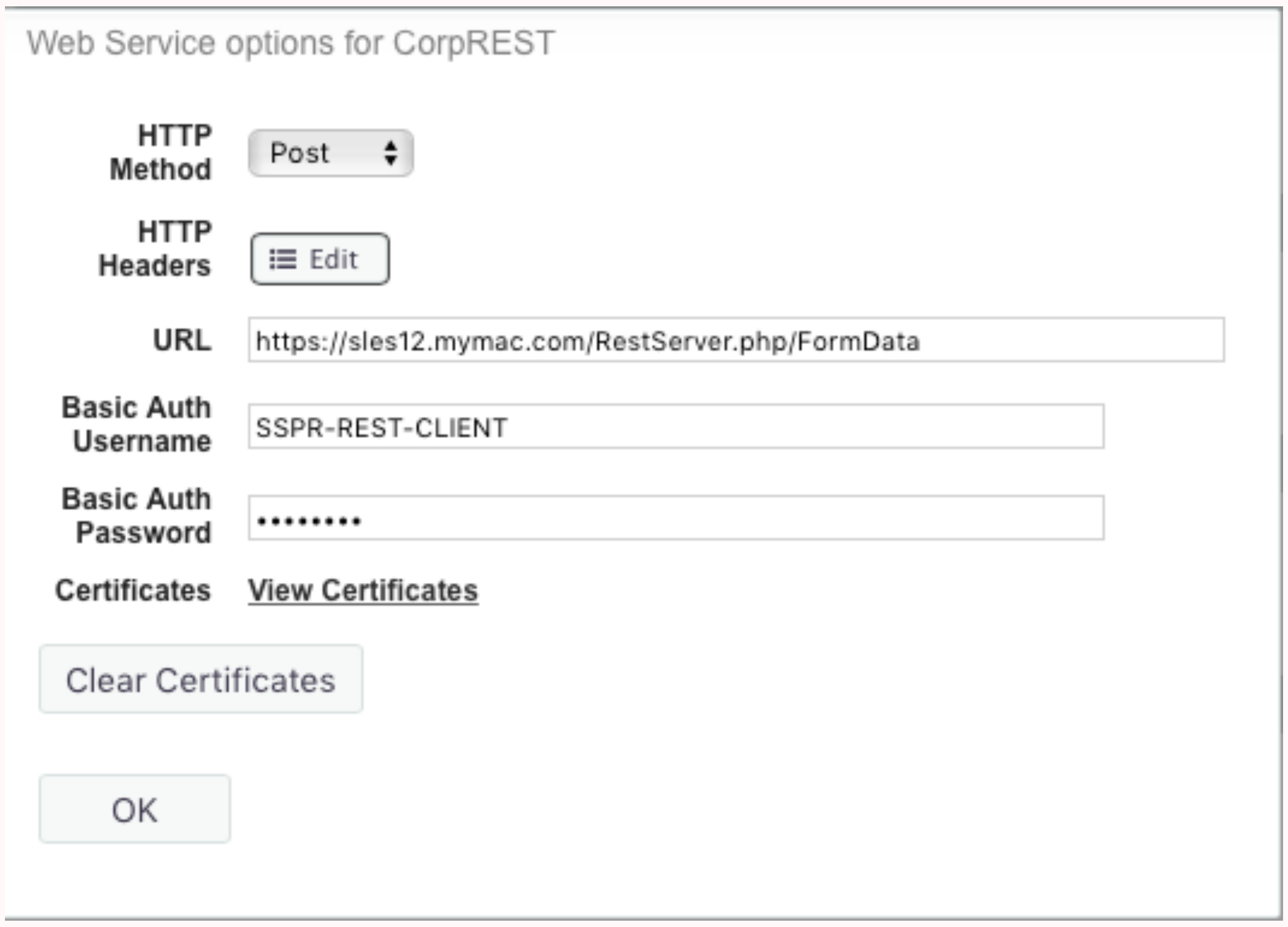

Please enter a descriptive name for the web service.

CorpREST

Click *OK* to create the new web service configuration.


Name: CorpR...

To configure the new web service, click *Options*.



Web Service options for CorpREST

HTTP Method: Post

HTTP Headers

URL: https://sles12.mymac.com/RestServer.php/FormData

Basic Auth Username: SSPR-REST-CLIENT

Basic Auth Password: ••••••••

Certificates: View Certificates

**Web Service Options:**

**HTTP Method** *Post*

*Currently, only Post is listed in the pull-down list.*

**HTTP Headers** *Optional*

In this field, HTTP headers may be specified as required by the external HTTP/REST service.

Example: `MyCorpSignature: nlsYWNsYXZl`

**URL** *Required.*

This field is the address and path to the *external HTTP/REST service*.

**Basic Auth Username** *Optional*

This field, along with the *Basic Auth Password* field are used to compile an *Authorization: Basic ...* HTTP header.

**Basic Auth Password** *Optional*

This field, along with the *Basic Auth Username* field are used to compile an *Authorization: Basic ...* HTTP header.

**Certificates** *Status - View Only*

Indicates if HTTPS:// certificates have been imported.

**None** Indicates that no certificates have been imported.

**View Certificates** Click to see imported certificates.

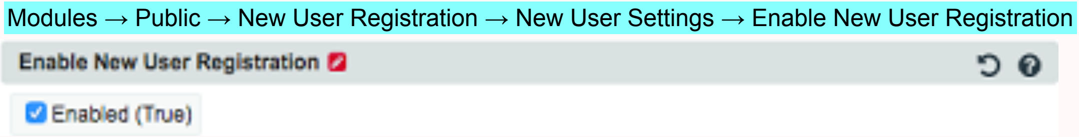
**Import Certificates** *Optional*

If the URL requires a secure (HTTPS://) connection, the required client certificate(s) can be imported from the *external REST server* by clicking *Import Certificates*.

When finished with the *external HTTP/REST server* options, click *OK*.

## Enable the New User Registration Module

The *Remote Form Data Service* feature is currently limited to the *New User Registration module*, allowing new users to register themselves, and create a user account. The *New User Registration module* can be enabled in SSPR's Configuration Editor:



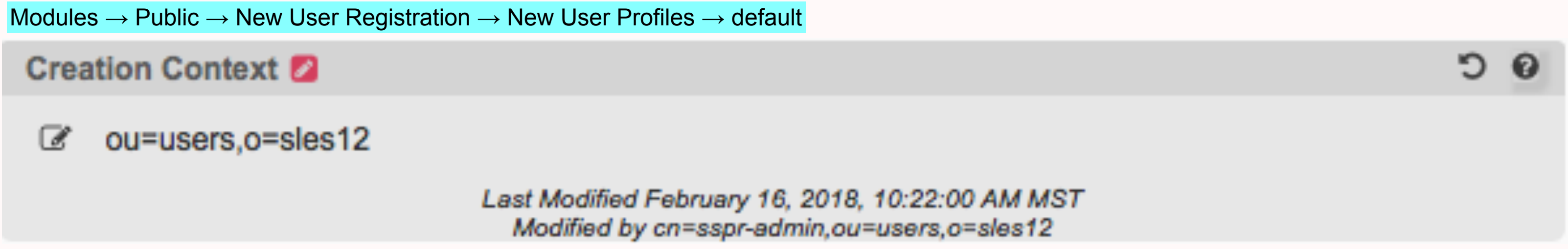
Modules → Public → New User Registration → New User Settings → Enable New User Registration

Enable New User Registration

☑ Enabled (True)

## Configure a New User Profile

There are additional configurations required for successful new user registrations in *New User Profiles*, where multiple new user profiles may be defined. A *New User Profile* defines (within an LDAP directory) where new user accounts will be created. For example, to view/edit the default new user profile in the SSPR configuration editor:



Modules → Public → New User Registration → New User Profiles → default

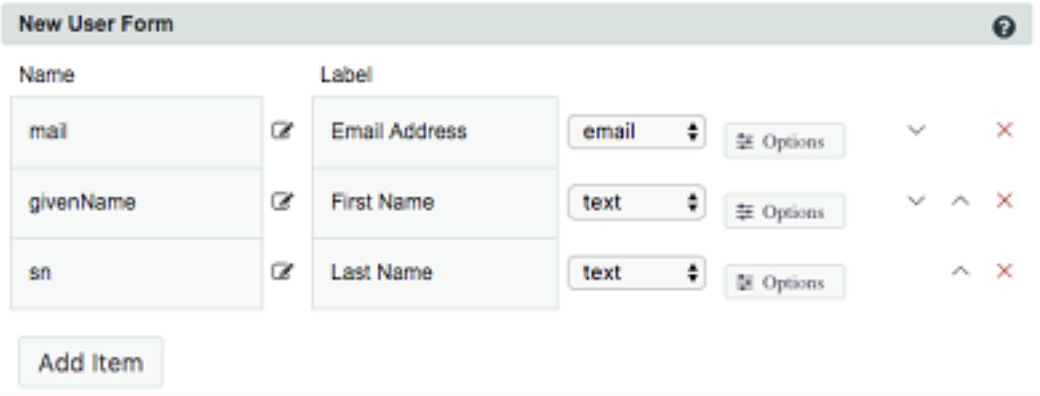
Creation Context

ou=users,o=sles12

*Last Modified February 16, 2018, 10:22:00 AM MST*
*Modified by cn=sspr-admin,ou=users,o=sles12*

## Configure Additional Form Fields

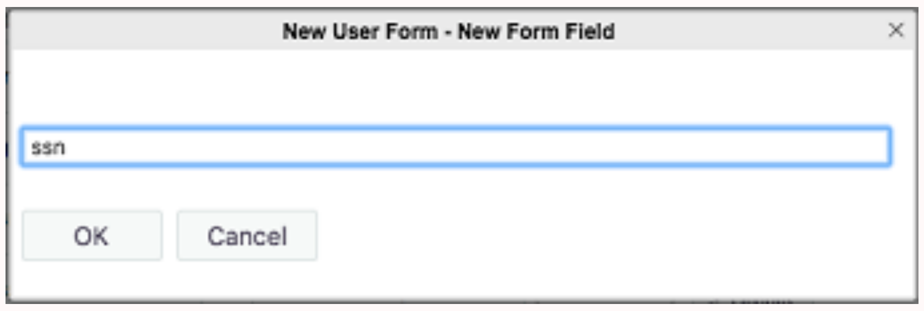
In the New User Profile's *New User Form* section, a list of form fields are listed. By default, the listed fields are used to create new LDAP accounts:



New User Form

| Name | Label | | |
|---|---|---|---|
| mail | Email Address | email | Options |
| givenName | First Name | text | Options |
| sn | Last Name | text | Options |

A new field can be added, who's value will be sent to an *external REST server application*. This is done by clicking the *Add Item* button, which will bring up a dialog box where the name of the new field is to be entered:

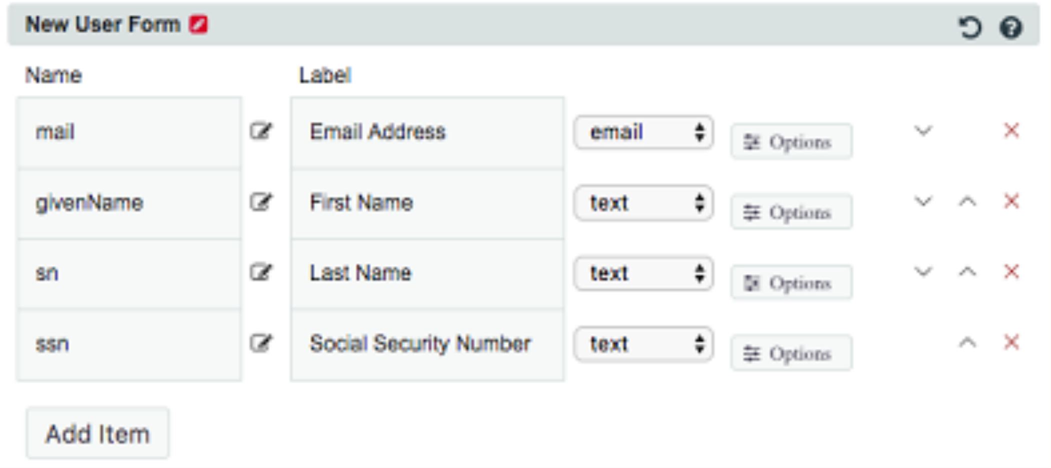


New User Form - New Form Field

ssn

In this example, a new form field named *ssn* is specified. Click *OK* to create the new field.

Now, the *New User Form* will display the new *ssn* field:

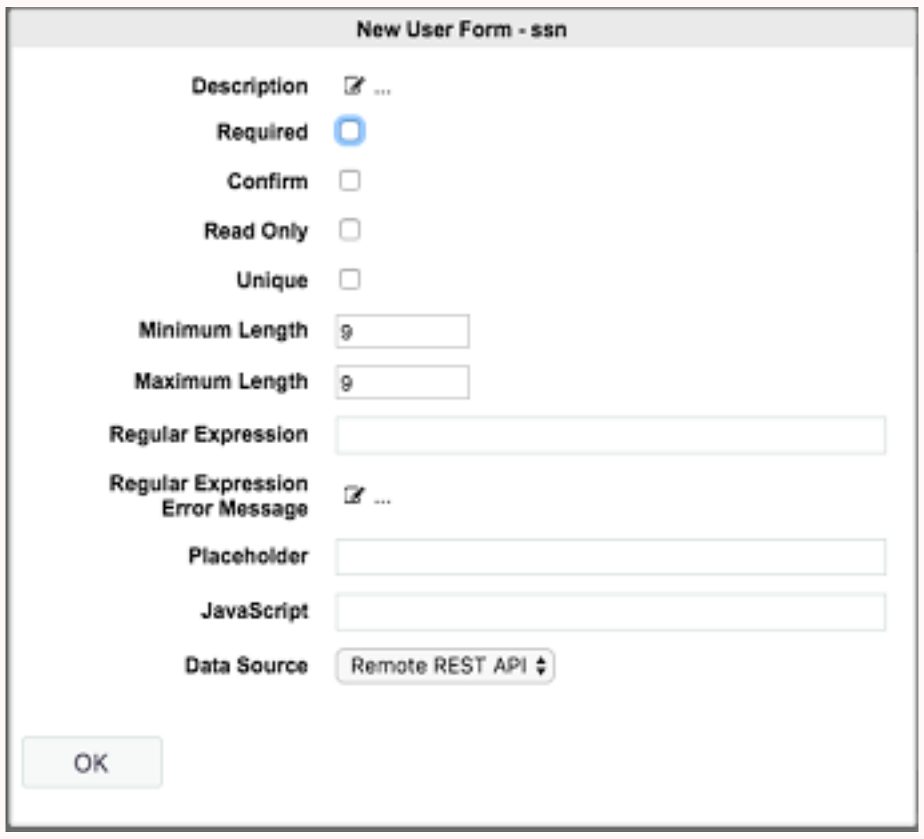


New User Form

| Name | Label | | |
|---|---|---|---|
| mail | Email Address | email | Options |
| givenName | First Name | text | Options |
| sn | Last Name | text | Options |
| ssn | Social Security Number | text | Options |

By default, SSPR will cause the new *ssn* form field to be sent to the LDAP directory server (along with the mail, givenName and sn fields).

In order to cause the field to be sent to the external REST server application, *(and not to the LDAP directory)*, click the *Options* link on the *ssn*.

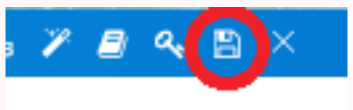


New User Form - ssn

| | |
|---|---|
| Description | ... |
| Required | □ |
| Confirm | □ |
| Read Only | □ |
| Unique | □ |
| Minimum Length | 9 |
| Maximum Length | 9 |
| Regular Expression | |
| Regular Expression Error Message | ... |
| Placeholder | |
| JavaScript | |
| Data Source | Remote REST API |

Change the *Data Source* from **LDAP** to **Remote REST API**, then click *OK*.

Before leaving SSPR's *Configuration Editor*, don't forget to save the changes.



# POST Method

## Request Headers

**Accept** *Content protocol type*

SSPR will only accept the content type *application/json* for the resulting SSPR REST response.

**Content-Type** *Request content protocol type.*

This SSPR REST request will only produce the content type:
*application/json; charset=UTF-8*

**Accept-Language:** *Language code*

The host (REST server) should processes the request in the context of the specified language code. For more detail on SSPR locale codes, see Appendix F: Locales (Languages) and Flags.

**Content-Length** *Request content length*

Length of request content in bytes.

**Host** *Target host (& port) of request*

Specifies the target Internet (virtual) host and port number of the resource being requested. This value is specified in SSPR's configuration:
Settings → Web Services → REST Clients → Remote Form Data Service

**Connection** *Specifies connection options.*

This SSPR request will indicate the *Keep-Alive* connection request.

**User-Agent** *Agent originating the request.*

Requests from SSPR will indicate the PWM snapshot version used in the SSPR build.

**Accept-Encoding** *Acceptable response content encoding options.*

This SSPR request will indicate that *gzip* or *deflate* encoding options are acceptable.

## Request Query String Parameters

None

## Request Content

The content is represented in the json format. The json root object contents include the following child objects:

**formInfo**

Describes REST call and form meta-data. This object contains the following child objects:

**module**

The SSPR module that made this REST request. For SSPR 4.2, the value of this field will be *NewUser*, indicating the *New User Registration Form*.

**moduleProfileID**

The moduleProfileID field equates to the entries under *New User Profiles* in SSPR's *Configuration Editor*.

**mode**

The value of this field indicates a requested action for the REST service handler. This object will contain one of the following values:

**verify**

Indicates a request from SSPR, to the REST server, to validate the form's fields.

**write**

Indicates a request from SSPR, to the REST server, to store the form's values.

**sessionID**

The session ID value is presented to aid the HTTP/REST server in caching records, etc. This value represents the SSPR session with the client through all *form validation* requests, as well as the eventual *write* request.

### formValues

This array object includes an array member object for each form field, both those flagged to be sent to the *Remote Form Data Service*, as well as those flagged to be sent to LDAP to create the LDAP user.

**Note:** The *password* field is not currently passed under *formValues*.

Each array member object contains a form field name and value pair.

### formConfigurations

This array object includes an array member object for each form field, both those flagged to be sent to the *Remote Form Data Service*, as well as those flagged to be sent to LDAP to create the LDAP user.

**Note:** The *password* field is not currently passed under *formValues*.

Each array member object contains child objects that describe the configuration parameters for a form field (as defined by SSPR). These child objects include:

**name**

The field's name, as it is recognized at it's LDAP, or remote, *source*. (See the *source* field below.)

**minimumLength**

The configured minimum length of the field's value.

**maximumLength**

The configured maximum length of the field's value.

**type**

Field types include:

- *text*
- *email*
- *number*
- *password*
- *random*
- *tel*
- *hidden*
- *url*
- *select*
- *checkbox*

**source**

This object is only present for *remote* data sources (as specified in the *SSPR Configuration Editor* in the applicable *New User Profile*, in the form field's *Options*, in the *Data Source* selection).

If this object is present, the object value is *remote*; which indicates that the target field value will not be stored by LDAP. If it is stored at all, it must be stored by the *Remote Form Data Service*.

**required**

Specifies if the form field must be given a value before submitting the form.

**true**

Indiates that the field cannot be left blank.

**false**

Indicates that the field may be left blank.

**confirmationRequired**

**true**

Indicates that the field is represented twice on the form, and that the user is required to enter the value twice. SSPR will require that the field values entered match. The duplicate *confirm* field is represented in the *formValues* with the same field name with the postfix ***_confirm*** along with the value of the duplicate field.

**false**

No confirmation of the field value is performed.

**readonly**

**Note:** In an SSPR New User Form context, this is only useful when there is signedForm data injected, as in the case of social auth. For example, the NetIQ NAM product will inject (pre-populate) initial values from the social auth provider by pre-signing it with the /signing/form endpoint, then start the newuser registration, by including those form values on the link, to SSPR. For some of those values, like the social-auth GUID, the field value should not be changed (only viewed).

**true**

The form field value cannot be modified, only viewed.

**false**

The form field value can be modified.

**unique**

Unique Attributes feature ensures that specified attributes always have unique values within a directory.

**true**

Indicates that the value should always be unique when stored in a database, etc. If the REST service handler finds that the proposed value is not unique, they reply should report an error.

For fields maintained in LDAP, the associated value will always have a unique value.

**false**

The field value's uniqueness is not relevant.

**multivalue**

In the context of the *New User* form, the value of this field will be *false* when sent by SSPR to the REST call handler.

**labels**

Lists the label, per locale, that is displayed on the form for this field.

The first entry represents the American English *(en)* locale, and has a blank *(ie: "")* locale value. *(For more detail on SSPR locale codes, see Appendix F: Locales (Languages) and Flags.)*

Example: `"","First Name"`

Subsequent field labels will indicate a local code.

Example: `"es":"Nombre de pila"`

**regexErrors**

Defines a search pattern, used by SSPR, where if the field value matches the pattern, the field value is unacceptable.

**description**

Long description of the field. This value is presented on the form near the field label.

The first entry represents the American English (en) locale, and has a blank (ie: "") locale value. *(For more detail on SSPR locale codes, see Appendix F: Locales (Languages) and Flags.)*

Example: `"","Please enter your first name"`

Subsequent field labels will indicate a local code.

Example: `"es":"Por favor, introduzca su nombre de pila"`

**regex**

Defines a search pattern, used by SSPR, where if the field value does not match the pattern, the field value is unacceptable.

**placeholder**

Specifies a hint that describes the expected value of a form field that is displayed in the field prior to the user entering a value.

**selectOptions**

In the context of the *New User* form, the value of this field will be empty or missing.

## Response Status Codes

The host that handles this request will return an HTTP status in the response. SSPR will expect one of the following status (code) values:

**200 OK** *Successful*

The call was successfully serviced by the REST service. Other status code values will cause SSPR to display, on the *New User* form, the error status:

*`A required service is unavailable. PLease try again later.`*

**500 Internal Server Error** *Failure*

The call could not be serviced by the REST service.

## Response Headers

**Date** *HTTP-date*

Optional. SSPR currently ignore's this header. The date and time that the response originated in *HTTP-date* format as defined by RFC 7231 Date/Time Formats.

**Server** *server*

Optional. SSPR currently ignore's this header. Information about the software used by the host that handled the request.

**Content-Length** *bytes*

Number of octets (8-bit bytes) found in the *response content*.

**Connection** *Connection options.*

This SSPR will respond with supportable connection options, as found in the REST request header (such as *Keep-Alive* etc.).

**Content-Type** *application/json*

Content protocol type (following the HTTP response headers). SSPR will currently only accept the content type *application/json* for this SSPR REST response.

## Response Content

The reply content is represented in the json format. The json root object contents may include the following child objects:

**error**

Required.

**true**

Indicates that the REST *validation* or *write* request failed. For a *validation* request, this indicates that there are problems with one or more of the field values.

**false**

Indicates that the REST *validation* or *write* request succeeded.

**errorCode**

Required when the *error* field value is *true*. The actual errorCode value is generated by the external REST service. This value is noted in SSPR.log, but otherwise ignored by SSPR.

**errorMessage**

Required when the *error* field value is *true*.

Example: `Accept-Locale: en`

For more information on *Accept-Locale* values, see Appendix F: Locales (Languages) and Flags

**successMessage**

Required when the *error* field value is *false*.

This value is localized to the REST request's header.

Example: `Accept-Locale: en`

For more information on *Accept-Locale* values, see Appendix F: Locales (Languages) and Flags

**errorDetail**

Required when the *error* field value is *true*.

The value of this field is displayed on SSPR's *New User* form as an error status.

This value is not localized, and should always be in English.

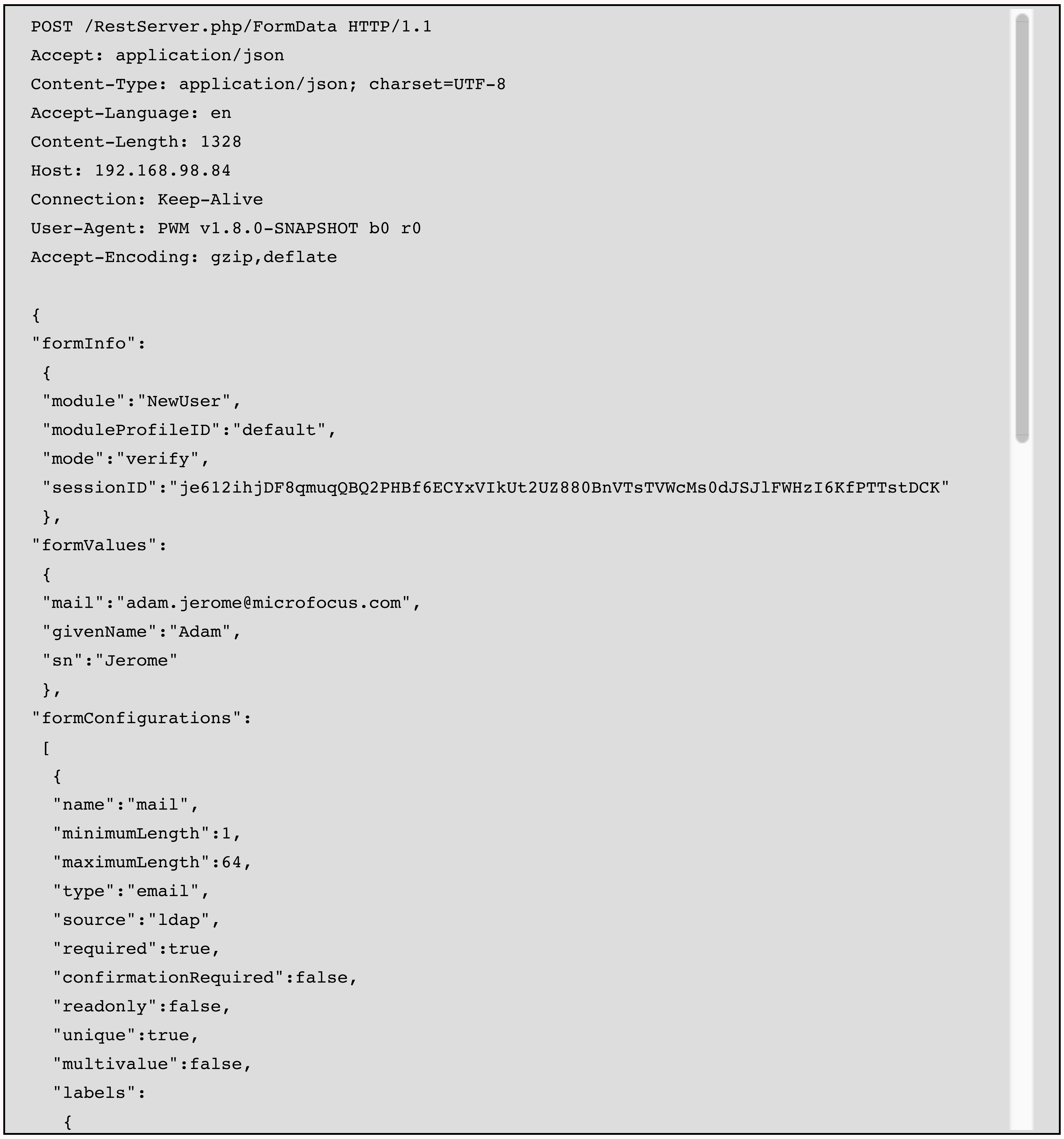
## Examples

### Example #1: SSPR Remote Form Data Service Validation Request

A form field validation request is an HTTP/REST request from SSPR to validate the content of user fields. Several of these requests may be initiated as a user is filling out the form.

The REST *form validation request* will include any optional HTTP headers that were specified by the *Web Service Options* for the *Remote Form Data Service*.

The content of the REST validation request is in JSON format (as indicated by the HTTP header: `Accept: application/json`).



```
POST /RestServer.php/FormData HTTP/1.1
Accept: application/json
Content-Type: application/json; charset=UTF-8
Accept-Language: en
Content-Length: 1328
Host: 192.168.98.84
Connection: Keep-Alive
User-Agent: PWM v1.8.0-SNAPSHOT b0 r0
Accept-Encoding: gzip,deflate

{
"formInfo":
 {
 "module":"NewUser",
 "moduleProfileID":"default",
 "mode":"verify",
 "sessionID":"je612ihjDF8qmuqQBQ2PHBf6ECYxVIkUt2UZ880BnVTsTVWcMs0dJSJlFWHzI6KfPTTstDCK"
 },
"formValues":
 {
 "mail":"adam.jerome@microfocus.com",
 "givenName":"Adam",
 "sn":"Jerome"
 },
"formConfigurations":
 [
  {
  "name":"mail",
  "minimumLength":1,
  "maximumLength":64,
  "type":"email",
  "source":"ldap",
  "required":true,
  "confirmationRequired":false,
  "readonly":false,
  "unique":true,
  "multivalue":false,
  "labels":
   {
```

Successful HTTP Response to SSPR example:

```
HTTP/1.1 200 SUCCESS
Date: Tue, 27 Feb 2018 19:12:13 GMT
Server: Apache
Content-Length: 51
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: application/json

{
"error":false,
"successMessage":
"Looks good."
}
```

Unsuccessful, due to an unsuitable field value, JSON body response to SSPR, example:

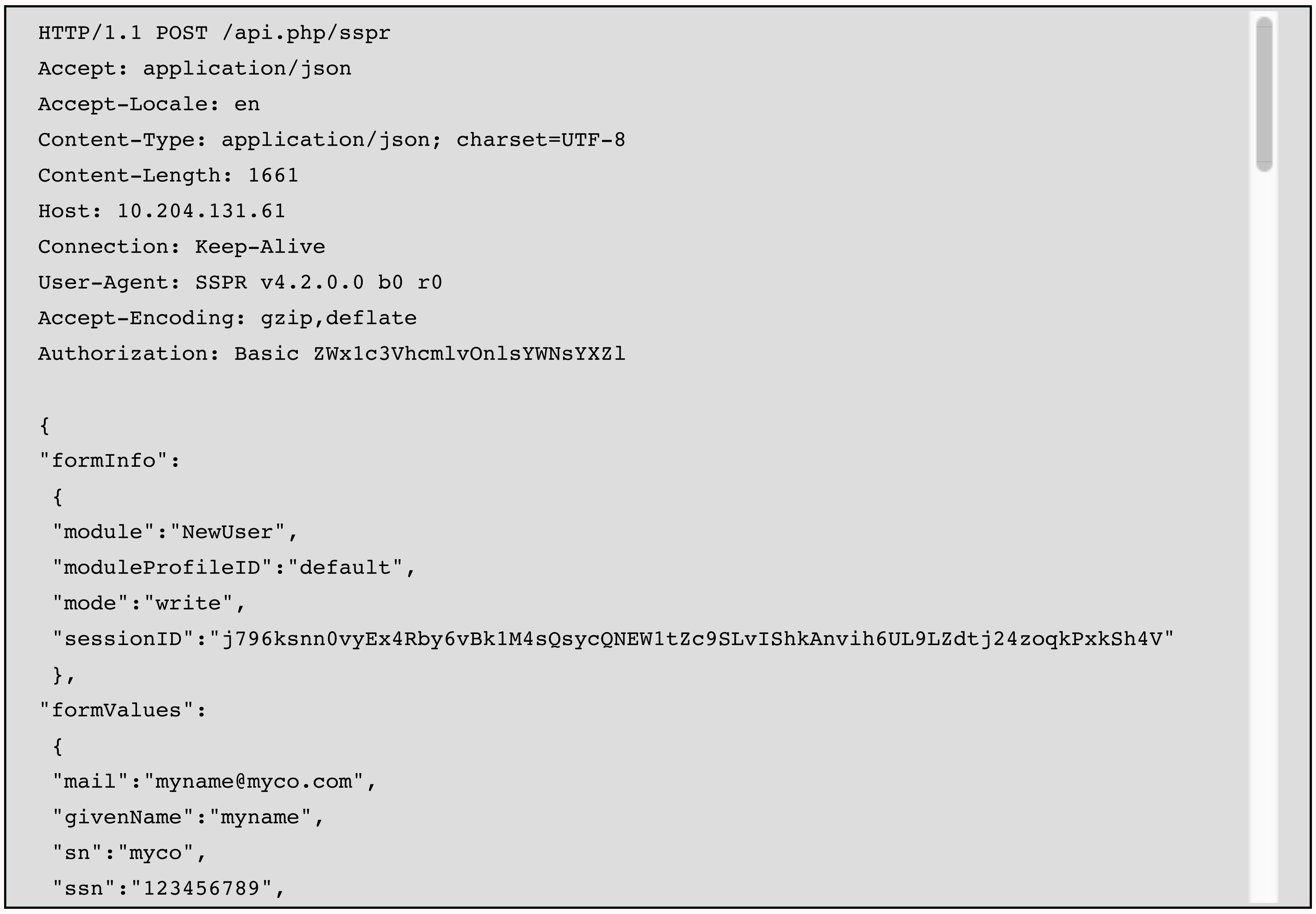
```
{
"error":true,
"errorCode":12345,
"errorMessage":"Invalid credit card number.",
"errorDetail":"Invalid credit card number."
}
```

### Example #2: SSPR Remote Form Data Service Write Request

A form field write request is an HTTP/REST request from SSPR to write, or store, the content of the form fields.

The REST *form write request* will include any optional HTTP headers that were specified by the *Web Service Options* for the *Remote Form Data Service*.

The content of the REST write request is in JSON format (as indicated by the HTTP header: `Accept: application/json`).



```
HTTP/1.1 POST /api.php/sspr
Accept: application/json
Accept-Locale: en
Content-Type: application/json; charset=UTF-8
Content-Length: 1661
Host: 10.204.131.61
Connection: Keep-Alive
User-Agent: SSPR v4.2.0.0 b0 r0
Accept-Encoding: gzip,deflate
Authorization: Basic ZWx1c3VhcmlvOnlsYWNsYXZl

{
"formInfo":
 {
 "module":"NewUser",
 "moduleProfileID":"default",
 "mode":"write",
 "sessionID":"j796ksnn0vyEx4Rby6vBk1M4sQsycQNEW1tZc9SLvIShkAnvih6UL9LZdtj24zoqkPxkSh4V"
 },
"formValues":
 {
 "mail":"myname@myco.com",
 "givenName":"myname",
 "sn":"myco",
 "ssn":"123456789",
```

# Appendix A: Errors

| Error # | Key | Resource Key |
|---|---|---|
| 4001 | PASSWORD_MISSING_CONFIRM | Password_MissingConfirm |
| Password meets requirements, please type confirmation password | | |
| 4002 | PASSWORD_MISSING | Password_Missing |
| Password missing | | |
| 4003 | PASSWORD_DOESNOTMATCH | Password_DoesNotMatch |
| Passwords do not match | | |
| 4004 | PASSWORD_PREVIOUSLYUSED | Password_PreviouslyUsed |
| New password has been used previously | | |
| 4005 | PASSWORD_BADOLDPASSWORD | Password_BadOldPassword |
| The old password is not correct | | |
| 4006 | PASSWORD_BADPASSWORD | Password_BadPassword |
| New password does not meet rule requirements | | |
| 4007 | PASSWORD_TOO_SHORT | Password_TooShort |
| New password is too short | | |
| 4008 | PASSWORD_TOO_LONG | Password_TooLong |
| New password is too long | | |
| 4009 | PASSWORD_NOT_ENOUGH_NUM | Password_NotEnoughNum |
| New password does not have enough numbers | | |
| 4010 | PASSWORD_NOT_ENOUGH_ALPHA | Password_NotEnoughAlpha |
| New password does not have enough letters | | |
| 4011 | PASSWORD_NOT_ENOUGH_SPECIAL | Password_NotEnoughSpecial |
| New password does not have enough symbol (non alpha-numeric) characters | | |
| 4012 | PASSWORD_NOT_ENOUGH_LOWER | Password_NotEnoughLower |
| New password does not have enough lower case letters | | |
| 4013 | PASSWORD_NOT_ENOUGH_UPPER | Password_NotEnoughUpper |
| New password does not have enough upper case letters | | |
| 4014 | PASSWORD_NOT_ENOUGH_UNIQUE | Password_NotEnoughUnique |
| New password does not have enough unique characters | | |
| 4015 | PASSWORD_TOO_MANY_REPEAT | Password_TooManyRepeat |
| New password has too many repeating characters | | |
| 4016 | PASSWORD_TOO_MANY_NUMERIC | Password_TooManyNumeric |
| New password has too many numbers | | |
| 4017 | PASSWORD_TOO_MANY_ALPHA | Password_TooManyAlpha |
| New password has too many letters | | |
| 4018 | PASSWORD_TOO_MANY_LOWER | Password_TooManyLower |
| New password has too many lower case letters | | |
| 4019 | PASSWORD_TOO_MANY_UPPER | Password_TooManyUpper |
| New password has too many upper case letters | | |
| 4020 | PASSWORD_FIRST_IS_NUMERIC | Password_FirstIsNumeric |
| The first character must not be numeric | | |
| 4021 | PASSWORD_LAST_IS_NUMERIC | Password_LastIsNumeric |
| The last character must not be numeric | | |
| 4022 | PASSWORD_FIRST_IS_SPECIAL | Password_FirstIsSpecial |
| The first character must not be a symbol (non alpha-numeric) character | | |
| 4023 | PASSWORD_LAST_IS_SPECIAL | Password_LastIsSpecial |
| The last character must not be a symbol (non alpha-numeric) character | | |

| Error # | Key | Resource Key |
|---|---|---|
| 4024 | PASSWORD_TOO_MANY_SPECIAL | Password_TooManyNonAlphaSpecial |
| | New password has too many symbol (non alpha-numeric) characters | |
| 4025 | PASSWORD_INVALID_CHAR | Password_InvalidChar |
| | New password has an invalid character | |
| 4026 | PASSWORD_REQUIREDMISSING | Password_RequiredMissing |
| | New password is missing a required character | |
| 4027 | PASSWORD_INWORDLIST | Password_InWordlist |
| | New password is too common | |
| 4028 | PASSWORD_SAMEASOLD | Password_SameAsOld |
| | New password is the same as the current password | |
| 4029 | PASSWORD_SAMEASATTR | Password_SameAsAttr |
| | New password is too obvious | |
| 4030 | PASSWORD_MEETS_RULES | Password_MeetsRules |
| | New password accepted, please click change password | |
| 4031 | PASSWORD_TOO_MANY_OLD_CHARS | Password_TooManyOldChars |
| | New password contains too many characters from your old password | |
| 4032 | PASSWORD_HISTORY_FULL | Password_HistoryFull |
| | New password history is full | |
| 4033 | PASSWORD_TOO_SOON | Password_TooSoon |
| | Not enough time has passed since last password change | |
| 4034 | PASSWORD_USING_DISALLOWED | Password_UsingDisallowedValue |
| | New password is using a value that is not allowed | |
| 4035 | PASSWORD_TOO_WEAK | Password_TooWeak |
| | Password is too weak. Try adding more numbers, symbols or mixed case letters. | |
| 4036 | PASSWORD_TOO_MANY_NONALPHA | Password_TooManyNonAlpha |
| | New password has too many non-letter characters | |
| 4037 | PASSWORD_NOT_ENOUGH_NONALPHA | Password_NotEnoughNonAlpha |
| | New password does not have enough non-letter characters | |
| 4038 | PASSWORD_UNKNOWN_VALIDATION | Password_UnknownValidation |
| | New password does not meet requirements. Please try using a different password. | |
| 4039 | PASSWORD_NEW_PASSWORD_REQUIRED | Password_NewPasswordRequired |
| | A new password is required before you may continue. | |
| 4040 | PASSWORD_EXPIRED | Password_Expired |
| | The password is expired. | |
| 4041 | PASSWORD_CUSTOM_ERROR | Password_CustomError |
| | New password does not meet rule requirements | |
| 4042 | PASSWORD_NOT_ENOUGH_GROUPS | Password_NotEnoughGroups |
| | New password does not contain enough different types of characters | |
| 4043 | PASSWORD_TOO_MANY_CONSECUTIVE | Password_TooManyConsecutive |
| | New password has too many consecutive characters (such as 123456... or abcdef...) | |
| 5001 | ERROR_WRONGPASSWORD | Error_WrongPassword |
| | The user name or password is not valid. Please try again. | |
| 5002 | ERROR_INCORRECT_RESPONSE | Error_WrongResponse |
| | One or more responses are not correct. Please try again. | |
| 5003 | ERROR_USERAUTHENTICATED | Error_UserAuthenticated |
| | You are already authenticated. | |
| 5004 | ERROR_AUTHENTICATION_REQUIRED | Error_AuthenticationRequired |
| | Authentication required. | |
| 5006 | ERROR_RESPONSES_NORESPONSES | Error_Response_NoResponse |
| | The user name is not valid or is not eligible to use this feature | |

| Error # | Key | Resource Key |
|---|---|---|
| 5007 | ERROR_RESPONSE_WORDLIST | Error_Response_Wordlist |
| | The response for question "%1%" is too commonly used | |
| 5008 | ERROR_RESPONSE_TOO_SHORT | Error_Response_TooShort |
| | The response for question "%1%" is too short | |
| 5009 | ERROR_RESPONSE_TOO_LONG | Error_Response_TooLong |
| | The response for question "%1%" is too long | |
| 5010 | ERROR_RESPONSE_DUPLICATE | Error_Response_Duplicate |
| | The response for question "%1%" can not be the same as another response | |
| 5011 | ERROR_CHALLENGE_DUPLICATE | Error_Challenge_Duplicate |
| | Each question must be unique. | |
| 5012 | ERROR_MISSING_CHALLENGE_TEXT | Error_Missing_Challenge_Text |
| | Missing text for a user supplied question | |
| 5013 | ERROR_MISSING_PARAMETER | Error_MissingParameter |
| | A required parameter is missing. | |
| 5015 | ERROR_UNKNOWN | Error_Unknown |
| | An error has occurred. If this error occurs repeatedly please contact your help desk. | |
| 5016 | ERROR_CANT_MATCH_USER | Error_CantMatchUser |
| | Unable to find user name. Please try again. | |
| 5017 | ERROR_DIRECTORY_UNAVAILABLE | Error_DirectoryUnavailable |
| | Directory unavailable. If this error occurs repeatedly please contact your help desk. | |
| 5018 | ERROR_ACTIVATION_VALIDATIONFAIL | Error_ActivationValidationFailed |
| | One or more values are not correct. | |
| 5019 | ERROR_SERVICE_NOT_AVAILABLE | Error_ServiceNotAvailable |
| | Service is not enabled. | |
| 5020 | ERROR_USER_MISMATCH | Error_UserMisMatch |
| | Authentication error, please close your browser. | |
| 5021 | ERROR_ACTIVATE_NO_PERMISSION | Error_ActivateUserNoQueryMatch |
| | Your user account is not eligible for activation. | |
| 5022 | ERROR_NO_CHALLENGES | Error_NoChallenges |
| | No challenges have been configured. | |
| 5023 | ERROR_INTRUDER_USER | Error_UserIntruder |
| | Maximum login attempts for your userID have been exceeded. Try again later. | |
| 5024 | ERROR_INTRUDER_ADDRESS | Error_AddressIntruder |
| | Maximum login attempts have been exceeded. Try again later. | |
| 5025 | ERROR_INTRUDER_SESSION | Error_SessionIntruder |
| | Maximum login attempts for this session have been exceeded. Try again later. | |
| 5026 | ERROR_BAD_SESSION_PASSWORD | Error_BadSessionPassword |
| | Unable to establish session password. | |
| 5027 | ERROR_UNAUTHORIZED | Error_Unauthorized |
| | You do not have permission to perform the requested action. | |
| 5028 | ERROR_BAD_SESSION | Error_BadSession |
| | Unable to establish a session with your browser. Please close your browser and try again. | |
| 5029 | ERROR_MISSING_REQUIRED_RESPONSE | Error_MissingRequiredResponse |
| | Please type all of the required responses. | |
| 5030 | ERROR_MISSING_RANDOM_RESPONSE | Error_MissingRandomResponse |
| | Please add an additional random response. | |
| 5031 | ERROR_BAD_CAPTCHA_RESPONSE | Error_BadCaptchaResponse |
| | Incorrect verification code, please try again. | |

| Error # | Key | Resource Key |
|---|---|---|
| 5032 | ERROR_CAPTCHA_API_ERROR | Error_CaptchaAPIError |
| | An error occurred while validating CAPTCHA response. Please close your browser and try again. If this error occurs repeatedly contact your help desk. | |
| 5033 | ERROR_INVALID_CONFIG | Error_InvalidConfig |
| | The configuration is invalid or corrupt. Please correct the error, or remove the configuration file. | |
| 5034 | ERROR_INVALID_FORMID | Error_InvalidFormID |
| | The browser session is invalid or has expired. Please try again. | |
| 5035 | ERROR_INCORRECT_REQ_SEQUENCE | Error_IncorrectRequestSequence |
| | An out of order page request has been received. Please do not use the browser back button. Please try again. | |
| 5036 | ERROR_TOKEN_MISSING_CONTACT | Error_TokenMissingContact |
| | There is no contact information available for your account. Please contact your administrator. | |
| 5037 | ERROR_TOKEN_INCORRECT | Error_TokenIncorrect |
| | Incorrect code, please try again. | |
| 5038 | ERROR_BAD_CURRENT_PASSWORD | Error_BadCurrentPassword |
| | Current password is incorrect, please try again. | |
| 5039 | ERROR_CLOSING | Error_Closing |
| | The operation can not complete because the application is shutting down. | |
| 5040 | ERROR_MISSING_GUID | Error_Missing_GUID |
| | Unable to locate a GUID for user. Please contact your administrator. | |
| 5041 | ERROR_TOKEN_EXPIRED | Error_TokenExpired |
| | The token you have entered is expired and is no longer valid. Please try again. | |
| 5042 | ERROR_MULTI_USERNAME | Error_Multi_Username |
| | Multiple users match the given user name "%1%". Please refine your search. | |
| 5043 | ERROR_ORIG_ADMIN_ONLY | Error_Orig_Admin_Only |
| | Only the original administrator can perform this property | |
| 5044 | ERROR_SECURE_REQUEST_REQUIRED | Error_SecureRequestRequired |
| | Non-secure (HTTP) connections are not permitted to this system. Please try again using a secure (HTTPS) connection. | |
| 5045 | ERROR_WRITING_RESPONSES | Error_Writing_Responses |
| | An error occurred during the save of your response questions. Please contact your administrator. | |
| 5046 | ERROR_UNLOCK_FAILURE | Error_Unlock_Failure |
| | An error occurred while unlocking your account. Please contact your administrator. | |
| 5047 | ERROR_UPDATE_ATTRS_FAILURE | Error_Update_Attrs_Failure |
| | An error occurred while saving your profile information. Please contact your administrator. | |
| 5048 | ERROR_ACTIVATION_FAILURE | Error_Activation_Failure |
| | An error occurred while activating your account. Please contact your administrator. | |
| 5049 | ERROR_NEW_USER_FAILURE | Error_NewUser_Failure |
| | An error occurred while creating your new user account. Please contact your administrator. | |
| 5050 | ERROR_ACTIVATION | Error_Activation |
| | Unable to activate your account using the information you have provided. Please try again. | |
| 5051 | ERROR_DB_UNAVAILABLE | Error_DB_Unavailable |
| | Database Unavailable. If this error occurs repeatedly please contact your help desk. | |
| 5052 | ERROR_LOCALDB_UNAVAILABLE | Error_LocalDB_Unavailable |
| | LocalDB Unavailable. Please contact your administrator. | |

| Error # | Key | Resource Key |
|---|---|---|
| 5053 | ERROR_APP_UNAVAILABLE | Error_App_Unavailable |
| | The application is unavailable or is restarting. If this error occurs repeatedly please contact your help desk. | |
| 5054 | ERROR_UNREACHABLE_CLOUD_SERVICE | Error_UnreachableCloudService |
| | A remote service was unreachable. | |
| 5055 | ERROR_INVALID_SECURITY_KEY | Error_InvalidSecurityKey |
| | Security Key is missing or invalid. | |
| 5056 | ERROR_CLEARING_RESPONSES | Error_Clearing_Responses |
| | An error occurred during the clearing of the response questions. Please contact your administrator. | |
| 5057 | ERROR_SERVICE_UNREACHABLE | Error_ServiceUnreachable |
| | A required service is unavailable. Please try again later. | |
| 5058 | ERROR_CHALLENGE_IN_RESPONSE | Error_ChallengeInResponse |
| | The response for question "%1%" cannot contain part of the question text. | |
| 5059 | ERROR_CERTIFICATE_ERROR | Error_CertificateError |
| | A certificate error has been encountered: %1%. | |
| 5060 | ERROR_SYSLOG_WRITE_ERROR | Error_SyslogWriteError |
| | A problem writing to the syslog server has been encountered, error: %1% | |
| 5061 | ERROR_TOO_MANY_THREADS | Error_TooManyThreads |
| | Maximum thread count limit exceeded, please try again later | |
| 5062 | ERROR_PASSWORD_REQUIRED | Error_PasswordRequired |
| | A password is required to perform this operation | |
| 5063 | ERROR_SECURITY_VIOLATION | Error_SecurityViolation |
| | A security violation has occurred. Please try again later. | |
| 5064 | ERROR_TRIAL_VIOLATION | Error_TrialViolation |
| | Trial limits have been exceeded. | |
| 5065 | ERROR_ACCOUNT_DISABLED | Error_AccountDisabled |
| | Account is disabled. | |
| 5066 | ERROR_ACCOUNT_EXPIRED | Error_AccountExpired |
| | Account is expired. | |
| 5087 | ERROR_NO_OTP_CONFIGURATION | Error_NoOtpConfiguration |
| | No one-time password has been configured. | |
| 5088 | ERROR_INCORRECT_OTP_TOKEN | Error_WrongOtpToken |
| | Incorrect one-time password. | |
| 5086 | ERROR_WRITING_OTP_SECRET | Error_Writing_Otp_Secret |
| | An error occurred during the save of your OTP secret. Please contact your administrator. | |
| 5067 | ERROR_INTRUDER_ATTR_SEARCH | Error_AttrIntruder |
| | Maximum search attempts have been exceeded. Try again later. | |
| 5068 | ERROR_AUDIT_WRITE | Error_AuditWrite |
| | Unable to write audit record. | |
| 5069 | ERROR_INTRUDER_LDAP | Error_LdapIntruder |
| | Maximum login attempts for your userID have been exceeded. Try again later. | |
| 5070 | ERROR_NO_LDAP_CONNECTION | Error_NoLdapConnection |
| | A connection to the required directory is not available. | |
| 5071 | ERROR_OAUTH_ERROR | Error_OAuthError |
| | An error using the OAuth authentication protocol has occurred. Please try again later. | |
| 5072 | ERROR_REPORTING_ERROR | Error_ReportingError |
| | An error during report generation occurred | |
| 5073 | ERROR_INTRUDER_TOKEN_DEST | Error_TokenDestIntruder |
| | Maximum attempts have been exceeded. Try again later. | |

| Error # | Key | Resource Key |
|---|---|---|
| 5074 | ERROR_OTP_RECOVERY_USED | Error_OtpRecoveryUsed |
| | The recovery could has been previously used and cannot be used again. | |
| 5075 | ERROR_REDIRECT_ILLEGAL | Error_RedirectIllegal |
| | The requested redirect url is not permitted. | |
| 5076 | ERROR_CRYPT_ERROR | Error_CryptError |
| | An unexpected cryptography error has occurred. | |
| 5078 | ERROR_SMS_SEND_ERROR | Error_SmsSendError |
| | Unable to send sms message: %1% | |
| 5079 | ERROR_LDAP_DATA_ERROR | Error_LdapDataError |
| | An LDAP data error has occurred. | |
| 5080 | ERROR_MACRO_PARSE_ERROR | Error_MacroParseError |
| | Macro parse error: %1% | |
| 5081 | ERROR_NO_PROFILE_ASSIGNED | Error_NoProfileAssigned |
| | No profile is assigned for this operation. | |
| 5082 | ERROR_STARTUP_ERROR | Error_StartupError |
| | An error occurred while starting the application. Check the log files for information. | |
| 5083 | ERROR_ENVIRONMENT_ERROR | Error_EnvironmentError |
| | An error with the application environment has prevented the application from starting. | |
| 5084 | ERROR_APPLICATION_NOT_RUNNING | Error_ApplicationNotRunning |
| | This functionality is not available until the application configuration is restricted. | |
| 5085 | ERROR_EMAIL_SEND_FAILURE | Error_EmailSendFailure |
| | Error sending email item %1%, error: %2% | |
| 5089 | ERROR_PASSWORD_ONLY_BAD | Error_PasswordOnlyBad |
| | Password incorrect. Please try again. | |
| 5100 | ERROR_FIELD_REQUIRED | Error_FieldRequired |
| | %1% is required | |
| 5101 | ERROR_FIELD_NOT_A_NUMBER | Error_FieldNotANumber |
| | %1% must be a number | |
| 5102 | ERROR_FIELD_INVALID_EMAIL | Error_FieldInvalidEmail |
| | %1% is not a valid email address | |
| 5103 | ERROR_FIELD_TOO_SHORT | Error_FieldTooShort |
| | %1% is too short | |
| 5104 | ERROR_FIELD_TOO_LONG | Error_FieldTooLong |
| | %1% is too long | |
| 5105 | ERROR_FIELD_DUPLICATE | Error_FieldDuplicate |
| | %1% is already used, please use a different value | |
| 5106 | ERROR_FIELD_BAD_CONFIRM | Error_FieldBadConfirm |
| | %1% fields do not match | |
| 5107 | ERROR_FIELD_REGEX_NOMATCH | Error_FieldRegexNoMatch |
| | %1% is not the correct format | |
| 5200 | CONFIG_UPLOAD_SUCCESS | Error_ConfigUploadSuccess |
| | File uploaded successfully | |
| 5201 | CONFIG_UPLOAD_FAILURE | Error_ConfigUploadFailure |
| | File failed to upload. | |
| 5202 | CONFIG_SAVE_SUCCESS | Error_ConfigSaveSuccess |
| | Configuration saved successfully. Application restart has been requested. The application may be unavailable while restarting. If the restart request fails you may need to restart the application server manually. | |
| 5203 | CONFIG_FORMAT_ERROR | Error_ConfigFormatError |
| | Configuration format error: %1% | |

| | | | |
|---|---|---|---|
| 5204 | | CONFIG_LDAP_FAILURE | Error_ConfigLdapFailure |
| | Unable to connect to LDAP directory server. | | |
| 5205 | | CONFIG_LDAP_SUCCESS | Error_ConfigLdapSuccess |
| | Successfully connected to LDAP directory server | | |
| 5300 | | ERROR_HTTP_404 | Error_HTTP_404 |
| | The page you requested could not be found. | | |
| 6000 | | ERROR_REMOTE_ERROR_VALUE | Error_RemoteErrorValue |
| | Remote Error: %1% | | |
| 6001 | | ERROR_TELEMETRY_SEND_ERROR | Error_TelemetrySendError |
| | Error_TelemetrySendError | | |

# Appendix B: Event Statistics

| Key | Label | Description |
|---|---|---|
| **ACTIVATED_USERS** | Activated Users | Number of users that have successfully completed the user activation process. |
| **PWM_STARTUPS** | Application Startups | Number of times the application has started, including restarts due to configuration changes. |
| **AUDIT_EVENTS** | Audit Events | Number of audit events generated and stored locally. |
| **AUTHENTICATION_FAILURES** | Authentication Failures | Number of failed user authentications that have occurred. |
| **AUTHENTICATIONS** | Authentications | Number of successful user authentications that have occurred. |
| **AUTHENTICATION_EXPIRED** | Authentications with Expired Password | Number of authentications that were successful but the user's password was expired. |
| **AUTHENTICATION_EXPIRED_WARNING** | Authentications with Expired Warning | Number of authentications that were successful and the user's password was not yet expired, however the expiration time was within the expiration warning time window. |
| **AUTHENTICATION_PRE_EXPIRED** | Authentications with Pre-Expired Password | Number of authentications that were successful and the user's password was not yet expired, however the expiration time was within the pre-expire time window. |
| **AVG_AUTHENTICATION_TIME** | Average Authentication Time | Average time (in milliseconds) for authentications of all types to complete. |
| **AVG_LDAP_SEARCH_TIME** | Average LDAP Search Time | Average duration (in milliseconds) of LDAP searches. |
| **AVG_PASSWORD_STRENGTH** | Average Password Strength | Average password strength rating (0-100) of passwords set or changed in the application. |
| **AVG_PASSWORD_SYNC_TIME** | Average Password Sync Time | Average time (in milliseconds) users spend waiting for the password sync progress to complete. |
| **CAPTCHA_FAILURES** | CAPTCHA Failures | Number of incorrect user CAPTCHA attempts. |
| **CAPTCHA_PRESENTATIONS** | CAPTCHA Presentations | Number of CAPTCHA challenges presented to a user. |
| **CAPTCHA_SUCCESSES** | CAPTCHA Successes | Number of times user have successfully passed CAPTCHA verification. |
| **DB_UNAVAILABLE_COUNT** | Database Unavailable Count | Number of database unreachable errors encountered by the application. |

| Key | Label | Description |
|---|---|---|
| **EMAIL_SEND_DISCARDS** | Email Send Discards | Number of times an email item has been discarded due to an error. |
| **EMAIL_SEND_FAILURES** | Email Send Failures | Number of times a temporary error has occurred while sending an email. |
| **EMAIL_SEND_SUCCESSES** | Email Send Successes | Number of successfully delivered email items. |
| **FOREIGN_SESSIONS_ACCEPTED** | Foreign Sessions Accepted | Number of sessions generated on a foreign or server or previous instance of this server and accepted as valid authentication. |
| **RECOVERY_FAILURES** | Forgotten Password Failures | Number of user failures during forgotten password identification and verification process. |
| **RECOVERY_OTP_FAILED** | Forgotten Password OTP Secrets Failed | Number of invalid attempts to verify OTP secrets used by the forgotten password process. |
| **RECOVERY_OTP_PASSED** | Forgotten Password OTP Secrets Verified | Number of OTP secrets used by the forgotten password process successfully and correctly verified. |
| **RECOVERY_SUCCESSES** | Forgotten Password Successes | Number of times users have successfully validated their identity using the forgotten password module. |
| **RECOVERY_TOKENS_FAILED** | Forgotten Password Tokens Failed | Number of incorrect token attempts during forgotten password process. |
| **RECOVERY_TOKENS_PASSED** | Forgotten Password Tokens Passed | Number of tokens used for forgotten password process verified and claimed. |
| **RECOVERY_TOKENS_SENT** | Forgotten Password Tokens Sent | Number of tokens used for forgotten password process issued and sent via email or SMS. |
| **FORGOTTEN_USERNAME_FAILURES** | Forgotten User Name Failures | Number of user failures during forgotten user name identification and verification process. |
| **FORGOTTEN_USERNAME_SUCCESSES** | Forgotten User Name Successes | Number of successes using the forgotten user name process. |
| **GENERATED_PASSWORDS** | Generated Random Passwords | Number of system generated password values. |
| **GUESTS** | Guest Users Created | Number of user accounts that have been created using the guest management module. |
| **UPDATED_GUESTS** | Guest Users Updated | Number of user accounts have been updated/modified using the guest management module. |

| Key | Label | Description |
|---|---|---|
| **HTTP_REQUESTS** | HTTP Requests | Number of dynamic HTTP requests processed by the application. |
| **HTTP_RESOURCE_REQUESTS** | HTTP Resource Requests | Number of resource (static) HTTP requests processed by the application. |
| **HTTP_SESSIONS** | HTTP Sessions | Number of unique HTTP sessions processed by the application. |
| **HELPDESK_VERIFY_OTP** | Help Desk OTP Verifications | Number of successful Help Desk OTP verification user events. |
| **HELPDESK_PASSWORD_SET** | Help Desk Password Resets | Number of password modifications initiated using the Help Desk module. |
| **HELPDESK_TOKENS_SENT** | Help Desk Tokens Sent | Number of Help Desk verification tokens sent. |
| **HELPDESK_UNLOCK** | Help Desk Unlocks | Number of Help Desk unlock user events. |
| **HELPDESK_USER_LOOKUP** | Help Desk User Lookups | Number of Help Desk user detail views requested by Help Desk operators. |
| **INTRUDER_ATTEMPTS** | Intruder Attempts | Number of intruder attempts of any type. |
| **LOCKED_ADDRESSES** | Intruder Locked Addresses | Number of local intruder lock events that have occurred due to multiple attempts associated with the same network address. |
| **LOCKED_ATTRIBUTES** | Intruder Locked Attributes | Number of local intruder lock events that have occurred due to multiple attempts associated with the same form attributes. |
| **LOCKED_TOKENDESTS** | Intruder Locked Token Destinations | Number of local intruder lock events that have occurred due to multiple attempts associated with the same token destination address. |
| **LOCKED_USERIDS** | Intruder Locked User IDs | Number of local intruder lock events that have occurred due to multiple attempts associated with the same LDAP entry. |
| **LOCKED_USERS** | Intruder Locked User Names | Number of local intruder lock events that have occurred due to multiple attempts associated with the same user name. |
| **LDAP_UNAVAILABLE_COUNT** | LDAP Unavailable Count | Number of LDAP unreachable errors encountered by the application. |
| **NEW_USERS** | New User Self Registrations | Number of users that have successfully completed the new user self-registration process. |
| **OBSOLETE_URL_REQUESTS** | Obsolete URL Requests | Number of web requests to obsolete URLs. |
| **PASSWORD_CHANGES** | Password Changes | Number of times users have changed a password using the application. |
| **PWNOTIFY_EMAILS_SENT** | Password Notification Job Emails Sent | Number of emails that have been sent by password expiration notification jobs. |

| Key | Label | Description |
|---|---|---|
| **PWNOTIFY_JOB_ERRORS** | Password Notification Job Errors | Number of password expiration notification jobs that have ended with an error. |
| **PWNOTIFY_JOBS** | Password Notification Jobs Started | Number of password expiration notification jobs that have been started. |
| **PASSWORD_RULE_CHECKS** | Password Rule Checks | Number of password rule validation requests processed while users are typing a new password. |
| **PEOPLESEARCH_CACHE_HITS** | PeopleSearch Cache Hits | Number of cache hits when reading people search data. |
| **PEOPLESEARCH_CACHE_MISSES** | PeopleSearch Cache Misses | Number of cache misses when reading people search data. |
| **PEOPLESEARCH_DETAILS** | PeopleSearch Detail Views | Number of detailed user views executed using the people search module. |
| **PEOPLESEARCH_ORGCHART** | PeopleSearch Org Chart Views | Number of organizational chart views executed using the people search module. |
| **PEOPLESEARCH_SEARCHES** | PeopleSearch Searches | Number of directory searches executed using the people search module. |
| **UPDATE_ATTRIBUTES** | Profile Updates | Number of times users have completed the update profile process. |
| **SMS_SEND_DISCARDS** | SMS Send Discards | Number of times an SMS item has been discarded due to an error. |
| **SMS_SEND_FAILURES** | SMS Send Failures | Number of times a temporary error has occurred while sending an SMS. |
| **SMS_SEND_SUCCESSES** | SMS Send Successes | Number of successfully delivered SMS items. |
| **SETUP_RESPONSES** | Saved Secret Answers | Number of times users have saved challenge/responses answers. |
| **SETUP_OTP_SECRET** | Setup OTP Secret | Number of times users have saved an OTP secret. |
| **SHORTCUTS_SELECTED** | Shortcuts Selected | Number of shortcut items clicked on by users. |
| **SYSLOG_MESSAGES_SENT** | Syslog Messages Sent | Number of successfully sent syslog messages. |
| **TOKENS_PASSSED** | Tokens Claimed | Number of tokens used for any purpose verified and claimed. |
| **TOKENS_SENT** | Tokens Issued | Number of tokens used for any purpose issued and sent via email or SMS. |
| **PWM_UNKNOWN_ERRORS** | unhandled Errors | Number of times an unhandled error has occurred. |
| **REST_CHALLENGES** | WebService Challenge Calls | Number of external web service calls to the /challenges REST interface. |
| **REST_CHECKPASSWORD** | WebService Check Password Calls | Number of external web service calls to the /checkpassword REST interface. |

| Key | Label | Description |
|---|---|---|
| **REST_HEALTH** | WebService Health Calls | Number of external web service calls to the /health REST interface. |
| **REST_PROFILE** | WebService Profile Calls | Number of external web service calls to the /profile REST interface. |
| **REST_RANDOMPASSWORD** | WebService RandomPassword Calls | Number of external web service calls to the /randompassword REST interface. |
| **REST_SETPASSWORD** | WebService SetPassword Calls | Number of external web service calls to the /setpassword REST interface. |
| **REST_SIGNING_FORM** | WebService Signing Form Calls | Number of external web service calls to the /signing/form REST interface. |
| **REST_STATISTICS** | WebService Statistic Calls | Number of external web service calls to the /statistics REST interface. |
| **REST_STATUS** | WebService Status Calls | Number of external web service calls to the /status REST interface. |
| **REST_VERIFYCHALLENGES** | WebService Verify Challenge Calls | Number of external web service calls to the /verifychallenges REST interface. |
| **REST_VERIFYOTP** | WebService Verify OTP Calls | Number of external web service calls to the /verifyotp REST interface. |

# Appendix C: Events Per Second Statistics

| Key | Description |
|---|---|
| **AUTHENTICATION_DAY** | Average user authentication events per second for the current day. |
| **AUTHENTICATION_HOUR** | Average user authentication events per second for the current hour. |
| **AUTHENTICATION_MINUTE** | Average user authentication events per second for the current minute. |
| **DB_READS_DAY** | Average database read events per second for the current day. |
| **DB_READS_HOUR** | Average database read events per second for the current hour. |
| **DB_READS_MINUTE** | Average database read events per second for the current minute. |
| **DB_WRITES_DAY** | Average database write events per second for the current day. |
| **DB_WRITES_HOUR** | Average database write events per second for the current hour. |
| **DB_WRITES_MINUTE** | Average database write events per second for the current minute. |
| **INTRUDER_ATTEMPTS_DAY** | Average intruder attempt events per second for the current day. |
| **INTRUDER_ATTEMPTS_HOUR** | Average intruder attempt events per second for the current hour. |
| **INTRUDER_ATTEMPTS_MINUTE** | Average intruder attempt events per second for the current minute. |
| **PASSWORD_CHANGES_DAY** | Average user password change events per second for the current day. |
| **PASSWORD_CHANGES_HOUR** | Average user password change events per second for the current hour. |
| **PASSWORD_CHANGES_MINUTE** | Average user password change events per second for the current minute. |
| **PWMDB_READS_DAY** | Average local database read events per second for the current day. |
| **PWMDB_READS_HOUR** | Average local database read events per second for the current hour. |
| **PWMDB_READS_MINUTE** | Average local database read events per second for the current minute. |
| **PWMDB_WRITES_DAY** | Average local database write events per second for the current day. |
| **PWMDB_WRITES_HOUR** | Average local database write events per second for the current hour. |
| **PWMDB_WRITES_MINUTE** | Average local database write events per second for the current minute. |
| **REQUESTS_DAY** | Average client request events per second for the current day. |
| **REQUESTS_HOUR** | Average client request events per second for the current hour. |
| **REQUESTS_MINUTE** | Average client request events per second for the current minute. |
| **SESSIONS_DAY** | Average new client session events per second for the current day. |
| **SESSIONS_HOUR** | Average new client session events per second for the current hour. |
| **SESSIONS_MINUTE** | Average new client session events per second for the current minute. |

# Appendix D: Macro Values

| Macro Name | Description |
|---|---|
| @LDAP:name@ | Lookup the LDAP value of the user's LDAP attribute name. |
| @LDAP:name:length@ | Lookup the LDAP value of the user's LDAP attribute name. If the value is longer than length, then truncate the value to the specified length. |
| @LDAP:name:length:padding@ | Lookup the LDAP value of the user's LDAP attribute name. If the value is longer than length, then truncate the value to the specified length. If the value is shorter than length, then pad the value with the value of length. |
| @LDAP:DN@ | Replace with LDAP value of user's LDAP Distinguished Name |
| @User:PwExpireTime@ | Time user's password will expire in default ISO format. |
| @User:PwExpireTime:pattern@ | Time user's password will expire where pattern is a SimpleDateFormat pattern |
| @User:DaysUntilPwExpire@ | Number of days until the user's password will expire |
| @User:ID@ | User's UserID (if authenticated) |
| @User:Email@ | User's Email Address |
| @User:Password@ | User's current password (if authenticated). Use caution, this will allow password to appear in logs and whichever function the macro is used in. |
| @InstanceID@ | Instance ID of the application |
| @CurrentTime@ | Current time in default format. |
| @CurrentTime:pattern@ | Current time where pattern is a SimpleDateFormat pattern |
| @CurrentTime:pattern:tz@ | Current time where pattern is a SimpleDateFormat pattern, and the timezone is a tz specified as a valid TimeZone ID. |
| @Site:URL@ | URL of the site (http://www.example.com/password) |
| @Site:Host@ | Hostname of the site (www.example.com) |
| @RandomChar@ | A single random character of visible upper & lower ASCII characters and digits. |
| @RandomChar:length@ | Random characters, where length is the number of random characters to generate. |
| @RandomChar:length:characters@ | Random characters, where length is the number of random characters to generate and characters is the list of characters to be used as random characters. |
| @Encode:type:[[value]]@ | Encode a value using the specified type of encoding, where type is the type of encoding and where value is the value to encode. The value may include other macros. Types permitted are urlPath, urlParameter and base64. |
| @Hash:type:[[value]]@ | Hash a value using the specified hash type, where type is the type of hash and where value is the value to hash. The value may include other macros. Hash types permitted are md5, sha1, sha256, and sha512. |

**Macro Examples:**

| Macro String | Expanded Macro Value |
|---|---|
| jason/@mail.com | jason@mail.com |
| http/:////www.microfocus.com | http://www.microfocus.com |
| @LDAP:givenName@ | Jason |
| @LDAP:givenName:3@ | Jas |
| @LDAP:givenName:8:-@ | Jason--- |
| @RandomChar:8:0123456789ABCDEF@ | 4B420A120F |
| @LDAP:givenName:1@@LDAP:sn:7@@RandomChar:2:0123456789@ | JSmith84 |
| @LDAP:givenName:1@@LDAP:sn:7:0@@RandomChar:2:0123456789@ | JSmith0084 |
| The time is @CurrentTime:EEE, MMM d, yyyy@. | The time is Sat, Jan 1, 2000. |

# Appendix E: Challange/Response Profile Attributes

## Profile Attributes

**Attribute** **type**

**Minimum Password Required** *integer*

Specifies the minimum number of responses, to random questions, the user is required to submit when attempting to recover from a forgotten password.

Note that the number of random questions presented to the user may be greater than this value.

**Minimum Random Challenges Required During Setup** *Integer*

Specifies the minimum number of random questions that a user is required to complete during the Response Setup. If this number is higher than the available randoms, or lower than the minimum required, SSPR adjusts it accordingly. Setting the value to zero will force users to configure all available random questions at the time of setup.

## Profile Challenge/Response Attributes

**Attribute** **type**

**challengeText** *UTF-8 string*

Text of the challenge/response question, to which an *answer* will be supplied to verify.

**minLength** *Integer*

The minimum length of the user-supplied response string.

**maxLength** *Integer*

The maximum length of the user-supplied response string.

**adminDefined** *boolean*

Indicates if value of the *challengeText* attribute, the challenge question, was supplied by the system administrator, or if it was supplied by the user.

**required** *boolean*

Indicates if this challenge is *required*, or *random*.

Users must correctly answer all *required* challenges to be validated.

Random challenges are placed in a pool; where the user may be required to correctly answer a specified number of them in order to be validated.

**enforceWordlist** *boolean*

Indicates if the answer will be rejected if any of the words in the answer are found in the configured global word-list:

Settings → Word Lists → Word List File URL

**maxQuestionCharsInAnswer** *integer*

Maximum number of characters from the challenge question that are permitted in the challenge response.

# Appendix F: Locales (Languages) and Flags

List of default locales available in SSPR 4.3:

| Web Browser Locale Code | ISO Alpha-2 Country Code | Language |
|---|---|---|
| **en** [*1] | *us* | English (United States) |
| **en_CA** | *ca* | English (Canada) |
| **ca** | *es* | Catalan |
| **da** | *dk* | Danish |
| **de** | *de* | German (Standard) |
| **es** | *es* | Spanish |
| **fr** | *fr* | French (Standard) |
| **fr_CA** | *ca* | French (Canada) |
| **iw** [*2] | *il* | Hebrew (Israel) |
| **it** | *it* | Italian (Standard) |
| **ja** | *jp* | Japanese |
| **nl** | *nl* | Dutch (Standard) |
| **pl** | *pl* | Polish |
| **pt_BR** | *br* | Portuguese (Brazil) |
| **ru** | *ru* | Russian |
| **sv** | *se* | Swedish |
| **zh_CN** | *cn* | Chinese (PRC) |
| **zh_TW** | *tw* | Chinese (Taiwan) |

[*1] The empty string language code "" in SSPR is treated as "en" English (United States).

[*2] "iw" from the older ISO 639:1988 standard. Newer ISO standards have replaced "iw" with "he".

SSPR's default list of *Locales (Languages) and Flags* can be augmented and modified in the *SSPR Configuration Editor*:

Settings → Application → Localization → Locales (Languages) and Flags

In the *SSPR Configuration Editor*, the code is in two parts separated by two colons (::). The first part is the *browser locale code*, the second field is the *iso country code* used for the flag value.

# Appendix G: Password Policy Attributes

SSPR requires that each user have an assigned *password policy*, which consists of various policy attributes, defined here.

**Attribute** *Type* Default Value

**ADComplexityMaxViolations** *string (integer)* 2

Specifies the maximum number of *Active Directory 2008 Level Complexity* category violations.

This setting has no effect unless *Active Directory Password Complexity* is set to *AD 2008 Level Complexity* in the *Configuration Editor*.

Policies → Password Policies → *[profile]* → Active Directory Password Complexity = "AD 2008 Level Complexity"

**AllowFirstCharNumeric** *string (boolean)* true

Indicates if the first character of the password is allowed to be numeric. Only applies if the password policy attribute *AllowNumeric* value is *true*.

**AllowFirstCharSpecial** *string (boolean)* true

Indicates if the first character of the password is allowed to be a special character. Only applies if the password policy attribute *AllowSpecial* value is *true*.

**AllowLastCharNumeric** *string (boolean)* true

Indicates if the last character of the password is allowed to be numeric. Only applies if the password policy attribute *AllowNumeric* value is *true*.

**AllowLastCharSpecial** *string (boolean)* true

Indicates if the last character of the password is allowed to be a special character. Only applies if the password policy attribute *AllowSpecial* value is *true*.

**AllowMacroInRegExSetting** *string (boolean)* true

In the SSPR *Configuration Editor* it is possible to specify a Regular Expression pattern the policy user password must match...

Policies → Password Policies → Required Regular Expression Matches

or must not match...

Policies → Password Policies → Disallowed Regular Expression Matches

to be valid.

The *AllowMacroInRegExSetting* object indicates if these Regular Expressions may contain SSPR macro values. See Appendix D for more information on SSPR macro values.

Normally, the value of this object will be *true*. The value can only be manipulated under the direction of NetIQ's technical support.

**AllowNumeric** *string (boolean)* true

Indicates if numeric characters are allowed in the password.

**AllowSpecial** *string (boolean)* true

Indicates if special (non alpha-numeric) characters are allowed in the password.

**CaseSensitive** *string (boolean)* true

Indicates if passwords are (upper/lower) case sensitive.

The value of this field is not manipulated within a *password policy*. Rather, the value is maintained globally by SSPR. The value is manipulated in SSPR's *Configuration Editor*:

Settings → Password Settings → Password is Case Sensitive

**Attribute** *Type* Default Value

**ChangeMessage** *string*

Specifies a message that is displayed to the users during password changes. May include HTML markup. This value is overridden by the existence of a *change password message read* of an LDAP password policy.

**CharGroupsMinMatch** *string (integer)* 0

Specifies the minimum number of *regular expression* matches required for a valid password value. The regular expressions are defined in the *CharGroupsValues* profile attribute.

**CharGroupsValues** *string (multi-line text)* : ".*[0-9]\n.*[^A-Za-z0-9]\n.*[A-Z]\n.*[a-z]

Specifies a *newline delimited* list of LDAP filter expressions that contains a list of regular expression character matches. Used in conjunction with the *CharGroupsMinMatch* profile attribute to allow the creation of a complex list of requirements that valid user passwords need only to partially match.

For example, this type of policy can replicate the Active Directory *3 out of 5* rules, but with more flexibility and customization.

**DisallowCurrent** *string (boolean)* true

Specifies if it is prohibited to re-use the current LDAP password as a new password value.

*In order for this rule to be enforced, it must be permitted for SSPR to read the user's current LDAP password.*

**DisallowedAttributes** *string (multi-line text)* givenName\ncn\nsn

Specifies a *newline delimited* list of LDAP attributes, who's values not allowed to be used as passwords. For a given user, SSPR reads the listed values and does not permit the users to use it as part of the password value. This check is case-insensitive.

Note: Specifying a number after the attribute name restricts how many consecutive characters PWM disallows in the value. For example: "Language:4" means the password cannot contain: "Engl", "ngli", "glis", or "lish", for English speaking users

**DisallowedValues** *string (multi-line text)* password\ntest

Specifies a *case insensitive*, *newline delimited*, list of values are not allowed as passwords.

**EnableWordlist** *string (boolean)* true

Indicates if passwords will be disqualified if they are present in the SSPR's global word list. configured *Word List*.

**ExpirationInterval** *string (integer)* 0

Indicates the number of seconds that a user's password will remain valid before a password change will be required. A value of zero disables this check.

**MaximumAlpha** *string (integer)* 0

Specifies the maximum number of alphabetic characters required. A value of zero disables this check.

**MaximumConsecutive** *string (integer)* 0

Specifies the maximum number of characters in a sequence allowed in a password. For example, sequences such as *0123456789* or *abcdefghijk*. More specific character sequences are defined by the unicode character order of each character after it converts the entire value to lowercase. A value of 0 disables this check.

**Attribute** *Type* Default Value

**MaximumLength** *string (integer)* 12

The maximum length of the password. A value of zero disables this check.

Although SSPR allows large values, the underlying LDAP directory may have limitations on the maximum supported password length.

**MaximumLowerCase** *string (integer)* 0

Specifies the minimum number of lowercase characters required. A value of zero disables this check.

**MaximumNonAlpha** *string (integer)* 0

Specifies the maximum number of non-alphabetic characters required. A value of zero disables this check.

**MaximumNumeric** *string (integer)* 0

Specifies the maximum amount of numeric characters required (if the password policy allows numeric). A value of zero disables this check.

**MaximumRepeat** *string (integer)* 0

Specifies the maximum amount of times users may repeat any character throughout the password. A value of zero disables this check.

This check ignores character (upper/lower) case.

**MaximumSequentialRepeat** *string (integer)* 0

Specifies the maximum times users may sequentially repeat any character throughout the password. A value of zero disables this check.

This check ignores character (upper/lower) case.

**MaximumSpecial** *string (integer)* 0

Specifies the maximum number of special characters required (if the password policy allows special characters). A value of zero disables this check.

**MaximumUpperCase** *string (integer)* 0

Specifies the maximum number of uppercase characters required. A value of zero disables this check.

**MinimumAlpha** *string (integer)* 0

Specify the minimum number of alphabetic characters required. A value of zero disables this check.

**MinimumLength** *string (integer)* 4

Minimum length of the password. A value of zero disables this check.

**MinimumLifetime** *string (numeric)* 0

Specifies the minimum number of seconds that must pass between password changes. A value of zero disables this check.

**MinimumLowerCase** *string (integer)* 0

Specifies the maximum number of lowercase characters required. A value of zero disables this check.

**MinimumNonAlpha** *string (integer)* 0

Specifies the minimum number of non-alphabetic characters required. A value of zero disables this check.

**Attribute** *Type* Default Value

**MinimumNumeric** *string (integer)* 0

Specifies the minimum amount of numeric characters required (if numeric characters are allowed). A value of zero disables this check.

**MinimumSpecial** *string (integer)* 0

Specifies the minimum number of special characters required (if the password policy allows special characters). A value of zero disables this check.

**MinimumStrength** *string (integer)* 0

Specifies the minimum strength required for valid passwords. SSPR judges the password strengths on a strength on a scale of 0 to 100 irrespective of other password policy settings.

| **Value** | **Meaning** |
|---|---|
| 0 | Check is disabled |
| 45 | Good |
| 70 (or greater) | Strong |

**MinimumUnique** *string (numeric)* 0

Specifies the minimum number of unique characters allowed. A value of zero disables this check.

**MinimumUpperCase** *string (integer)* 0

Specifies the minimum number of uppercase characters required. A value of zero disables this check.

**PolicyEnabled** *string (boolean)* true

**RegExMatch** *string (regex)*

Specifies a Regular Expression pattern the password must match to be valid. Multiple patterns may be specified. A pattern must match the entire password to be applied. Partial matches are ignored. Macros may be used.

**RegExNoMatch** *string (regex)*

Specifies a Regular Expression pattern the password must not match to be valid. Multiple patterns may be specified. A pattern must match the entire password to be applied. Partial matches are ignored. Macros may be used.

**UniqueRequired** *string (boolean)* false

### Active Directory Password Complexity

This value can be manipulated in the *Configuration Editor*:

Policies → Password Policies → *[profile]* → Active Directory Password Complexity

Options include:

**None - Do not enforce AD Complexity Rules**

**AD 2003 Level Complexity:**

- Cannot contain the user's account name or parts of the user's full name that exceed two consecutive characters.
- Minimum 6 characters.
- Maximum 128 characters.
- Must contain characters from three of the following four categories:

English uppercase characters (A through Z).
English lowercase characters (a through z).
Base 10 digits (0 through 9).
Non-alphabetic characters (For example, !, $, #, %).

**AD 2008 Level Complexity:**

- Cannot contain the user's account name or parts of the user's full name that exceed two consecutive characters.
- Minimum 6 characters.
- Maximum 512 characters.
- Must contain characters from three of the following four categories:

European language uppercase alphabetic characters.
European language lowercase alphabetic characters.
Base 10 digits (0 through 9).
Non-alphabetic characters (for example, !, $, #, %).
Other alphabetic characters not included in the other categories.

# Appendix H: User Information

**Field Name** — **Type**

**accountExpirationTime** *ISO 8601, Combined date and time, Coordinated Universal Time*

Indicates the time when the user's account is set to expire.

**lastLoginTime** *ISO 8601, Combined date and time, Coordinated Universal Time*

Indicates the time when the target user successfully logged in to SSPR.

**ldapProfile** *string*

The LDAP profile of the target SSPR user.

**passwordExpirationTime** *ISO 8601, Combined date and time, Coordinated Universal Time*

Indicates the time when target user's password will expire.

**passwordLastModifiedTime** *ISO 8601, Combined date and time, Coordinated Universal Time*

Indicates the time when the target user's password was last modified.

**passwordStatus** *object*

**expired** *boolean*

Indicates that the target user's current password value has expired and is no longer valid.

**preExpired** *boolean*

Indicates that the target user's current password is set to expire very soon. If the user attempts to login, a password change will be required.

**violatesPolicy** *boolean*

Indicates that the target user's current password value violates LDAP or SSPR policy.

**warnPeriod** *boolean*

Indicates that the target user's current will expire soon. If the user attempts to login, a warning will be displayed, requesting that the user change the password as soon as convenient.

**requiresInteraction** *boolean*

Indicates if SSPR will require target user interaction at the next login attempt.

This value will be *true* if *requiresOtpConfig*, *requiresUpdateProfile*, *requiresResponseConfig* or *requiresNewPassword* are true.

**requiresNewPassword** *boolean*

Indicates if SSPR will require the target user to set a new password at the next login attempt.

**requiresOtpConfig** *boolean*

Indicates if the user has been configured to require a *One-Time Password* enrollment.

**requiresResponseConfig** *boolean*

Indicates if SSPR will require the target user to configure the challenge/response configuration at the next login attempt.

**Field Name** **Type**

**requiresUpdateProfile** *boolean*

Indicates if SSPR will require the target user to update their profile at the next login attempt.

**userDN** *string*

The target user's LDAP *Distinguished Name* (DN).

**userEmailAddress** *string*

The email address of the target user.

**userEmailAddress2** *string*

Alternate email address of the target user.

**userEmailAddress3** *string*

Second alternate email address of the target user.

**userGUID** *string*

The SSPR user GUID of the target user.

**userID** *string*

The SSPR user ID of the target user.

**userSmsNumber** *string*

The SMS phone number of the target user.

**userSmsNumber2** *string*

Alternate SMS phone number of the target user.

**userSmsNumber3** *string*

Second alternate SMS phone number of the target user.

# Appendix I: Known Issues

| Bug # | Description |
|---|---|
| 1087169 | REST: /sspr/public/rest/challenges POST Username in queryString: "7000 ERR_REST_INVOCATION_ERROR" |
| 1084033 | REST: /sspr/public/rest/challenges DELETE Failure for x-www-form-urlencoded Content-Type. |
| 1087344 | REST: /sspr/public/rest/challenges DELETE Fails when target is the REST user. |
| 1087773 | REST: /sspr/public/rest/challenges DELETE Fails when target is the REST user. |
| 1087776 | REST: /sspr/public/rest/profile POST No username parameter: LDAP Err 50 |
| 1087797 | REST: /sspr/public/rest/randompassword POST Generates invalid passwords. |
| 1087993 | REST: /sspr/public/rest/signing/form POST Content-Type .../x-www-form-urlencoded. |
| 1089267 | REST: /sspr/public/rest/statistics GET Remove "statName" parameter. |
| 1089394 | REST: /sspr/public/rest/statistics GET Returned "key data" lists statistic "variable names", not "keys". |
| 1084214 | REST: /sspr/public/rest/verifyresponses POST reports: 5015 ERROR_UNKNOWN |