

Login Snippet

This folder contains the code for the login snippet. `login_snippet.jsp` and `login_snippet.js` contain the JavaScript code for the login snippet and `login_snippet.css` contains the styles. You only need to use one of `login_snippet.jsp` or `login_snippet.js`, not both. If you choose to use `login_snippet.jsp` it must be placed in the JSP folder of your IDP server (ex: `/opt/novell/nids/lib/webapp/jsp/`). The `.jsp` version of the snippet contains translations for all strings and will check if users are authenticated before loading the snippet. The `.js` version only provides strings in English and assumes that the user is not already authenticated. This folder also contains `'b2c_nam_login.jsp'`, this is used to have NAM login page branding with two factor authentications or any follow-up method to be used with contract. Do refer to the section "Custom Branding" below

Example

The `/example` sub-directory contains a simple page that uses the login snippet. To configure it follow these steps:

- 1) Edit `/example/example.html` replacing the "apiHost" value with a valid Access Manager URL. 2) Save the file.
- 3) Open the file in a browser.
- 4) If you see an "An error occurred" message in the upper right side of the page, possible error are:
 - a. The "apiHost" URL is incorrect. A valid URL format is: `https://internal.server.com:8443`.
 - b. The Access Manager is not started or inaccessible from the client workstation.
 - c. The browser does not trust the Access Manager certificate.

To troubleshoot these error conditions, navigate directly to the "apiHost" URL in a second browser window. You should see the Access Manager login page. If you get a security warning, accept the server certificate.

Installation

The script must be loaded on your page before being invoked. The invocation is accomplished by calling `namLogin()` with a valid configuration. The following is an example of how to invoke the snippet with the most minimal configuration:

```
<script src="https://prvdrvnam001.namdom006.lab:8443/nidp/jsp/login_snippet.jsp"></script>
<script>    namLogin({        apiHost:
'https://prvdrvnam001.namdom006.lab:8443'
    });
</script>
```

A more complex configuration might look like this:

```
<script src="https://prvdrvnam001.namdom006.lab:8443/nidp/jsp/login_snippet.jsp"></script>
<script>
namLogin({
  apiHost: 'https://prvdrvnam001.namdom006.lab:8443',
  loginRedirectUrl: 'http://www.microfocus.com',
  passwordLabelPlaceholder: false,
  usernameLabelPlaceholder: false,      parent:
  'logincontainer',      text: {
  usernameLabel: 'Email',      submit: 'Submit'
    }
});
</script>
```

Configuration

The following are the names, descriptions, and default values of the configuration properties recognized by the login snippet. 'apiHost' is the only required option.

```

    // REQUIRED. IDP Server host (ex: https://prvcdvnam001.namdom006.lab:8443). If not specified, it
    is expected that 'loginUrl' be provided      apiHost: null,

    // CSS class names used by elements in the generated HTML
    classNames: {
        authProviderList: 'nam-auth-provider-list',
        loginContainer: 'nam-login-container',
        contract: 'nam-contract',            dirty: 'namdirty',
        error: 'nam-error',                errorMessage: 'nam-
        message',                formActions:
        'nam-actions',
        inputContainer: 'nam-input-container'
    },

    // Element ids for elements in the generated HTML
    ids: {
        form: 'nam-form',                passwordInput:
        'nam-password',                usernameInput: 'nam-
        username',                loginSubmit:
        'nam-login-submit'
    },

    // Submit login form through AJAX request instead of a full browser POST. A full browser
    POST does not allow for any processing after login attempts      loginUsingAjax: false,

    // IDP Server path that authenticates username and password credentials
    loginUrl: null,

    // Redirect user to another page after successful login. By default, this URL is extracted
    from the HTTP response that is received after successfully logging in.      loginRedirectUrl: null,

    // Javascript function to be executed after failed login.
    loginFailureCallback: null,

    // Javascript function to be executed after successful login. Default success callback simply
    redirects to the provided loginRedirectUrl or the URL received from the server.
    loginSuccessCallback: null,

    // String or HTMLElement reference for the element under which the login HTML should be added
    parent: document.body,

    // SSPR 'Forgot Password' URL. Overrides the value that is returned from the REST endpoint
    forgotPasswordUrl: null,

    // Open 'Forgot Password' link in a new tab. Set to false to open in the same browser tab.
    forgotPasswordLinkNewTab: true,

    // Show label for password input as [placeholder] attribute instead of <label> element. Set to
    false to use label.      passwordLabelPlaceholder: true,

    // SSPR 'Register' URL. Overrides the value that is returned from the REST endpoint
    registerUrl: null,

    // Open 'Register' link in a new tab. Set to false to open in the same browser tab.
    registerLinkNewTab: false,

```

```
// Provide social authentication links. This property is an array of objects.
```

```

// The objects should be of the form { url: String, name: String, icon: String }

// (ex: {
//   url:
"https://prvdrvnam001.namdom006.lab:8443/nidp/app/login?sid=1&sid=1&authid=facebook&option=credential",
//   name: "FACEBOOK",
//   icon: "https://fbstatic-a.akamaihd.net/rsrc.php/yl/r/H3nkt0a7ZMg.ico"
// }
// )    socialAuthProviders:
null,

// Open social authentication links in a new tab. Set to false to open in the same browser tab.
socialLinksNewTab: true,

// Show label for username input as [placeholder] attribute instead of <label> element. Set
to false to use label.    usernameLabelPlaceholder: true,

// Text used by the generated login HTML. These strings are translated when the login
snippet is served from /nidp/jsp/login_snippet.jsp    text: {    forgotPassword:
'Can\'t sign in?',    loginFailedError: 'Login failed, please try again.',
passwordLabel: 'Password',    requiredField: 'Required',    register: 'Register',
socialTitle: 'Or use a secure social account',    submit: 'Sign In',    title:
'Sign In',    usernameLabel: 'Username'    }

```

Stylesheets

A default stylesheet is provided at '/nidp/css/login_snippet.css'.

```

<link rel="stylesheet" href="https://prvdrvnam001.namdom006.lab:8443/nidp/css/login_snippet.css">

```

You are also free to ignore or override any of these styles with a custom stylesheet.

Custom Branding

When NAM is configured with custom branding and in some scenario after B2C login form submit using login snippet, branding might be lost on following authentication methods like two factor authentications. To overcome this situation, form submit URL must be override with new one.

To do this follow these steps,

1. Copy the 'b2c_nam_login.jsp' to IDP at "/opt/novell/nam/idp/webapps/nidp/jsp" folder
2. Add 'b2c_nam_login.jsp' to public access list of JSP in web.xml
3. Add the following to web.xml on IDP at "/opt/novell/nam/idp/webapps/nidp/WEB-INF/web.xml"

```

<servlet>
    <servlet-name>B2CLoginServlet</servlet-name>
    <display-name>B2C Login Servlet</display-name>
    <jsp-file>/jsp/b2c_nam_login.jsp</jsp-file>

```

```
</servlet>
<servlet-mapping>
    <servlet-name>B2CLoginServlet</servlet-name>
    <url-pattern>/app/b2c_login</url-pattern>
</servlet-mapping>
```

4. Restart IDP
5. Configure login snippet parameter loginUrl with new login URL. Please do refer above for login snippet configuration parameters. Example below.
loginUrl: 'https://<IDP DNS name>:<port>/nidp/app/b2c_login'