

Create User @ <https://final-rashj.uc.r.appspot.com>

API @ <https://final-rashj.uc.r.appspot.com>

Final Project: API Spec

CS 493: Cloud Application Development Spring 2022

Oregon State University

Jason Rash

Last Update: June 1 2:25 PM Central

Change log

Version	Change	Date
1.0	Initial version.	June 1, 2022

Change log and Notes	1
Data Model	2
GET all Users	3
Get all Projects for User	4
Get a Project	6
Post a Project	7
PUT a Project	9
PATCH a Project	11
Delete a Project	13
Get all Issues	14
Get an Issue	15
Post an Issue	16
PUT an Issue	18
Patch an Issue	20
Delete an Issue	22

NOTES

- Each project has a user associated with it (one-to-one relationship). The specified user is the only user that has put, patch, and delete access to that project.
- When a User is signed up, a JWT is associated with a user. The User is given a random user_id and the token field is set to the sub field of the JWT.
- For a POST request, Authentication is done by first verifying the JWT token, and getting the sub field. Then the sub field is used to search for a user that the token field matches the sub field of the JWT. If the user exists, then the project is saved in the database along with the user_id of the user.
- For PUT/PATCH and DELETE requests, Authentication is done by verifying the JWT token and getting the sub field. Then if the project_id exists, the user_id field of the project is used and the token field of the user is compared to the sub field of the JWT. If they match then the PUT/PATCH/DELETE request is resolved. If they do not match then a 401 error 'Access Denied' JWT sub does not match user token for project' is sent.
- For POST requests the JWT must be supplied.
- Users can have many projects that they are responsible for (one-to-many relationship).
- There can be many issues associated with one project (many-to-one relationship).
- Issues are not protected. Anyone can post, patch, put, or delete them.

Data Model

The app stores three kinds of entities in Datastore. (Users, Projects and Issues)

NOTE: Users are not to be created manually. User is created after user login via OAuth.

Users

Property	Data Type	Notes
id	Integer	The id of the user. Datastore automatically generates it. Don't add it yourself as a property of the entity.
firstName	String	First name of the user.
lastName	String	Last name of the user.
token	String	The sub property of JSON Web Token from OAuth 2.0
projects	Array	Array of projects assigned to the user.
self	String	The url that points to the user.

Projects

Property	Data Type	Notes
id	Integer	The id of the project. Datastore automatically generates it. Don't add it yourself as a property of the entity.
name	String	Name of project.
description	String	Description of project.
date	String	Start date of project.
issues	Array	Array of issues assigned to the project.
user_id	String	The id of the user who created the project.
self	String	The url that points to the project.

Issues

Property	Data Type	Notes
id	Integer	The id of the issue. Datastore automatically generates it. Don't add it yourself as a property of the entity.
name	String	Name of the issue.
description	String	Description of the issue.
priority	String	Priority of the issue.
project_id	String	The id of the project associated with the issue.
self	String	The url that points to the issue.

GET all Users

Allows you to get all Users.

GET /users

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	

Response Examples

Success

```
Status: 200 OK
{
  "users": [
    {
      "lastName": "Rash",
      "token": "115195949759170028933",
      "projects": [
        {
          "id": "5754702511734784",
          "self": "http://127.0.0.1:8080/projects/5754702511734784"
        },
        {
          "id": "5717649963089920",
          "self": "http://127.0.0.1:8080/projects/5717649963089920"
        }
      ],
      "firstName": "Jason",
      "id": "5651294463197184",
      "self": "http://127.0.0.1:8080/users/5651294463197184"
    }
  ]
}
```

Get all Projects for User

Allows you to get all projects for a user if the JWT associated with the user of the project is included in headers authorization.

GET /projects/

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	JWT is missing or JWT does not match token for user.
Failure	404 Not Found	No project with this project_id exists
Failure	406 Not Acceptable	Accept header must be set to application/json

Response Examples

Success

```
Status: 200 OK
{
  "projects": [{
    "name": "Second Project",
    "description": "This is the second project for user1",
    "date": "5/31/22",
    "issues": [],
    "user_id": "5651294463197184",
    "id": "5722078678351872",
    "self": "http://127.0.0.1:8080/projects/5722078678351872"
  },
  {
    "date": "5/25/22",
    "description": "Project and Issue Tracking API",
    "name": "Issue API",
    "user_id": "5651294463197184",
    "issues": [{
      "id": "5734484288733184",
      "self": "http://127.0.0.1:8080/issues/5734484288733184"
    }],
    "id": "5756751345352704",
    "self": "http://127.0.0.1:8080/projects/5756751345352704"
  }
]
```

Failure

```
Status: 401 Unauthorized
{
  "Error": "Access Denied. JWT sub does not match user token for project."
}
```

Status: 404 Not Found

```
{  
  "Error": "No project with this project_id exists"  
}
```

Status: 406 Not Acceptable

```
{  
  "Error": "Accepts type is Not Acceptable. Must be application/json."  
}
```

Get a Project

Allows you to get an existing project if the JWT associated with the user of the project is included in headers authorization.

GET /projects/:project_id

Request

Path Parameters

Name	Description
project_id	ID of the project

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	JWT is missing or JWT does not match token for user.
Failure	404 Not Found	No project with this project_id exists
Failure	406 Not Acceptable	Accept header must be set to application/json

Response Examples

Success

<pre>Status: 200 OK { "name": "Issue API", "date": "5/25/22", "description": "Project and Issue Tracking API", "user_id": "5685839489138688", "issues": [], "self": "http://127.0.0.1:8080/projects/5700985489981440", "id": "5700985489981440" }</pre>

Failure

<pre>Status: 401 Unauthorized { "Error": "Access Denied. JWT sub does not match user token for project." }</pre>
<pre>Status: 404 Not Found { "Error": "No project with this project_id exists" }</pre>
<pre>Status: 406 Not Acceptable { "Error": "Accepts type is Not Acceptable. Must be application/json." }</pre>

Post a Project

Allows you to post a project if the JWT associated with the user of the project is included in headers authorization.

POST /projects/

Request

Path Parameters

Name	Description
project_id	ID of the project

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
name	Name of the project. (Project Name must be unique)	Yes
description	Description of the project.	Yes
date	Date project was started	Yes

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	Missing at least one required attribute in request body.
Failure	401 Unauthorized	JWT is missing or JWT does not match token for user.
Failure	403 Forbidden	Project name must be unique
Failure	404 Not Found	No project with this project_id exists
Failure	406 Not Acceptable	Accept header must be set to application/json

Request Body Example

```
{
  "name": "Issue API",
  "description": "Project and Issue Tracking API",
  "date": "5/25/22",
}
```

Response Examples

Success

```
Status: 200 OK
{
  "name": "Issue API",
  "date": "5/25/22",
  "description": "Project and Issue Tracking API",
  "user_id": "5685839489138688",
  "issues": [],
  "self": "http://127.0.0.1:8080/projects/5700985489981440",
  "id": "5700985489981440"
}
```

Failure

```
Status: 400 Bad Request
{
  "Error": "The request object is missing at least one of the required attributes"
}
```

```
Status: 401 Unauthorized
{
  "Error": "Access Denied. JWT sub does not match user token for project."
}
```

```
Status: 403 Forbidden
{
  "Error": "Project name must be unique"
}
```

```
Status: 404 Not Found
{
  "Error": "No project with this project_id exists"
}
```

```
Status: 406 Not Acceptable
{
  "Error": "Accepts type is Not Acceptable. Must be application/json."
}
```


PUT a Project

Allows you to edit all attributes of a project if the JWT associated with the user of the project is included in headers authorization.

PUT /projects/:project_id

Request

Path Parameters

Name	Description
project_id	ID of the project

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
name	Name of the project. (Project Name must be unique)	Yes
description	Description of the project.	Yes
date	Date project was started	Yes

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	Missing at least one required attribute in request body.
Failure	401 Unauthorized	JWT is missing or JWT does not match token for user.
Failure	403 Forbidden	Project name must be unique
Failure	404 Not Found	No project with this project_id exists
Failure	406 Not Acceptable	Accept header must be set to application/json

Request Body Example

```
{
  "name": "Issue API Put",
  "description": "Project and Issue Tracking API Put",
  "date": "5/25/22",
}
```

Response Examples

Success

```
Status: 200 OK
{
  "name": "Issue API Put",
  "date": "5/25/22",
  "description": "Project and Issue Tracking API Put",
  "user_id": "5685839489138688",
  "issues": [],
  "self": "http://127.0.0.1:8080/projects/5700985489981440",
  "id": "5700985489981440"
}
```

Failure

```
Status: 400 Bad Request
{
  "Error": "The request object is missing at least one of the required attributes"
}
```

```
Status: 401 Unauthorized
{
  "Error": "Access Denied. JWT sub does not match user token for project."
}
```

```
Status: 403 Forbidden
{
  "Error": "Project name must be unique"
}
```

```
Status: 404 Not Found
{
  "Error": "No project with this project_id exists"
}
```

```
Status: 406 Not Acceptable
{
  "Error": "Accepts type is Not Acceptable. Must be application/json."
}
```

PATCH a Project

Allows you to edit one or more attributes of a project if the JWT associated with the user of the project is included in headers authorization.

PATCH /projects/:project_id

Request

Path Parameters

Name	Description
project_id	ID of the project

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

At least one of these 3 attributes must be present in the request body.

Name	Description	Required?
name	Name of the project. (Project Name must be unique)	No
description	Description of the project.	No
date	Date project was started	No

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	Body must have at least one valid attribute
Failure	401 Unauthorized	JWT is missing or JWT does not match token for user.
Failure	403 Forbidden	Project name must be unique
Failure	404 Not Found	No project with this project_id exists
Failure	406 Not Acceptable	Accept header must be set to application/json

Request Body Example

```
{
  "name": "Issue API Patch"
}
```

Response Examples

Success

```
Status: 200 OK
{
  "name": "Issue API Patch",
  "date": "5/25/22",
  "description": "Project and Issue Tracking API",
  "user_id": "5685839489138688",
  "issues": [],
  "self": "http://127.0.0.1:8080/projects/5700985489981440",
  "id": "5700985489981440"
}
```

Failure

```
Status: 400 Bad Request
{
  "Error": "The request object must include at least one project attribute"
}
```

```
Status: 401 Unauthorized
{
  "Error": "Access Denied. JWT sub does not match user token for project."
}
```

```
Status: 403 Forbidden
{
  "Error": "Project name must be unique"
}
```

```
Status: 404 Not Found
{
  "Error": "No project with this project_id exists"
}
```

```
Status: 406 Not Acceptable
{
  "Error": "Accepts type is Not Acceptable. Must be application/json."
}
```

Delete a Project

Allows you to get an existing project if the JWT associated with the user of the project is included in headers authorization. When a project is deleted, the project_id is removed from the user.projects array. Then all issues related to the project are deleted. Finally the project is deleted.

DELETE /projects/:project_id

Request

Path Parameters

Name	Description
project_id	ID of the project

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	401 Unauthorized	JWT is missing or JWT does not match token for user.
Failure	404 Not Found	No project with this project_id exists

Response Examples

Success

Status: 204 No Content

Failure

Status: 401 Unauthorized { "Error": "Access Denied. JWT sub does not match user token for project." }
Status: 404 Not Found { "Error": "No project with this project_id exists" }

Get all Issues

Allows you to get all issues.

GET /issues/

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	No project with this project_id exists
Failure	406 Not Acceptable	Accept header must be set to application/json

Response Examples

Success

```
Status: 200 OK
{
  "issues": [
    {
      "priority": "low",
      "description": "POST Status Code received 200 instead of 201 from issue endpoint",
      "name": "Wrong Status Code",
      "project_id": "5090006997663744",
      "id": "5693366486433792",
      "self": "http://127.0.0.1:8080/issues/5693366486433792"
    }
  ]
}
```

Failure

```
Status: 404 Not Found
{
  "Error": "No project with this project_id exists"
}

Status: 406 Not Acceptable
{
  "Error": "Accepts type is Not Acceptable. Must be application/json."
}
```

Get an Issue

Allows you to get an issue by id.

GET /issues/:issue_id

Request

Path Parameters

Name	Description
issue_id	ID of the issue

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	No issue with this issue_id exists
Failure	406 Not Acceptable	Accept header must be set to application/json

Response Examples

Success

Status: 200 OK
<pre>{ "priority": "low", "description": "POST Status Code received 200 instead of 201 from issue endpoint", "name": "Wrong Status Code", "project_id": "5090006997663744", "id": "5693366486433792", "self": "http://127.0.0.1:8080/issues/5693366486433792" }</pre>

Failure

Status: 404 Not Found
<pre>{ "Error": "No project with this project_id exists" }</pre>
Status: 406 Not Acceptable
<pre>{ "Error": "Accepts type is Not Acceptable. Must be application/json." }</pre>

Post an Issue

Allows you to post an issue.

POST /issues

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
name	Name of the issue. (Project Name must be unique)	Yes
description	Description of the issue.	Yes
priority	Priority of the issue. (Low, Medium, High)	Yes
project_id	The id for the project associated with this issue.	Yes

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	Missing at least one required attribute in the request body.
Failure	403 Forbidden	Issue name must be unique
Failure	406 Not Acceptable	Accept header must be set to application/json

Request Body Example

```
{
  "name": "Wrong Status Code",
  "description": "POST Status Code received 200 instead of 201 from issue endpoint",
  "priority": "low",
  "project_id": "5730364475572224"
}
```


Response Examples

Success

```
Status: 200 OK
{
  "id": "5651673292734464",
  "name": "Wrong Status Code",
  "description": "POST Status Code received 200 instead of 201 from issue endpoint",
  "priority": "low",
  "project_id": "5730364475572224",
  "self": "http://127.0.0.1:8080/issues/5651673292734464"
}
```

Failure

```
Status: 400 Bad Request
{
  "Error": "The request object is missing at least one of the required attributes"
}
```

```
Status: 403 Forbidden
{
  "Error": "Issue name must be unique"
}
```

```
Status: 406 Not Acceptable
{
  "Error": "Accepts type is Not Acceptable. Must be application/json."
}
```

PUT an Issue

Allows you to edit all attributes of an issue.

PUT /issues/:issue_id

Request

Path Parameters

Name	Description
issue_id	ID of the issue

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
name	Name of the issue. (Project Name must be unique)	Yes
description	Description of the issue.	Yes
priority	Priority of the issue. (Low, Medium, High)	Yes

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	Missing at least one required attribute in request body.
Failure	403 Forbidden	Issue name must be unique
Failure	404 Not Found	No issue with this issue_id exists
Failure	406 Not Acceptable	Accept header must be set to application/json

Request Body Example

```
{
  "name": "Wrong Status Code ",
  "description": "POST Status Code received 200 instead of 201 from issue endpoint",
  "priority": "medium"
}
```

Response Examples

Success

```
Status: 200 OK
{
  "id": "5651673292734464",
  "name": "Wrong Status Code",
  "description": "POST Status Code received 200 instead of 201 from issue endpoint",
  "priority": "medium",
  "project_id": "5730364475572224",
  "self": "http://127.0.0.1:8080/issues/5651673292734464"
}
```

Failure

```
Status: 400 Bad Request
{
  "Error": "The request object is missing at least one of the required attributes"
}
```

```
Status: 403 Forbidden
{
  "Error": "Issue name must be unique"
}
```

```
Status: 404 Bad Request
{
  "Error": "No issue with this issue_id exists"
}
```

```
Status: 406 Not Acceptable
{
  "Error": "Accepts type is Not Acceptable. Must be application/json."
}
```

Patch an Issue

Allows you to edit one or more attributes on an issue.

PATCH /issues/:issue_id

Request

Path Parameters

Name	Description
issue_id	ID of the issue

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
name	Name of the issue. (Project Name must be unique)	No
description	Description of the issue.	No
priority	Priority of the issue. (Low, Medium, High)	No

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	Missing at least one required attribute in request body.
Failure	403 Forbidden	Issue name must be unique
Failure	404 Not Found	No issue with this issue_id exists
Failure	406 Not Acceptable	Accept header must be set to application/json

Request Body Example

<pre>{ "priority": "high" }</pre>

Response Examples

Success

```
Status: 200 OK
{
  "id": "5651673292734464",
  "name": "Wrong Status Code",
  "description": "POST Status Code received 200 instead of 201 from issue endpoint",
  "priority": "high",
  "project_id": "5730364475572224",
  "self": "http://127.0.0.1:8080/issues/5651673292734464"
}
```

Failure

```
Status: 400 Bad Request
{
  "Error": "The request object must include at least one issue attribute"
}
```

```
Status: 403 Forbidden
{
  "Error": "Issue name must be unique"
}
```

```
Status: 404 Bad Request
{
  "Error": "No issue with this issue_id exists"
}
```

```
Status: 406 Not Acceptable
{
  "Error": "Accepts type is Not Acceptable. Must be application/json."
}
```

Delete an Issue

Allows you to delete an issue. When deleting an issue the `issue_id` is removed from the project associated with it.

DELETE /issues/:issue_id

Request

Path Parameters

Name	Description
issue_id	ID of the issue

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	404 Not Found	No issue exists with issue_id

Response Examples

Success

Status: 204 OK

Failure

Status: 404 Bad Request { "Error": "No issue with this issue_id exists" }
--