

Integrating with LMExcellence

1 OVERVIEW

LMExcellence Enterprise edition provides a suite of secure RESTful APIs using which organization could improve interoperability between LMExcellence and other systems through enhanced security and automated data flow.

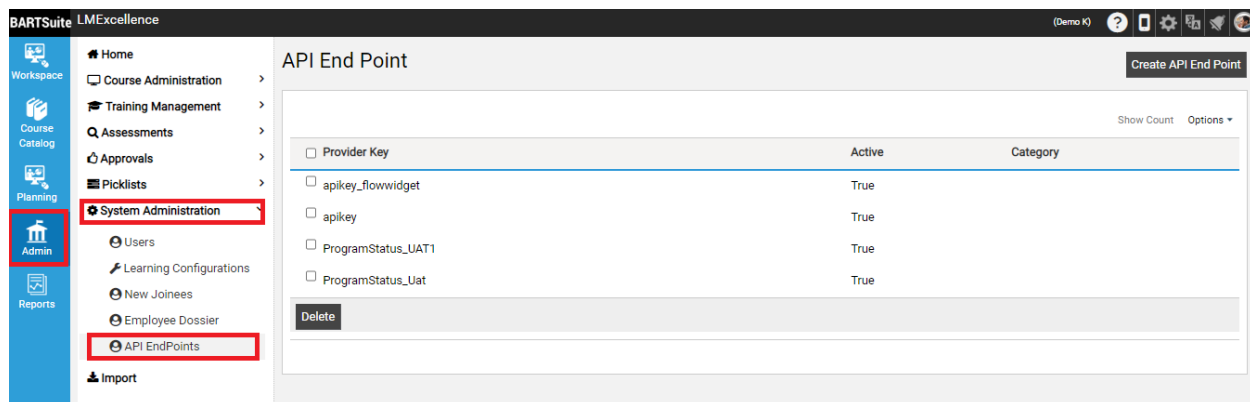
This section of the user manual provides a quick introduction to various integrations, which is possible between LMExcellence as well as other tools, as well as details on working with integration APIs.

As a pre-requisite, your enterprise account has to be enabled to adopt these integrations, contact your administrator or support@lmexcellence.com if you cannot find the keys or relevant features/sections are not accessible in your instance of LMExcellence.



2 GETTING STARTED WITH INTEGRATION

LMExcellence would provide unique keys, which would authorize them to establish connection and manage data movement between systems using structured APIs. Navigate to Admin -> Expand System Administration and click on API EndPoints and you should be able to find your keys.



All the APIs require to provide below information based on different environments

- URL – Url of the Lm Portal.
- HttpMethod – Get/Post
- Tenant ID – Identifier of the tenant
- API Endpoint Key Name and Key Value
- TypeID and Method Name – Will be different per API

All the above parameters need to be specified in every API call.

Sample c# code

```
var url = System.Configuration.ConfigurationSettings.AppSettings[ "url" ];
var key = System.Configuration.ConfigurationSettings.AppSettings[ "apikey" ];
string secret = System.Configuration.ConfigurationSettings.AppSettings[ "apisecret" ];
string tid = System.Configuration.ConfigurationSettings.AppSettings[ "tenantid" ];

var httpWebRequest = ( HttpWebRequest )WebRequest.Create( url );
httpWebRequest.ContentType = "application/json";
httpWebRequest.Method = "POST";

httpWebRequest.Headers.Add( "X-BIZAPP-TID", tid );
httpWebRequest.Headers.Add( "X-BIZAPP-APIKEY", key );
httpWebRequest.Headers.Add( "X-BIZAPP-APISECRET", secret );

httpWebRequest.Headers.Add( "X-BIZAPP-TYPEID", "System1242" );
httpWebRequest.Headers.Add( "X-BIZAPP-METHOD", "UpdateOrCreateUsers" )
```

3 AVAILABLE APIs

Below are the listing of available APIs, their endpoints as well as a brief description on the data being passed to and returned from these APIs –

3.1 MANAGING ORGANIZATION INFORMATION -

API	Endpoint details
Company Info Create/Update	<i>Populate LMExcellence with details pertaining to your company as well as any group companies. Users are bound to the company and the company a user belongs to determines many of the logic such as access to system, visibility to data, etc.</i>
	END POINT.... <pre> var url = "https://demo-staging.lmexcellence.com/apihandler.ashx"; var httpWebRequest = (HttpWebRequest)WebRequest.Create(url); httpWebRequest.ContentType = "application/json"; httpWebRequest.Method = "POST"; httpWebRequest.Headers.Add("X-BIZAPP-TID", tid); httpWebRequest.Headers.Add("X-BIZAPP-APIKEY", key); httpWebRequest.Headers.Add("X-BIZAPP-APISECRET", secret); httpWebRequest.Headers.Add("X-BIZAPP-TYPEID", "System1242"); httpWebRequest.Headers.Add("X-BIZAPP-METHOD", "UpdateOrCreateCompany"); using (var streamWriter = new StreamWriter(httpWebRequest.GetRequestStream())) { string json = GetJSONFromResources(" LM_API_Samples.JSON.CompanyJson.txt"); streamWriter.Write(json); } var httpResponse = (HttpWebResponse)httpWebRequest.GetResponse(); using (var streamReader = new StreamReader(httpResponse.GetResponseStream())) { var result = streamReader.ReadToEnd(); } </pre>
Department Info Create/Update	<i>These are your organizational departments, with employees belonging to one of these departments</i>
	END POINT.... <pre> var url = "https://demo-staging.lmexcellence.com/apihandler.ashx"; var httpWebRequest = (HttpWebRequest)WebRequest.Create(url); httpWebRequest.ContentType = "application/json"; httpWebRequest.Method = "POST"; httpWebRequest.Headers.Add("X-BIZAPP-TID", tid); httpWebRequest.Headers.Add("X-BIZAPP-APIKEY", key); httpWebRequest.Headers.Add("X-BIZAPP-APISECRET", secret); httpWebRequest.Headers.Add("X-BIZAPP-TYPEID", "System1242"); httpWebRequest.Headers.Add("X-BIZAPP-METHOD", "UpdateOrCreateDepartment"); </pre>

	<pre> using (var streamWriter = new StreamWriter(httpWebRequest.GetRequestStream())) { string json = GetJSONFromResources("LM_API_Samples.JSON.DeptJSON.txt"); streamWriter.Write(json); } var httpResponse = (HttpResponseMessage)httpWebRequest.GetResponse(); using (var streamReader = new StreamReader(httpResponse.GetResponseStream())) { var result = streamReader.ReadToEnd(); } </pre>
Employee Info Create/Update	<p><i>This is the key information which identifies a user with-in LMExcellence who are authorized to access & play various roles with in</i></p> <p>END POINT....</p> <p>SAMPLE code in C#</p>

3.2 DATA MODELS

This section provides detailed structure for all in and out parameters used to create/update various information as well as to fetch information from LMExcellence using above APIs.

3.2.1 Working with Company

Significance of a company in LMExcellence starts with every user being associated to a company as an employee. Company can have nested group companies under it, with access/visibility to information being limited to the company and its parent companies. E.g., a user belonging to root company can access information pertaining to users across its child companies or courses being associated to a department in its child companies.

Company comprises of following data points which is either passed or returned as part of company creation / updation –

Field	Type	Description
Code	String	Unique identifier of the company
Name	String	Name of the company
ParentCompanyCode	String	<i>In case of group company, this field is set to parent company</i>
Address	String	
City	String	
State	String	
Country	String	

3.2.2 Working with Departments

Courses in LMExcellence can be limited to users of a specific department, e.g. R&D, Quality, etc... By having courses associated with a department, one can ensure controlled access as well as distributed management of courses across departments.

Note: Access to department bound courses would be limited to employees who belong to the department.

Field	Type	Description
Code	String	Unique identifier for the department
Name	String	Name of the department, as departments with same name could exist across company and group-companies, this is not required to be unique
ParentDepartment	String	In case of group company, this field is set to parent company
Company	String	Company that department belongs to
...		

3.2.3 Working with employees / Users

Employee data is key to overall functioning of users, beyond authentication and logging into the system, having this data setup right would be crucial to ensuring that user can perform the operations their role requires them to, e.g. signing up for a course, nominating a reportee, etc.

Field	Type	Description
LoginID	String	
FirstName	String	First name of the employee
LastName	String	Last name of the employee
FullName	String	FullName of the employee
UserCompany	String	Code of the company
EmployeeId	String	Unique identifier for the department
eMailID	String	Email ID of the employee, is the unique identifier for the employee
Active	boolean	Employee active status. Value can be true/false
Department	String	Code of the department that employee belongs to
Designation	String	Designation of employee
Manager	String	LoginID of the Manager
CostCenterName	String	Name of the cost center
CostCenterCode	String	Code of the cost center
Grade	String	Grade of the employee
Address	String	Address of the employee
DateOfJoining	datetime	Date of joining

3.3 MANAGING LEARNING ACTIVITY –

Unlike APIs dealing with organization data, these are **reporting APIs**, which would return various information as managed in LMExcellence.

API	Endpoint	Description
Get Users Learning Profile	Get List of all Eligible (Mandatory only), Ongoing and Completed courses, as –	<ul style="list-style-type: none"> ▪ Eligible - Course Name, Category, Type, Department ▪ Ongoing – Course Name, Category, IsMandatory, Status, Started On, Department ▪ Completed – Course Name, Category, Certification Status, Score, Attempts, Completed On, Department
		END POINT....
		SAMPLE code in C# - Please refer section 3.3.1
Get Training Report	Learning profile of all users starting from a date (beginning of the year).	
		END POINT....
		SAMPLE code in C# - Please refer section 3.3.1

3.3.1 Selecting fetching data

All reporting API accepts filter parameters, which would allow users to fetch selective data, e.g. all certifications for the year, department specific certification, etc. Below is the listing of permissible filters as well as illustration of how to work with them –

API	Endpoint	Description
Get Users Learning Profile	<p>Applicable parameters are –</p> <ul style="list-style-type: none"> Employee ID, Date range, Status – Completed, Ongoing <p>Application of parameters as filters -</p> <ul style="list-style-type: none"> Ongoing – Any activity being carried out during the period specified in the date range for the given employee-id Completed – Courses completed during the specified period for the given employee-id <p>END POINT....</p> <pre>var url = "https://demo-staging.lmexcellence.com/apihandler.ashx" + "?fromdate=22-MAY-2020 12:00:00.000&todate=22-MAY-2020 23:59:59.000&employeeid=APP010&status=completed"; var httpWebRequest = (HttpWebRequest)WebRequest.Create(url); httpWebRequest.ContentType = "application/json"; httpWebRequest.Method = "GET"; httpWebRequest.Headers.Add("X-BIZAPP-TID", tid); httpWebRequest.Headers.Add("X-BIZAPP-APIKEY", key); httpWebRequest.Headers.Add("X-BIZAPP-APISECRET", secret); httpWebRequest.Headers.Add("X-BIZAPP-TYPEID", "EQMS7feb"); httpWebRequest.Headers.Add("X-BIZAPP-METHOD", "GetUsersLearningProfile"); var httpResponse = (HttpResponseMessage)httpWebRequest.GetResponse(); using (var streamReader = new StreamReader(httpResponse.GetResponseStream())) { var result = streamReader.ReadToEnd(); }</pre>	
Get Training Report	<p>Learning profile of all users for the given date range. Parameters passed would be fromdate, todate and Returnmandatorytrainingsonly (to specify whether to return all or only mandatory trainings)</p> <p>Notes – results are paginated, by default would return 100 records</p> <p>END POINT....</p> <pre>var url = "https://demo-staging.lmexcellence.com/apihandler.ashx" + "?fromdate=22-MAY-2020 12:00:00.000&ct=&todate=22-MAY-2020 23:59:59.000&returnmandatorytrainingsonly=true"; var httpWebRequest = (HttpWebRequest)WebRequest.Create(url); httpWebRequest.ContentType = "application/json"; httpWebRequest.Method = "GET"; httpWebRequest.Headers.Add("X-BIZAPP-TID", tid); httpWebRequest.Headers.Add("X-BIZAPP-APIKEY", key); httpWebRequest.Headers.Add("X-BIZAPP-APISECRET", secret); httpWebRequest.Headers.Add("X-BIZAPP-TYPEID", "EQMS7feb"); httpWebRequest.Headers.Add("X-BIZAPP-METHOD", "GetTrainingCompletionReport"); var httpResponse = (HttpResponseMessage)httpWebRequest.GetResponse(); using (var streamReader = new StreamReader(httpResponse.GetResponseStream())) { var result = streamReader.ReadToEnd(); }</pre>	

3.3.2 Working with larger sets of report data

This section explain how pagination is handled with reporting APIs and illustrates how the integrating tools could work with larger sets of data.

By default reports are paginated and return 100 records. With the first set of records, the resultset returns a continuation token (with key name 'ct'). This needs to be passed on to the further calls to get sub-sequent results.

Note: Sample project for the above APIs can be downloaded from <https://github.com/appoint-release/lme-docs/tree/master/Samples/Integration>