# API design v0.1

1. Create user

|  |  |
| --- | --- |
| Description | create a user, then add the user to sessions, mark him as signed in. |
| URL | /users |
| Request Method | POST |
| Request Header | Empty, maybe add some security token in next sprint |
| Request Body | {osu\_id, last\_name, first\_name, role} |
| Return Value | User info {id, osu\_id, last\_name, first\_name, role, session\_id} |
| Return Code | Success: 201, Failure: 403 |

1. Sign in

|  |  |
| --- | --- |
| Description | User sign in, add user to sessions |
| URL | /sessions |
| Request Method | POST |
| Request Header | Empty, maybe add some security token in next sprint |
| Request Body | {osu\_id } |
| Return Value | User info {id, osu\_id, last\_name, first\_name, role, session\_id} |
| Return Code | Success: 201, Failure: 403 |

1. Sign out

|  |  |
| --- | --- |
| Description | User sign out, remove user from sessions |
| URL | /sessions |
| Request Method | DELETE |
| Request Header | user session |
| Request Body |  |
| Return Value |  |
| Return Code | Success: 204, Failure: 403 |

1. (TA) create class

|  |  |
| --- | --- |
| Description | TA create course. TA role need to validate first. After an item is added in ‘t\_class’ table, add a new item in ‘r\_user\_class’ table. |
| URL | /classes |
| Request Method | POST |
| Request Header | user session |
| Request Body | {name} |
| Return Value | Class info {id, name} |
| Return Code | Success: 201, Failure: 403 |

1. User (TA or Students) query classes

|  |  |
| --- | --- |
| Description | TA or students query classes. If ‘category==all’, return all classes in our system. If ‘category == self’, return the classes that they are in. Pagination will be solved in next sprint. |
| URL | /classes?category={all,self} |
| Request Method | GET |
| Request Header | user session |
| Request Body |  |
| Return Value | Class info array [{id, name}, {id, name} …] |
| Return Code | Success: 200, Failure: 404 |

1. User (TA or Students) choose classes

|  |  |
| --- | --- |
| Description | TA or students choose a class. Add an item in ‘r\_user\_class’ table. Maybe can choose many classes once in next sprint. |
| URL | /classes |
| Request Method | PUT |
| Request Header | user session |
| Request Body | {class\_id} |
| Return Value |  |
| Return Code | Success: 200, Failure: 404 |

1. User (TA or Students) quit classes

|  |  |
| --- | --- |
| Description | TA or students quit a class. Maybe can quit many classes once in next sprint. |
| URL | /classes |
| Request Method | DELETE |
| Request Header | user session |
| Request Body | {class\_id} |
| Return Value |  |
| Return Code | Success: 204, Failure: 404 |

1. Student post question

|  |  |
| --- | --- |
| Description | Student post a question |
| URL | /questions |
| Request Method | POST |
| Request Header | user session |
| Request Body | {class\_id, title, description, course\_keywords, preferred\_time} |
| Return Value | Question Info {all attributes in ‘t\_question’ table} |
| Return Code | Success: 201, Failure: 403 |

1. User (TA or Student) query question

|  |  |
| --- | --- |
| Description | TA or students query classes. Pagination will be solved in next sprint. |
| URL | /questions?class\_id={class\_id} |
| Request Method | GET |
| Request Header | user session |
| Request Body |  |
| Return Value | Question Info Array [{all attributes in ‘t\_question’ table}, {}, …] |
| Return Code | Success: 200, Failure: 404 |

1. Student add himself to question

|  |  |
| --- | --- |
| Description | Student add himself to question, create a new item in ‘t\_question\_consern’ table |
| URL | /questions/{question\_id}/conserns |
| Request Method | PUT |
| Request Header | user session |
| Request Body |  |
| Return Value | Question Info {all attributes in ‘t\_question’ table} |
| Return Code | Success: 201, Failure: 404 |

1. TA set the status of a question

|  |  |
| --- | --- |
| Description | TA set the status of a question (status 0: proposed, status 1: answered, status 2: deleted, status 3: signed) |
| URL | /questions |
| Request Method | PETCH |
| Request Header | user session |
| Request Body | {question\_id, statues} |
| Return Value | Question Info {all attributes in ‘t\_question’ table} |
| Return Code | Success: 200, Failure: 404 |