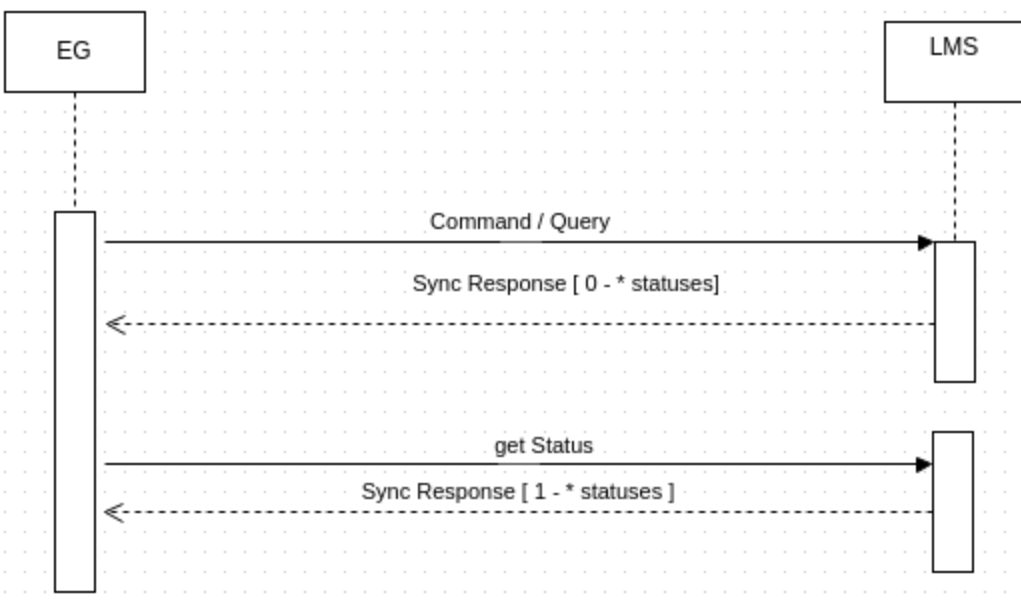


## Employee Guard - LMS Integration

### EG-LMS API

Key points:

- 1) The interaction between EG and LMS is performed by HTTP Post requests.
- 2) The data sent is a JSON formatted message.
- 3) Each JSON message will contain a message id and one or more payloads.
- 4) Multiple payloads is for batching purposes.
- 5) The payload is the message to be processed. The payload will contain a payload id to uniquely identify this payload.
- 6) The interaction will contain both sync and async inbound / outbound messaging.
- 7) For the initial release all messaging will be initiated from the EG side.
- 8) A call into the LMS will perform either a command, a query or a status request. A command executes and action like adding a user. A query returns specific data regarding users, orgs, assignments, etc. An status request polls for statuses for previous command or query requests.
- 9) The synchronous response from the LMS will contain a status code indicating the message was received, this code doesn't mean the message was processed already, just received.
- 10) The synchronous response from the LMS could also contain a payload containing one or multiple statuses to previous commands and queries.
- 11) EG polls the LMS to retrieve the statuses from the previous command / query messages.



## JSON Message

The JSON message contains an id to uniquely identify it. It also contains one or more payloads. The payload also has an id to uniquely identify it. The ids are used to track messages and payloads across the two systems.

Message	Response correlating refId with id for message and payload
<pre>{   "id": 5049,   "payload": [     {       "id": 4337,       "command": "addUser",       "data": {         "userId": 8500,         "username": "tester",         "type": "employee",         "accountIds": "1234,5678",         "firstName": "The",         "lastName": "Tester",         "locale": "en_US",         "email": "test@tester.com"       }     },     {       "id": 4400,       "command": "assignUserToCourse",       "data": {         "userId": 601,         "courseId": 9132       }     }   ] }</pre>	<pre>{   "refId": 5049,   "statusCode": 0,   "httpStatusCode": 200,   "status": [     {       "refId": 4337,       "statusCode": 0,       "data": null     },     {       "refId": 4400,       "statusCode": 0,       "data": null     }   ] }</pre>

## Commands

### Users (aka employee)

Command	Parameters	Desc
addUser	userId, userType, accountId, firstName, lastName, locale, email	Adds a user (employee). Parameter accountId specifies the org's account id they have a relationship with
updateUser	userId, userType, accountId, firstName, lastName, locale, email	updates an user
deactivateUser	userId	Marks the user as not active
deleteUser	userId	deletes the user
getUsersByIds	ids (comma separate list)	Provided one or more ids, return user info
getAllUsersIds		Returns all user ids

### Assignments

- getAssignmentByCourseId
- getAssignmentByUserId
- getAssignmentByUserIdCourseId
- getAssignmentById
- getAllAssignmentsIds

todo: Describe each command in detail.

Here is the complete list of command with their payload data

Complete messages example

```
{
  "id": 2798,
  "payload": [
    {
      "id": 4337,
```

```
"command": "addUser",
"data": {
  "userId": 8500,
  "username": "tester",
  "type": "employee",
  "accountIds": "1234,5678",
  "firstName": "The",
  "lastName": "Tester",
  "locale": "en_US",
  "email": "test@tester.com"
}
},
{
  "id": 5007,
  "command": "updateUser",
  "data": {
    "userId": 9227,
    "username": "tester",
    "type": "employee",
    "accountIds": "1234,5678",
    "firstName": "The",
    "lastName": "Tester",
    "locale": "en_US",
    "email": "test@tester.com"
  }
},
{
  "id": 4523,
  "command": "deactivateUser",
  "data": {"userId": 7538}
},
{
  "id": 5019,
  "command": "deleteUser",
  "data": {"userId": 6643}
},
{
  "id": 5046,
  "command": "getUsersByIds",
```

```
"data": "[ 123, 456, 789 ]"
},
{
  "id": 1685,
  "command": "getAllUserIds"
},
{
  "id": 8006,
  "command": "addPicsAccount",
  "data": {
    "name": "PICS ACCOUNT",
    "contactFirstName": "Account",
    "contactLastName": "Contact",
    "contactLocale": "de_DE",
    "contactEmail": "account.email@account.com"
  }
},
{
  "id": 4358,
  "command": "updatePicsAccount",
  "data": {
    "name": "PICS ACCOUNT",
    "contactFirstName": "Account",
    "contactLastName": "Contact",
    "contactLocale": "de_DE",
    "contactEmail": "account.email@account.com"
  }
},
{
  "id": 2726,
  "command": "deactivatePicsAccount",
  "data": {}
},
{
  "id": 1147,
  "command": "deletePicsAccount",
  "data": {"userId": 8274}
},
{
```

```
"id": 9786,
"command": "findPicsAccountByIds",
"data": "[ 123, 456, 789 ]"
},
{
  "id": 6362,
  "command": "getAllPicsAccountIds"
},
{
  "id": 1734,
  "command": "getAssignmentByCourseId",
  "data": "[ 123, 456, 789 ]"
},
{
  "id": 2934,
  "command": "getAssignmentByUserId",
  "data": "[ 123, 456, 789 ]"
},
{
  "id": 559,
  "command": "getAssignmentByUserIdCourseId",
  "data": "[ 3787, 8168 ]"
},
{
  "id": 8984,
  "command": "getAssignmentById",
  "data": "[ 7114 ]"
},
{
  "id": 2913,
  "command": "getAllAssignmentsIds"
},
{
  "id": 4400,
  "command": "assignUserToCourse",
  "data": {
    "userId": 601,
    "courseId": 9132
  }
}
```

}  
]  
}