AEM Mobile: Setting up Generic Identity Provider

Requirement:

Prerequisite knowledge

- Understanding of AEM Mobile

Required Products

- AEM Mobile

Adding a Generic Identity Provider

To set up Generic identity provider in AEM Mobile On-Demand Services, you need to provide the URL location of the sign-in UI that handles authentication requests. When users select "Sign In" from the AEM Mobile app, they are redirected to the provided sign-in UI that uses query parameters.

Query Parameters:

- **redirectUrl** – the callback URL that the sign-in UI will need to use to redirect the user to the app, after either a successful or failed sign-in
- **projectId** – the On-Demand Services project ID
- **appId** – the app bundle ID (example: com.adobe.test)
- **appVersion** – the version of the Runtime (example: 1.0.1)
- **uuid** – the device UUID

Query Format (requires URL decoding):

```
?redirectUri=https%3A%2F%2Fes.publish.adobe.com%2FgenericAuth&projectId
={project_id}&appId={app_bundle_id}&appVersion={app_version}&uuid={devi
ce_uuid}
```

Handling Successful Login

When users successfully log into the sign-in UI, the UI will need to generate an authentication token based on the user credentials and pass that token back in the callback URL as a query parameter. The authentication token will be used in the regular entitlement API (v2) workflow: /RenewAuthToken and /entitlements.

Query Parameters:

- **authToken** – the authentication token that the sign-in UI generates based on user credentials
- **expiresIn** (optional) – the duration (in seconds) before the authentication token expires.

Query Format:

```
https://es.publish.adobe.com/genericAuth?authToken={auth_token}
```

Handling Login Failure

When users fail to log in to the sign-in UI, the UI can either display the error message directly in the UI and allow users to retry, or close the UI and redirect the users back into the AEM Mobile app with a default error message, such as "An error occurred signing you in. Please try again later."

Query Parameter:

- **error** (optional) – the custom error message. Currently, this is being used purely for debugging purposes.

Note: Both authToken and error cannot be passed as parameter in the callback URL. If both or none are provided in the redirect callback, then the app will treat it as a login failure with default error message.

Sample Code

The following is an example for a Generic Identity Provider written in HTML:

```
<!DOCTYPE html>
<html>
<head>
      <title>Generic Identity Provider</title>
</head>
<body>
<!-- TODO: update login UI for users to provide their credentials -->
<form onsubmit="event.preventDefault(); login();">
      Username: <input type="text" />
      Password: <input type="text" />
      <input type="submit" />
</form>
<script>
// parse the URL query parameter,
// the .substring(1) will remove the "?" in the search string
//
i.e. ?redirectUri=https%3A%2F%2Fes.publish.adobe.com%2FgenericAuth&proj
ectId={project_id}&appId={app_bundle_id}&appVersion={app_version}&uuid=
{device_uuid}
var urlParam = window.location.search.substring(1);
// the URL query parameter is encoded, so you should decode it before
using
// before: https%3A%2F%2Fes.publish.adobe.com%2FgenericAuth
// after: https://es.publish.adobe.com/genericAuth
urlParam = decodeURIComponent(urlParam);

// store the parameters
var appData = {};
if (urlParam !== '') { // fail-safe
      var param, paramList = urlParam.split('&');
      for (var index in paramList) {
            param = paramList[index].split('=');
            if (param.length === 2) { // fail-safe
                  appData[param[0]] = param[1];
```

```
            }
      }
}

// simple handler for the login
function login() {
      // TODO: add logic to validate user credentials and check back
backend
      var success = true;
      // TODO: update the auth token with the one generated from the
backend,
      // this will be used in the entitlement v2 API: /RenewAuthToken,
/entitlements
      var authToken = 'AUTH_TOKEN_BASED_ON_USER_LOGIN';
      //
https://es.publish.adobe.com/genericAuth?authToken=eab963bf37d24eed1112
56792d6094de
      if (success && authToken) { // login succeeded
            // redirect the user back to the Runtime using the redirect
callback,
            // append the authToken as a URL query parameter,
            // if not provided, the Runtime will treat this as a login
failure
            window.location = appData.redirectUri + '?authToken=' +
authToken;
      } else { // login failed
            // TODO: do one of the following:
            // 1) provide custom error message within this login UI, or
            // 2) redirect user back to the Runtime without the
'authToken' or with 'error'
            window.location = appData.redirectUri + '?error="Internal
Error"';
      }
}
</script>
</body>
</html>
```

Legal Notice