

# Ensemble

A collection of all of the Ensemble endpoints.

## POST Create a new User

localhost:3000/api/projects

POST /api/users

REQUIRES: email (Unique, not null) first\_name (not null) last\_name (not null)  
password (not null, less than 72 characters)

RETURNS: JSON Formatted message id: User-ID email: User email  
first\_name: User first name last\_name: User last name

ERROR MESSAGES:

### HEADERS

#### Content-Type

application/json

### BODY

```
{
  "email" : "email",
  "first_name" : "First Name",
  "last_name" : "Last Name",
  "password" : "password"
}
```

### Sample Request

Create a new User

```
curl --request POST \
  --url localhost:3000/api/projects \
  --header 'content-type: application/json' \
  --data '{
    "email" : "email",
    "first_name" : "First Name",
    "last_name" : "Last Name",
    "password" : "password"
  }'
```

## POST Login a User

localhost:3000/api/login

POST /api/login

REQUIRES: email (Unique, not null) password (not null, less than 72 characters)

RETURNS: JSON Formatted message id: User-ID email: User email first\_name: User first name last\_name: User last name token: User Token

ERROR MESSAGES:

HEADERS

Content-Type

application/json

BODY

```
{
  "email" : "user@example.com",
  "password" : "password"
}
```

Sample Request

Login a User

```
curl --request POST \
  --url localhost:3000/api/login \
  --header 'content-type: application/json' \
  --data '{
    "email" : "user@example.com",
    "password" : "password"
  }'
```

**DELETE** Logout a User

localhost:3000/api/logout

DELETE /api/logout

REQUIRES: Do not send a token with the request

RETURNS: JSON Formatted message "Logout complete."

ERROR MESSAGES: N/A

HEADERS

Content-Type

application/json

Sample Request

Logout a User
<pre>curl --request DELETE \   --url localhost:3000/api/logout \   --header 'content-type: application/json'</pre>

**POST** Create a new Project

localhost:3000/api/projects

POST /api/projects

REQUIRES: title (string, Not null) description (text, not null) author (string, not null) project\_num (string, not null, less than 72 characters) token (Logged In User's Token)

RETURNS: JSON Formatted message id: Project-ID title: Project Title description: Project Description author: Project Author

id: User-ID email: User email first\_name: User first name last\_name: User last name

ERROR MESSAGES: Will return error messages if a required field isn't filled out.

HEADERS

**Content-Type**  
application/json

BODY

```
{  
  "title" : "Title",  
  "description" : "Description",  
  "author" : "Author",  
  "project_num" : "ProjectNum",  
  "token" : "UserToken"  
}
```

Sample Request

Create a new Project
<pre>curl --request POST \   --url localhost:3000/api/projects \   --header 'content-type: application/json' \   --data '{</pre>

```
"title" : "Title",
"description" : "Description",
"author" : "Author",
"project_num" : "ProjectNum",
"token" : "UserToKen"
}'
```

## POST Invite a User to a Project

```
localhost:3000/api/projects/:token/invitations
```

POST localhost:3000/api/projects/:project\_token/invitations

REQUIRES :project\_token :email (E-mail address of User to share project with)

RETURNS: JSON formatted message: a) ":email has been added to Project successfully!" b) ":email does not have an account and will have to sign up."

This will e-mail the users after the above checks are performed inviting them to sign up or view their project.

### HEADERS

#### Content-Type

application/json

### BODY

```
{
  "email" : "invited_user@example.com"
}
```

### Sample Request

Invite a User to a Project
<pre>curl --request POST \   --url localhost:3000/api/projects/:token/invitations \   --header 'content-type: application/json' \   --data '{     "email" : "invited_user@example.com"   }'</pre>

## GET Show a Project

```
localhost:3000/api/projects/:id?token=:user_token
```

GET /api/projects/:id?token=:token

REQUIRES: :id (Project ID) :token (Logged In User's Token)

RETURNS: JSON Formatted message { project\_id: Project ID, token: Project Token, description: Project Description, title: Title, author: Author for Project, project\_num: Project Number owner: { id: User ID first\_name: Owner Name last\_name: Owner Last Name }, members: [{ "id": Member ID, "first\_name": "Member First Name", "last\_name": "Member Last Name" }], comments: [{"left\_comments": [ { "id": "Comment ID", "body": "Comment Body", "user": { "id": 1, "first\_name": "User First Name", "last\_name": "User Last Name" }, "comments": [ { "id": Comment ID, "body": "Comment Body", "user": { "id": User ID, "first\_name": "User First Name", "last\_name": "User Last Name" }, "comments": [ { "id": Comment ID, "body": "Comment Body", "user": { "id": User ID, "first\_name": "User First Name", "last\_name": "User Last Name" }, "comments": [ ] } ] } ] } ] } ] } } } ERROR MESSAGES:

HEADERS

Content-Type

application/json

Sample Request

Show a Project

```
curl --request GET \
  --url 'localhost:3000/api/projects/:id?token=%3Auser_token' \
  --header 'content-type: application/json'
```

PATCH Edit a Project

localhost:3000/api/projects/:id?token=current\_usertoken

PATCH /api/projects/:id?token=:token

REQUIRES: :id (Project ID) :token (Project Owner's Token)

RETURNS: JSON Formatted message { project\_id: Project ID, token: Project Token, description: Project Description, title: Title, author: Author for Project, project\_num: Project Number owner: { id: User ID first\_name: Owner Name last\_name: Owner Last Name }, members: [{ "id": Member ID, "first\_name": "Member First Name", "last\_name": "Member Last Name" }], comments: [{"left\_comments": [ { "id": "Comment ID", "body": "Comment Body", "user": { "id": User ID, "first\_name": "User First Name", "last\_name": "User Last Name" }, "comments": [ { "id": Comment ID, "body": "Comment Body", "user": { "id": User ID, "first\_name": "User First Name", "last\_name": "User Last Name" }, "comments": [ { "id": Comment ID, "body": "Comment Body", "user": { "id": User ID, "first\_name": "User First Name", "last\_name": "User Last Name" }, "comments": [ ] } ] } ] } ] } ] } } } ERROR MESSAGES:

HEADERS

Content-Type

application/json

BODY

---

```
{
  "description": "description",
  "title": "title",
  "author": "author",
  "project_num": "project_num",
}
```