

---

## RoleSync

---

### Product Information

<b>Category</b>	RoleSync
<b>Sub-category</b>	
<b>Website</b>	
<b>Product version</b>	
<b>Documentation</b>	
<b>API Documentation</b>	

---

### About RoleSync

The RoleSync web service provides the link between the Customer Roles defined on the institution's backend systems, and the App Roles, defined in App Manager in the campusM Cloud.

Roles can be retrieved from a single source of data, including Authentication and Student Records systems. These Customer Role(s) determines the set of tiles and services that will be available to the end user.

If a Customer Role changes on the institution's backend system or is no longer applicable, the SetRolesUpdated service is invoked on the campusM cloud. The next time the end user enters the app, the RoleSync web service is triggered to update the App Roles for the end user.

RoleSync requires a customer [Connect Layer](#).

For more information on using Roles in campusM app manager, refer [here](#).

---

### Roles API

The minimal requirement for setting up RoleSync is an API call to get a list of the user roles. To ensure that campusM allows the maximum flexibility for institutions to provide their user roles, there is no single predefined schema for the API. For instance, institutions can have roles defined within AD/LDAP attributes, within a database view or available web services that can be queried.

Below are steps for a native app login process, covering where the Roles API fits into the standard login process.

1. User: Installs and opened the campusM app.
2. User: Selects the relevant profile and enters their credentials when prompted.
3. User: Clicks the Login button.

4. App: Calls the login/roles service deployed by the campusM team on the Connect Layer server.
5. campusM Connect Layer server: Authenticates the user against the institution's AD/ LDAP server.
  - If step 5 is successful, the Roles API is called, to get a list of user roles.
  - If step 5 is unsuccessful, the app returns a Not Authorised response.
6. campusM Connect Layer server: If step 6 is successful, returns back to the app with a successful response, including the user roles.
7. campusM Connect Layer server: If step 6 is unsuccessful, no roles are returned. The App shows content only from the Default role

Note that each time a user logs in, the institution Roles API will be utilized. To ensure its stability, the server hosting the available API or web service should be as resilient as the institution's AD/LDAP server.

## Integration Overview

The campusM integration for RoleSync supports the following functionality:

- Allow users to view the set of tiles services applicable to their Customer Role(s)

## Integration Method

- REST Service
- Database View
- LDAP Attribute

## Prerequisites

There are several prerequisites to the campusM integration for RoleSync beyond the Connect Layer. The below details the prerequisites for each of the 3 options (institutions will typically choose one of these options to implement):

### REST Service integration method

Prerequisite	Additional Information
Provide the Base URL	
Authentication Options: <ol style="list-style-type: none"> <li>1. No authentication</li> <li>2. Base64 username and password (system-level username/password which can get info on behalf of each user)</li> <li>3. Bearer token (system-level token which can get info on behalf of each user)</li> </ol>	

Prerequisite	Additional Information
Indicate which method (GET or POST) and parameter values to use to retrieve user identifier (typically username, email address or student ID)	<ul style="list-style-type: none"> <li>Both GET and POST methods are supported <ul style="list-style-type: none"> <li>If using GET, send the parameters in the URL e.g. <a href="https://www.exlibrisgroup.com?username={}">https://www.exlibrisgroup.com?username={}</a></li> <li>If using POST, send the parameters in the JSON body e.g. {username:&lt;username&gt;}</li> </ul> </li> <li>There is no set limitation on the number or parameter names</li> <li>The parameter value can either be username, LDAP attribute, token attribute or constant (hard coded string) <ul style="list-style-type: none"> <li>Note: A pair parameter name and value can be defined</li> </ul> </li> </ul>
Ensure response is returned (only) in the following JSON format, keeping the array structure even for single value.	<code>{"roles": ["role1", "role2"]}</code>
Ensure that the APIs and web services are available for calls from the Connect Layer server(s)	
Provide a minimum of <b>three (3) test accounts</b> containing comprehensive and typical data, including the following: <ul style="list-style-type: none"> <li>Valid log in credentials for campusM</li> <li>Different roles for each test user</li> </ul>	

## Database view integration method

Prerequisite	Additional Information
Provide <b>Database Connection Details</b> : <ul style="list-style-type: none"> <li>Hostname</li> <li>Port</li> <li>Database name</li> <li>Username</li> <li>Password</li> </ul>	
Ensure the <b>Connect Layer server(s) can connect to the database</b>	Required to allow Ex Libris to develop and test the created web services
The database view must include the following columns: <ul style="list-style-type: none"> <li>User ID</li> <li>User roles</li> </ul>	One column for user roles is required per user and all roles must be stored in a single column separated by a comma (',')

Prerequisite	Additional Information
Provide a minimum of <b>three (3) test accounts</b> containing comprehensive and typical data, including the following: <ul style="list-style-type: none"> <li>Valid log in credentials for campusM</li> <li>Different roles for each test user</li> </ul>	

## LDAP property integration method

Prerequisite	Additional Information
Provide the <b>LDAP attribute name</b> that will be used to retrieve the user roles	<ul style="list-style-type: none"> <li>A suffix can be used (end with comma) e.g. *CN=LIB-GRP-CAMPUSM-UG,*OU=Managed,OU=LIB-Groups,OU=LIB,OU=MQ-BusUnit-Res</li> <li>If more than one role exists, repeat the property</li> </ul>
<ul style="list-style-type: none"> <li>Provide a minimum of <b>three (3) test accounts</b> containing comprehensive and typical data, including the following:               <ul style="list-style-type: none"> <li>Valid log in credentials for campusM</li> <li>Different roles for each test user</li> </ul> </li> </ul>	

## User Experience

Users will be able to view the set of tiles and services relevant to their user roles in the iOS and Android apps as well as on the web portal.

## Sample Request

```
https://<roleswebservice.university.edu>/rest/roles \
-X POST \
--data "client_credentials&username=campusm150" \
--user "sys-wsclient:SERVICE_PASSWORD"
```

## Sample Response

```
{"roles":["student","medical-student","second-year"]}
```

## Offline Functionality

Not supported.

---

## Configuration Options

N/A

---

## Updating User Roles via campusM API

When a user role has changed, the institution needs to call the campusM setRolesUpdated API service. The following steps describe the update process:

1. The user is currently logged in to the app.
  2. User roles have changed—for example, a new role was added, or an existing role removed.
  3. Institution: Calls the campusM setRolesUpdated API, including the user email address.
  4. The user launches the app.
  5. The app calls the startup service on campusM, which informs the app that the roles of this user require an update.
  6. The app calls the login service on the campusM Connect Layer, triggering the institution Roles API to retrieve the user's new roles.
  7. The app presents the content and tiles according to the new user roles.
- 

## Sample SetRolesUpdated Service Request

The request URL should be as follows:

[https://campusm\\_client:<password>@<app URL>/axis2v3/services/CampusMService](https://campusm_client:<password>@<app URL>/axis2v3/services/CampusMService)

e.g. - [https://campusm\\_client:<password>@demo-na.campusm.exlibrisgroup.com/axis2v3/services/CampusMService](https://campusm_client:<password>@demo-na.campusm.exlibrisgroup.com/axis2v3/services/CampusMService)

Note: URL password, orgCode and password should be provided by your Ex Libris project or support contact by opening a CRM Salesforce request.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:cam="http://campusm.gw.com/campusm">
  <soapenv:Header/>
  <soapenv:Body>
    <cam:setRolesUpdatedRequest>
      <cam:orgCode>123</cam:orgCode>
      <cam:password>xyz</cam:password>
    <!--Optional:-->
    <cam:emails>
      <!--1 or more repetitions:-->
      <cam:email>user@campusM.com</cam:email>
    </cam:emails>
    </cam:setRolesUpdatedRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

---

## Suggested Testing Guidelines

The following acceptance criteria is recommend to be used as part of the testing and approval process:

- User can see the tiles that match their user roles