GET /projects

Returns the list of projects associated with a user.

```
arguments:
  username: (String) The username of the user you would like to retrieve projects for

returns: {'success': boolean, 'projects': [Object], ‘message’: String}
  success: (boolean) true if a possibly empty set of projects was returned, else false
  projects: (array) Array of projects associated with user. Is undefined when success is
  false. Each project contains:
  {
    leader: (string) the user in charge of the project
    name: (string) name of the project
    description (string) description of the project
    users: (array) all the users in the project
    tasks: (array) all the tasks associated with the project
  }
  message: (String) error message. Is undefined when success is true

returns: 401 Not Authorized if no user is logged in
```

POST /projects

Create a project with a given name, leader, description, and users.

```
arguments:
  name: (String) name of the project
  leader: (String) The person in charge of the project
  description: (String) description of the project
  users: (array) all the users in the project

returns: {‘success’: boolean, ‘id’: String, ‘message’: String}
  success: (boolean) true if a project was created, else false
  id: (String) the id of the newly created project. Is undefined when success is false
  message: (String) error message. Is undefined when success is true

returns: 401 Not Authorized if no user is logged in
```

GET /projects/{id}

Returns the project specified by the id. User must be a member of the project.

```
arguments: None
```

```
returns: {'success': boolean, ‘project’: Object, ‘message’: String}
  success: (boolean) true if the project exists and the user is a member, else false
  project: (Object) a project object. Is undefined when success is false. Contains the
  following fields:
  {
    leader: (string) the user in charge of the project
    name: (string) name of the project
    description (string) description of the project
    users: (array) all the users in the project
    tasks: (array) all the tasks associated with the project
  }
  message: (String) error message. Is undefined when success is true

returns: 401 Not Authorized if no user is logged in
```

POST /projects/{id}

Edit the project’s leader, name, description, and the set of users assigned to the project. User must be the project leader.

```
arguments:
  leader: (string) the new username in charge of project
  name: (string) new name of project
  description: (string) description of the updated project
  users: (array) all the users in the updated project

returns: {'success': boolean, ‘message’: String}
  success: (boolean) true if the update was successful, false otherwise
  message: (String) error message. Is undefined when success is true

returns: 401 Not Authorized if no user is logged in
```

DELETE /projects/{id}

Delete the project. User must be the project leader.

```
arguments: None

returns: {‘success’: boolean, ‘message’: String}
  success: (boolean) true if the deletion was successful, else false
  message: (String) error message. Is undefined when success is true

returns: 401 Not Authorized if no user is logged in
```

GET /projects/{id}/tasks

Returns the set of tasks assigned to the user for the project with the given id.

```
arguments:
  username: (String) username of the user logged in

returns: {‘success’: Boolean, ‘tasks’: [Object], ‘message’: String}
  success: (Boolean) true if the set of tasks is successfully retrieved, else false
  tasks: (array) the tasks assigned to the user for the project. Is undefined when
  success is false. Each task contains the following fields:
    assignee: (String) username of the user assigned to the task
    description: (String) a description of the task
    completion: (number) degree of completion of the task
    deadline: (date) the date at which the task is due
  message: (String) error message. Is undefined when success is true

returns: 401 Not Authorized if no user is logged in
```

POST /projects/{id}/tasks

Adds a new task to the project with the given id. User must be the project leader.

```
arguments:
  assignee: (String) username of the user assigned to the task
  description: (String) a description of the task
  completion: (number) degree of completion of the task
  deadline: (date) the date at which the task is due

returns: {‘success’: Boolean, ‘message’: String}
  success: (Boolean) true if task is successfully added, else false
  message: (String) error message. Is undefined when success is true

returns: 401 Not Authorized if no user is logged in
```

GET /projects/{id}/tasks/{id}

Returns the task specified by the identifiers. User must be a member of the project.

```
arguments: None

returns: {‘success’: Boolean, ‘task’: Object, ‘message’: String}
  success: (boolean) true if task is returned, else false
  task: (Object) a task object. Is undefined when success is false. Contains the
  following fields:
    assignee: (String) username of the user assigned to the task
    description: (String) a description of the task
    completion: (number) degree of completion of the task
    deadline: (date) the date at which the task is due
  message: (String) error message. Is undefined when success is true

returns: 401 Not Authorized if no user is logged in
```

PUT /projects/{id}/tasks/{id}

Change the assignee, description, degree of completion, and deadline of the task at id. User must be either the project leader or the assignee to update the task.

```
arguments:
  assignee: (String) username of the user assigned to the task
  description: (String) a description of the task
  completion: (number) degree of completion of the task
  deadline: (date) the date at which the task is due

returns: {‘success’: Boolean, ‘message’: String}
  success: true if task is successfully updated with the field values, else false
  message: (String) error message. Is undefined when success is true

returns: 401 Not Authorized if no user is logged in
```

DELETE /projects/{id}/tasks/{id}

Delete task specified by the project id and task id. User must be the project leader.

```
arguments: None

returns: {‘success’: Boolean, ‘message’: String}
  success: true if task is successfully deleted, else false
  message: (String) error message. Is undefined when success is true

returns: 401 Not Authorized if no user is logged in
```

GET /users

Returns all the users.

```
arguments: none
```

```
returns: [Object] array of users; each user contains the following fields:
  username: (String) username of the user
  department: (String) department of the user

returns: 401 Not Authorized if no user is logged in
```

POST /users

Creates a new user.

```
arguments:
  username: (String) username of the user
  password: (String) password of the user
  department: (String) department of the user

returns: {‘success’: Boolean, ‘message’: String}
  success: true if user was successfully created, else false
  message: (String) error message. Is undefined when success is true
```

POST /sessions

Attempts to start a new session with the given username and password.

```
arguments:
    username: (String) username of the user
    password: (String) password of the user

returns: Unauthorized if the login was unsuccessful, or {success: true} otherwise
```