
Retrieve Attendees for Events

Product Information

Category	Retrieve Attendees for Events
Sub-category	
Website	
Product version	
Documentation	
API Documentation	

About the Service

Retrieve Attendees is a webservice that retrieves information on the list of attendees for a given event. This webservice is being used as part of campusM Attendance module, specially for the lecturer check-in to view the list of all attendees against each event on the lecturer screen of attendance page.

The web service is deployed on the campusM connect layer and is a mandatory prerequisite for the customer to enable this service

Integration Overview

This Integration with campusM allows the end users to view list of attendees along with their details for a given event. The details include:

- Student ID (Unique Identifier)
- Email
- Given Name
- Surname

Institutions can choose/configure from the above details that they would like to show/hide as part of the user experience.

Integration Method

- Rest API Service
- Database

Prerequisites

The following are the list of pre-requisites to the campusM Integration for retrieving the events and Attendees using the Rest service and Database integration methods respectively.

1. [For Rest API Service Method](#)

Prerequisite	Additional Information
Provide the Base URL	<p>The base URL should include the parameters that are expected by the REST request. The options for the parameters to send on the URL are: (username, password, eventRef and lecturerId)</p> <p>For example: http://localhost:8080/extensions/getAttendees?user={username}&pass={password}&eventRef={eventRef}&lecturer={lecturerId}</p> <p>Or – http://localhost:8080/extensions/getAttendees/{eventRef}</p> <p>These placeholders will be replaced with actual data so be sure to use these placeholders exactly: ({username}, {password}, {eventRef}, {lecturerId})</p> <p>The username, password and eventRef when chosen will be taken from the service request, and the lecturerId will be retrieved by the info given in the LecturerID section (see below).</p>
<p>Authentication Options:</p> <ol style="list-style-type: none"> 1. No authentication 2. Base64 username and password (system-level username/password which can get info on behalf of each user) 3. Bearer token (system-level token which can get info on behalf of each user) 	
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"> • Both GET and POST methods are supported <ul style="list-style-type: none"> ◦ For both GET and POST, send the parameters in the URL e.g. http://localhost:8080/extensions/getAttendees?user={username}&pass={password}&eventRef={eventRef}&lecturer={lecturerId}
<p>Ensure response is returned as an array in JSON format</p> <p>Should include the fields:</p> <ul style="list-style-type: none"> • StudentID • Email • First Name • Last Name <p>Note: The fields 'First Name' and 'Last Name' are</p>	<p>Examples:</p> <pre>{ "attendees": [{ "emailAddress": "test1@gmail.com", "studentId": "1234", "givenName": "", "sureName": "program1" }, { "emailAddress": "test2@gmail.com",</pre>

Prerequisite	Additional Information
optional	<pre> "studentId": "5678", "givenName": "test2", "sureName": "program2" }] } Or – [{ "emailAddress": "test1@gmail.com", "studentId": "1234", "givenName": "", "sureName": "program1" }, { "emailAddress": "test2@gmail.com", "studentId": "5678", "givenName": "test2", "sureName": "program2" }] </pre>
Provide an email suffix (constant) - Optional	If there is a need to concatenate an email suffix to some field from the JSON response
Provide lecturerID to send on the URL - Optional	<p>Few options:</p> <ul style="list-style-type: none"> • Use the username as the lecturerID • Use a lecturerID from an ldap attribute • For SSO based authentication – use lecturerID from a SAML/ OAuth attribute (from the token) <p>Note: If the lecturerID is chosen it will replace {lecturerID} in the URL, but only if this placeholder exists in the URL</p>
Ensure that the APIs and web services are available for calls from the Connect Layer server(s)	
Provide a minimum of three (3) test accounts containing comprehensive and typical data, including the following:	<ul style="list-style-type: none"> • Valid log in credentials for campusM • Different students list for each test user

2. For Database View

Prerequisite	Additional Information
<p>Provide Database Connection Details:</p> <ul style="list-style-type: none"> • Hostname • Port • Database name • Username • Password 	
<p>Ensure the Connect Layer server(s) can connect to the database</p>	<p>Required to allow Ex Libris to develop and test the created web services</p>
<p>The database view should include the following columns:</p> <ul style="list-style-type: none"> • StudentID • Email • First Name • Last Name <p>Note: The fields 'First Name' and 'Last Name' are optional</p>	<p>The query should return values mapped to these column names (email, studentid, firstname, lastname), and should include the 'eventRef' as an input parameter (i.e. "where eventref = ?") and could optionally include the lecturerID as an input parameter (i.e. "where/AND lecturerid = ?")</p>
<p>Provide lecturerID to send on the query - Optional</p>	<p>Few options:</p> <ul style="list-style-type: none"> • Use the username as the lecturerID • Use a lecturerID from an ldap attribute • For SSO based authentication – use lecturerID from a SAML/ OAuth attribute (from the token) <p>Note: In the query, the eventRef has to come first, and then the lecturerID (i.e. "where eventref =?" OR "where eventref =? and lecturerid = ?")</p>
<p>Provide a minimum of three (3) test accounts containing comprehensive and typical data, including the following:</p> <ul style="list-style-type: none"> • Valid log in credentials for campusM • Different roles for each test user 	

User Experience

The end user will be able to view the list of all attendees for an event.

Sample Request

Sample Request

```
http://localhost:8080/extensions/getAttendees?user={username}&pass={password}&eventRef={eventRef}&lecturer={lecturerId}
```

Sample Response

Sample Response

```
{
  "attendees": [
    {
      "emailAddress": "test1@gmail.com",
      "studentId": "1234",
      "givenName": "",
      "sureName": "program1"
    },
    {
      "emailAddress": "test2@gmail.com",
      "studentId": "5678",
      "givenName": "test2",
      "sureName": "program2"
    }
  ]
}
```

Configuration Options

Institutions can choose/configure from the below details that they would like to show/hide as part of the user experience.

- Student ID (Unique Identifier)
- Email
- Given Name
- Surname

By default, 'Given Name' and 'Surname' will be displayed in the UI.

Suggested Testing Guidelines

The following acceptance criteria should be considered for testing and validating the functionalities.

- User should be able to view the list of attendees upon selecting an event.