## Statement Example:

Statements are basically the main format for request bodies and responses and follow all the standards in this file.

Each statement have 4 main objects:

1. Actor, which is the user who created the statement.
2. Verb, What action did the user do
3. Object, the item that was created or affected by this statement and all metadata
4. Context, which hosts all miscellaneous data  
     
   https://www.pythonanywhere.com/

| {  "id":"fd41c918-b88b-4b20-a0a5-a4c32391aaa0",  "timestamp": "2015-11-18T12:17:00+00:00",  "actor":{  "name": "John doe",  "id": "5942f46c-50cf-4bbe-b447-fadd1be1c840"  },  "verb":{  "id-enum":"verbs-took-exam",  "display":{   "en-US":"took an exam"   }  },  "object":{  "id":"5942f46c-50cf-4bbe-b447-fadd1be1c840",  "objectType": "Activity",  "definition":{  "name":{  "en-US":"simple CBT course"  },  "description":{  "en-US":"A fictitious example CBT course."  }  }  },  "Context":{  "result":{  "score":{  "scaled":"0.95"  },  "success":"true",  "completion":"true",  "duration": "PT12M34.15S"  }  } } |
| --- |

## 

## System IDs:

All system IDs should be [RFC](https://www.ietf.org/rfc/rfc4122.txt)-compliant so as to be a UUID

Example: '60222bd9-85da-497b-bd58-7a2c68981c7b'

Resources to help:

<https://github.com/uuidjs/uuid>

<https://stackoverflow.com/questions/105034/how-do-i-create-a-guid-uuid>







To create a UUID using JavaScript, use the first code if you are in a modern browser and the second in other environments but the first has more security.

You can run those codes in the inspect feature to see the output for yourself.

| function uuidv4() {  return "10000000-1000-4000-8000-100000000000".replace(/[018]/g, c =>  (c ^ crypto.getRandomValues(new Uint8Array(1))[0] & 15 >> c / 4).toString(16)  ); } |
| --- |

Or

| function generateUUID() { // Public Domain/MIT  var d = new Date().getTime();//Timestamp  var d2 = ((typeof performance !== 'undefined') && performance.now && (performance.now()\*1000)) || 0;//Time in microseconds since page-load or 0 if unsupported  return 'xxxxxxxx-xxxx-4xxx-yxxx-xxxxxxxxxxxx'.replace(/[xy]/g, function(c) {  var r = Math.random() \* 16;//random number between 0 and 16  if(d > 0){//Use timestamp until depleted  r = (d + r)%16 | 0;  d = Math.floor(d/16);  } else {//Use microseconds since page-load if supported  r = (d2 + r)%16 | 0;  d2 = Math.floor(d2/16);  }  return (c === 'x' ? r : (r & 0x3 | 0x8)).toString(16);  }); } |
| --- |

## TimeStamps and durations:

All system timestamps should follow iso 8601 standards

Example: ‘2013-04-01T12:00:00Z’

Resources:

<https://github.com/adlnet/xAPI-Spec/blob/master/xAPI-Data.md#45-iso-8601-timestamps>

JavaScript example code below

| const event = new Date('05 October 2011 14:48 UTC'); console.log(event.toString()); // Expected output: "Wed Oct 05 2011 16:48:00 GMT+0200 (CEST)" // Note: your timezone may vary  console.log(event.toISOString()); // Expected output: "2011-10-05T14:48:00.000Z"  //Gets the current time in the user system let timestamp = new Date().toISOString(); console.log(timestamp); |
| --- |

## Durations of any event:

All durations should follow the ISO standard and they will be mostly used for Quizzes and assignments

| **Example** | **Explanation** |
| --- | --- |
| PT4H35M59.14S | Four hours, thirty five minutes and 59.14 seconds. |
| PT16559.14S | The same time period as above represented in seconds. (Note: if the time period in question contained a leap second, this conversion would be inaccurate) |
| P3Y1M29DT4H35M59.14S | A Duration also including years, months and days. |
| P3Y | Approximately three years e.g. completion of a qualification. |
| P4W | Four weeks. Note that weeks cannot be combined with other time periods. 'P4W1D' is not valid. |

Resources:

<https://www.digi.com/resources/documentation/digidocs//90001488-13/reference/r_iso_8601_duration_format.htm>

Javascript example code below

| let timestamp1 = new Date("2023-10-04T13:25:48.284Z"); console.log(timestamp1);  let timestamp2 = new Date("2024-12-12T16:28:57.756Z"); console.log(timestamp2);  let timeDiff = new Date(timestamp2).getTime() - new Date(timestamp1).getTime(); console.log(timeDiff) console.log(turnIntoISODuration(timeDiff));  function turnIntoISODuration(duration){  let durString = "P"  let temp;   temp = (timeDiff - (timeDiff % 31622400000)) / 31622400000;//Years  timeDiff = timeDiff % 31622400000  if(temp!=0){durString += temp + "Y";}   temp = (timeDiff - (timeDiff % 2592000000)) / 2592000000; //Months  timeDiff = timeDiff % 2592000000;  if(temp!=0){durString += temp + "M";}   temp = (timeDiff - (timeDiff % 604800000)) / 604800000; //Weeks  timeDiff = timeDiff % 604800000;  if(temp!=0){durString += temp + "W";}   temp = (timeDiff - (timeDiff % 86400000)) / 86400000; //Days  timeDiff = timeDiff % 86400000;  if(temp!=0){durString += temp + "D";}     if(timeDiff!=0){durString+="T"}    temp = (timeDiff - (timeDiff % 3600000)) / 3600000; //Hours  timeDiff = timeDiff % 3600000;  if(temp!=0){durString += temp + "H";}   temp = (timeDiff - (timeDiff % 60000)) / 60000; //Minutes  timeDiff = timeDiff % 60000;  if(temp!=0){durString += temp + "M";}   temp = timeDiff / 1000; //Seconds  if(temp!=0){durString += temp + "S";}   return durString } |
| --- |

## Verbs id-enum:

Color coding for validation: user, admin, privileged user, server

| Verb Enum/Action | Description |
| --- | --- |
| sent-partial-course-data-successfully | server |
| request-denied | server refuses a request and sends more info in the context object. |
| get-privileges-list |  |
| sent-system-privileges-list | server |
| create-user | Used when creating a new account by a guest, not an admin. |
| user-created-successfully | server creates a new user and sends their info |
| update-user |  |
| user-updated-successfully | server |
| user-deleted-successfully | server |
| log-into-account | The user attempts to log in |
| admin-create-course | admin |
| course-created-successfully | server |
| admin-update-course | admin |
| course-updated-successfully | server |
| admin-delete-course | admin |
| course-deleted-successfully | server |
| admin-create-user | admin |
| admin-update-user | admin |
| admin-delete-user | admin |
| admin-request-course-code | admin |
| enroll-in-course |  |
| user-enrolled-successfully | server |
| user-get-courses |  |
| sent-courses-data-successfully | server |
| user-get-users |  |
| sent-users-data-successfully | server |
| user-get-posts |  |
| sent-posts-data-successfully | server |
| user-get-comments |  |
| sent-comments-data-successfully | server |
| user-get-materials |  |
| sent-materials-data-successfully | server |
| privileged-user-posts-in-course | privileged user |
| post-created-successfully | server |
| privileged-user-update-post | privileged user |
| post-updated-successfully | server |
| privileged-user-delete-post | privileged user |
| post-deleted-successfully | server |
| privileged-user-comments-on-post | privileged user |
| comment-created-successfully | sever |
| privileged-user-updates-comment | privileged user |
| comment-updated-successfully | server |
| privileged-user-deleted-comment | privileged user |
| comment-deleted-successfully | server |
| get-roles-list-in-course |  |
| get-privileges-list-for-user |  |
| privileged-user-create-quiz-in-course | privileged user |
| quiz-created-successfully | server |
| access-quiz-in-course |  |
| submit-quiz-in-course |  |
| quiz-submitted-successfully | server |
| privileged-user-creates-role-in-course | privileged user |
| role-created-successfully | server |
| privileged-user-updates-role-in-course | privileged user |
| role-updates-successfully | server |
| privileged-user-delete-role-in-course | privileged user |
| role-deleted-successfully | server |
| privileged-user-assigns-role-in-course | privileged user |
| role-assigned-successfully | server |

## 

## Objects Enum:

| Object Type | Meaning |
| --- | --- |
| Agent | A user |
| Activity | A Course/Post/Comment/Etc |
| Group | A group of objects |

| Material Type | Meaning |
| --- | --- |
| 1 | Multimedia text file (PDF,Docx,etc) |
| 2 | audio |
| 3 | Powerpoint |

Note: if a user tries to edit or delete an object and that user is the object creator, then he has full privileges.

Note: roles go from 0 upwards with 0 being the whole system admin and 1 being a course teacher.

| Privilege | code | Description |
| --- | --- | --- |
| create normal post | 1 |  |
| Edit lower-level posts. | 2 | Edit posts of use with the same and lower-level roles |
| Delete lower-level posts. | 3 | Delete posts of users with the same and lower-level roles |
| Create poll | 4 |  |
| Create quiz | 5 |  |
| Comment on post | 6 |  |
| Delete lower-level comments | 7 | Delete comments of users with the same and lower-level roles |
| Assign roles to lower-level users | 8 | Assign equal or lower level roles to users with lower level roles |

## Classes Object Example:

User:

| "user":{  "name":"Ahmed Hamad",  "id": "fb6fe636-d462-4ccf-ad3e-b8a132a254f8",  "assignedId":"20200458",  "email":"ahmed459@gmail.com",  "avatar": "LINK",  "courses": ["4970e877-04f6-47d2-83aa-b43003beb9a1","3f2c92a1-5e60-4305-83b2-ed9159d3deb1"]  } |
| --- |

### Course:

| "course":{  "name":"intro to computers",  "code":"IS256",  "id": "fb6fe636-d462-4ccf-ad3e-b8a132a254f8",  "maxCapacity":"250",   "teachers": ["3fc12698-ff06-41bd-b09d-d4ca01a49c4d","cb4cb09f-95c8-41c5-aa33-329c35621825"],  "Posts":[]  } |
| --- |

### Post:

| "post":{  "id": "e792e71f-0660-4a63-809f-4bf4a26b4276",  "courseId":"3303bb26-da01-4fb4-b8b1-9c60ec718507",  "postedBy": "ec1d1512-5712-4a22-99d8-161dc829fbd2",  "postedOn":"2015-11-18T12:17:00+00:00",  “editedOn”: "2015-11-18T12:17:00+00:00",  "content":"This is a very normal post",  “comments”:[],  “materials":[]  } |
| --- |

### 

### Comment:

| "comment":{  "id": "e792e71f-0660-4a63-809f-4bf4a26b4276",  "postedBy": "ec1d1512-5712-4a22-99d8-161dc829fbd2",  "postedOn":"2015-11-18T12:17:00+00:00",  “editedOn”: "2015-11-18T12:17:00+00:00",  "replyingTo": "1df6a686-3be7-4fdf-9938-0e38b98f8400",  "content":"This is a comment"  } |
| --- |

### Role:

| "role":{  "courseId": "ec1d1512-5712-4a22-99d8-161dc829fbd2",  "roleCode": "269984",  "role": "T.A",  "level": "2",   "privileges": ["2","3"]  } |
| --- |

### Privilege:

| "privilege":{  "privilegeCode":"3",  "Description":  "Delete posts of users with the same and lower-level roles"  } |
| --- |

### Material:

| "material":{  "id": "e792e71f-0660-4a63-809f-4bf4a26b4276",  "courseId":"ec1d1512-5712-4a22-99d8-161dc829fbd2",  "description":"This is a simple PDF",  "type": "5",  "URI":"LINK"  } |
| --- |

### Quiz:

| "quiz":{  "id": "e792e71f-0660-4a63-809f-4bf4a26b4276",  "courseId":"ec1d1512-5712-4a22-99d8-161dc829fbd2",  "questions": ["9a3bf4e3-6483-4eaf-a2bc-9aa204147073","48df9c58-85a8-43b3-8640-7aa2388a6b75"],  "retries": "5",  "duration": "PT1H",  "closeOn": "2015-11-18T12:17:00+00:00",  "showGradeAfter": "true",  "showWrongAnswersAfter": "true",  "showCorrectAnswersAfter":"false"  } |
| --- |

### Quiz question:

| "quizQuestion":{  "id": "e792e71f-0660-4a63-809f-4bf4a26b4276",  "question": " What methodology is iterative? ",  "options": ["waterfall", "Agile", "W3DT"],  "answer": "Agile",  "weight": "4"  } |
| --- |