# CrowdWisdom™ Portal Gateway

#### Summary:

Portal gateway allows learners to seamlessly log in to the YM CrowdWisdom™ Learning Portal from trusted third-party applications.

Learners requiring access to the CrowdWisdom™ Learning Portal will be pre-authenticated by the trusted third-party application, and then they will be redirected to the single access point URL defined below.

### Details:

**Base URL:** http://<context>.mycrowdwisdom.com/diweb/gateway/

|  |  |
| --- | --- |
| Parameter Name | Parameter Value |
| data | A Triple DES encrypted, base64 encoded string resulting from the encryption of the learner data |

* All parameters are required to access the Learning Portal via gateway.

#### *Encrypted Data (data parameter)*

Learner data must be prepared in the form of query string before encryption. For example:

*Timestap=1123029012376&ClientID=323033&*

Learner data must be prepared in the form of query string before encryption. For example:

Field names are not case-sensitive, and values must be URL encoded. Then the prepared query string must be encrypted using Triple DES encryption in CBC mode with PKCS5 padding (“DESede/CBC/PKCS5Padding”). – The secret encryption key and initialization vector to be used in the encryption algorithm will be provided by YMLearning in the form of base64 encoded strings – The data in the encrypted query string is used to automatically **register *new*** learners and **authenticate *returning*** learners. The following data fields are defined:

## 

|  |  |
| --- | --- |
| Field | Description |
| Timestamp | The date and time at which the encrypted query string was created. It is used to prevent process stale or expired data.  This field consists of a sequence of digits representing the number of milliseconds in GMT since midnight, January 01, 1970  The encrypted data will not be considered valid if the timestamp is older than the max age defined for the current portal. resulting from the encryption of the learner data |
| ClientID | unique client identifier (i.e. person ID in your system) |
| FirstName | First name of the learner |
| LastName | Last name or family name of the learner |
| Email | Learner email address |
| DisplayName | Learner Salutation if different from FullName(FirstName LastName ). For example (**Dr** FirstName LastName) can be used on Certificates |
| UserName | Learner username could be used to log into the learning portal manually. Useful only if the manual login is allowed for the current portal. |
| Password | Learner password could be used to log into the learning portal manually. Useful only if the manual login is allowed for the current portal. |
| LastUpdate | The date and time the learner demographic data was last updated. Same format as **timestamp**. |
| Address | Learner Address |
| City | City |
| State | State or Province |
| Zip | Postal or zip code |
| Country | Country code, ISO 3166 country code (e.g. US) |
| Phone | Contact phone number |
| Group | Membership group name(s), comma separated list of groups that the learner belongs to. (i.e. member, staff) |

Fields in bold are required, optional fields could be omitted from the query string. All other fields (including the ones not listed here) will be saved as learner demographic data.

Note: you may omit the learner demographic data (from FirstName to Phone) if you are certain that the learner is already registered with the portal and the profile hasn’t changed since registration.

For comments and questions, please contact your YMLearning Technical point-person.

#### Sample Client (pseudo code)

*//prepare parameters to be passed to the gateway*

Map<String,String> params = new HashMap<String, String>();

params.put("clientId", "123);

...

..

.

*//construct key/value pair query string*

StringBuilder sb = new StringBuilder();

for (Map.Entry<String,String> param : params.values()) {

if (sb.length() > 0) sb.append('&');

sb.append(param.getKey());.append('=');

*//url encode the value as this is query string*

sb.append(jURLEncoder.encode(param.getValue()));

}

*//create a cipher (TripleDES algorithm, CBC mode with PKCS5Padding)*

Cipher cipher = Cipher.getInstance("DESede/CBC/PKCS5Padding");

*//decode Digital Ignite provided KEY and IV*

byte[] key = Base64.decode(KEY); *//KEY is Base64 encoded*

byte[] iv = Base64.decode(IV); *// IV is Base64 encoded*

SecretKey secretkey= …; *// create secret key using key byte array*

IvParameterSpec ivSpec=…; *//create initialization vector using the iv byte array*

*//initialize the cipher for encryption*

cipher.init(Cipher.ENCRYPT\_MODE, secretKey, ivSpec);

*//use base64 encoding for resulting bytes*

String dataStream = Base64.encode(cipher.doFinal(sb.toString().getBytes()));

*//redirect to gateway (url encode the data stream as it is being attached to the url)* getRespone().redirect(gatewayURL + "?data=" + URLEncoder.encode(dataStream));