ONBOARD PROJECT DOCUMENTATION

Mentor: Hiep Nguyen

Trainee: Troy

2019

TABLE OF CONTENTS

I. API Documentation

1. Default error response:

- For each error response, the default form would be like this:

Name	Type	Description
message	String	The general name for the error
description	String	The detail of the error

- An error response example:

```
{
   "description": "Item with this id doesn't exist.",
   "message": "Not Found."
}
```

2. **Authentication**
a. **POST**

/auth

Allow client to authenticate by using email and password.

REQUEST

- **Parameters**: No content
- **Header**:

Key	**Value**
Content-Type	application/json

- **Body**:

Name	**Type**	**Description**	**Example**
email	String	Email of the user	admin@admin.com
password	String	Password of the user	123456

SUCCESSFUL RESPONSE

- **HTTP code**: **200 OK**
- **Headers**:

Key	**Value**
Content-Type	application/json

- **Response example**:

```
{
   "access_token":
"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpYXQiOjE1NzEyODAxMTEsIm5iZiI6MTU3MTI4MDExMS
wianRpIjoiMzU3YWJjNTYtNjRjOC00MzAzLWE3YmMtMDIxNmNmYWVhYmI4IiwiZXhwIjoxNTcxMjgxMDExLC
JpZGVudGl0eSI6MSwiZnJlc2giOmZhbHNlLCJ0eXBlIjoiYWNjZXNzIn0.M_r1meDSt_jpSW3SPjI5kVwyCQ
mgPKoBoPM-59NYJVU",
   "id": 1
}
```

ERROR RESPONSES

- Validation Error (400): If email/ password of the body doesn't pass the validation test.
- Not Found (404): If client try to login with an unregistered email.

3. **Category**

a. **GET**

/categories

Allow client to get all categories with pagination.

REQUEST

- **Parameters**:

Name	**Type**	**Description**	**Example**	**Default**
page	Int	The page of categories that client wants to get	1	1
size	Int	Number of categories each page has	5	5

- **Header**:

Key	**Value**
Content-Type	application/json

- **Body**: No content

SUCCESSFUL RESPONSE

- **HTTP code: 200 OK**
- **Headers**:

Key	**Value**
Content-Type	application/json

- **Response example**:

```
{
   "current_page": 1,
   "data": [
       {
           "created": "2019-10-17T09:42:04",
           "creator_id": 1,
           "description": "Minecraft stuffs...",
           "id": 1,
           "title": "Minecraft",
           "updated": "2019-10-17T09:42:04"
       },
       {
           "created": "2019-10-17T09:42:24",
           "creator_id": 1,
           "description": "Seafood stuffs...",
           "id": 2,
           "title": "Seafood",
           "updated": "2019-10-17T09:42:24"
       }
   ],
   "per_page": 5,
   "total": 2
}
```

ERROR RESPONSES

b. **GET**

/categories/:id

Allow client to get category by id.

REQUEST

- **Parameters**:

Name	**Type**	**Description**	**Example**
id	Int	Identifier of the category that the client wants to get	1

- **Header**:

Key	**Value**
Content-Type	application/json

- **Body**: No content

SUCCESSFUL RESPONSE

- **HTTP code: 200 OK**
- **Headers**:

Key	**Value**
Content-Type	application/json

- **Response example**:

```
{
   "data": {
       "created": "2019-10-17T09:42:04",
       "creator_id": 1,
       "description": "Minecraft stuffs...",
       "id": 1,
       "title": "Minecraft",
       "updated": "2019-10-17T09:42:04"
   }
}
```

ERROR RESPONSES

- Not Found (404): If client tries to get a category that doesn't exist.

c. **GET**

/categories/:id/items

Allow client to get all items of the category with id with pagination.

REQUEST

- **Parameters**:

Name	**Type**	**Description**	**Example**	**Default**
id	Int	Identifier of the category in which that the client wants to get all items	1	**requires**
page	Int	Page of the category's items client queries	1	1
size	Int	Number of category's items client wants to get	5	5

- **Header**:

Key	**Value**
Content-Type	application/json

- **Body**: No content

SUCCESSFUL RESPONSE

- **HTTP code: 200 OK**
- **Headers**:

Key	**Value**
Content-Type	application/json

- **Response example**:

```
{
   "current_page": 1,
   "data": [
       {
           "created": "2019-10-17T09:49:51",
           "creator_id": 1,
           "description": "Very shiny...",
           "id": 1,
           "title": "Minecraft Light Block",
           "updated": "2019-10-17T09:49:51"
       },
       {
           "created": "2019-10-17T09:49:54",
           "creator_id": 1,
           "description": "Very shiny...",
           "id": 2,
           "title": "Minecraft Lamp Block",
           "updated": "2019-10-17T09:49:54"
       }
   ],
   "per_page": 5,
   "total": 2
}
```

ERROR RESPONSES

- Not Found (404): If client tries to get a category that doesn’t exist.

d. **POST**

/categories

Allow client to create a category.

REQUEST

- **Parameters**: No content
- **Header**:

Key	**Value**
Content-Type	application/json
Authorization	Bearer jwt_token

- **Body**:

Name	**Type**	**Description**	**Example**
title	String	Title of the category	Minecraft
description	String	Description of the category	Minecraft stuffs

SUCCESSFUL RESPONSE

- **HTTP code**: **200 OK**
- **Headers**:

Key	**Value**
Content-Type	application/json

- **Response example**:

```
{
   "id": 2
}
```

ERROR RESPONSES

- Validation Error (400): If title/description doesn’t pass the validation test.
- Unauthorized (401): If client hasn’t logged in yet.
- Duplicate Entity (404): If client tries to create a category with the title that has already used by another category.

e. **PUT**

/categories/:id

Allow client to update a category.

REQUEST

- **Parameters**:

Name	**Type**	**Description**	**Example**
id	Int	Identifier of the category that the client wants to update	2

- **Header**:

Key	**Value**
Content-Type	application/json
Authorization	Bearer jwt_token

- **Body**:

Name	**Type**	**Description**	**Example**
title	String	Title of the category	Minecraft
description	String	Description of the category	Minecraft stuffs

SUCCESSFUL RESPONSE

- **HTTP code**: **200 OK**
- **Headers**:

Key	**Value**
Content-Type	application/json

- **Response example**:

```
{
   "id": 2
}
```

ERROR RESPONSES

- Validation Error (400): If title/description doesn’t pass the validation test.
- Duplicated Entity (400): If a category with the title has already existed.
- Unauthorized (401): If client hasn’t logged in yet.
- Forbidden (403): If client tries to update another user’s category.
- Not Found (404): If client tries to update a category that doesn’t exist.

f. **DELETE**

/categories/:id

Allow client to delete a category, all the items that belong to this category will be deleted also.

REQUEST

- **Parameters**:

Name	**Type**	**Description**	**Example**
id	Int	Identifier of the category that the client wants to delete	1

- **Header**:

Key	**Value**
Content-Type	application/json
Authorization	Bearer jwt_token

- **Body**: No content

SUCCESSFUL RESPONSE

- **HTTP code**: **204 No Content**
- **Headers**:

Key	**Value**
Content-Type	application/json

ERROR RESPONSES

- Unauthorized (401): If client hasn’t logged in yet.
- Forbidden (403): If client tries to delete another user’s category.
- Not Found (404): If client tries to delete a category that doesn’t exist.

4. **Item**

a. **GET**

/items

Allow client to get all categories with pagination.

REQUEST

- **Parameters**:

Name	**Type**	**Description**	**Example**	**Default**
page	Int	The page of items that client wants to get	1	1
size	Int	Number of items each page has	5	5

- **Header**:

Key	**Value**
Content-Type	application/json

- **Body**: No content

SUCCESSFUL RESPONSE

- **HTTP code: 200 OK**
- **Headers:**

Key	**Value**
Content-Type	application/json

- **Response example:**

```
{
   "current_page": 1,
   "data": [
       {
           "category": {
                 "id": 1,
                 "title": "Minecraft",
           },
           "created": "2019-10-17T09:49:51",
           "creator_id": 1,
           "description": "Very shiny...",
           "id": 1,
           "title": "Minecraft Light Block",
           "updated": "2019-10-17T09:49:51"
       },
       {
           "category": {
                 "id": 1,
                 "title": "Minecraft",
           },
           "created": "2019-10-17T09:49:54",
           "creator_id": 1,
           "description": "Very shiny...",
           "id": 2,
           "title": "Minecraft Lamp Block",
           "updated": "2019-10-17T09:49:54"
       }
   ],
   "per_page": 5,
   "total": 2
}
```

ERROR RESPONSES

b. **GET**

/items/:id

Allow client to get category by id.

REQUEST

- **Parameters**:

Name	Type	Description	Example
id	Int	Identifier of the category that the client wants to get	2

- **Header**:

Key	Value
Content-Type	application/json

- **Body**: No content

SUCCESSFUL RESPONSE

- **HTTP code**: **200 OK**
- **Headers**:

Key	**Value**
Content-Type	application/json

- **Response example**:

```
{
   "data": {
          "category": {
                "id": 1,
                "title": "Minecraft",
          },
          "created": "2019-10-17T09:49:54",
          "creator_id": 1,
          "description": "Very shiny...",
          "id": 2,
          "title": "Minecraft Lamp Block",
          "updated": "2019-10-17T09:49:54"
       }
}
```

ERROR RESPONSES

- Not Found (404): If client tries to get an item that doesn't exist.

c. **POST**

/items

Allow client to create an item.

REQUEST

- **Parameters**: No content
- **Header**:

Key	**Value**
Content-Type	application/json
Authorization	Bearer jwt_token

- **Body**:

Name	**Type**	**Description**	**Example**
title	String	Title of the item	Minecraft Sword
description	String	Description of the item	Yeah awesome...
category_id	Int	Identifier of the category that this item will belong to	1

SUCCESSFUL RESPONSE

- **HTTP code**: **200 OK**
- **Headers**:

Key	**Value**
Content-Type	application/json

- **Response example**:

```
{
   "id": 2
}
```

ERROR RESPONSES

- Validation Error (400): If title/description/category_id doesn’t pass the validation test.
- Duplicate Entity (400): If client tries to create an item with the title that has already used by another item.
- Unauthorized (401): If client hasn’t logged in yet.
- Not Found (404): If the category with that category_id doesn’t exist.

d. **PUT**

/items/:id

Allow client to update a category.

REQUEST

- **Parameters**:

Name	**Type**	**Description**	**Example**
id	Int	Identifier of the item that the client wants to update	1

- **Header**:

Key	**Value**
Content-Type	application/json
Authorization	Bearer jwt_token

- **Body**:

Name	**Type**	**Description**	**Example**
title	String	Title of the item	Minecraft Swordy
description	String	Description of the item	Minecraft is dug

SUCCESSFUL RESPONSE

- **HTTP code**: **200 OK**
- **Headers**:

Key	**Value**
Content-Type	application/json

- **Response example**:

```
{
   "id": 2
}
```

ERROR RESPONSES

- Validation Error (400): If title/description/category_id doesn't pass the validation test.
- Duplicated Entity (400): If a category with the title has already existed.
- Unauthorized (401): If client hasn't logged in yet.
- Forbidden (403): If client tries to update another user's item.
- Not Found (404): If the item with that id doesn't exist or the category having category_id doesn't exist.

e. **DELETE**

/items/:id

Allow client to delete a category.

REQUEST

- **Parameters**:

Name	**Type**	**Description**	**Example**
id	Int	Identifier of the item that the client wants to delete	1

- **Header**:

Key	**Value**
Content-Type	application/json
Authorization	Bearer jwt_token

- **Body**: No content

SUCCESSFUL RESPONSE

- **HTTP code**: **204 No Content**
- **Headers**:

Key	**Value**
Content-Type	application/json

ERROR RESPONSES

- Unauthorized (401): If client hasn't logged in yet.
- Forbidden (403): If client tries to delete another user's item.
- Not Found (404): If client tries to delete an item that doesn't exist.

5. **User**

a. **GET**

/users/:id

Allow login-ed client to see their user information.

REQUEST

- **Parameters**:

Name	**Type**	**Description**	**Example**
id	Int	Identifier of the client's user	1

- **Header**:

Key	**Value**
Content-Type	application/json
Authorization	Bearer jwt_token

- **Body**: No content

SUCCESSFUL RESPONSE

- **HTTP code**: **200 OK**
- **Headers**:

Key	**Value**
Content-Type	application/json

- **Response example**:

```
{
   "data": {
       "created": "2019-10-15T08:46:45",
       "email": "admin@gmail.com",
       "id": "1",
       "updated": "2019-10-15T08:46:45"
   }
}
```

ERROR RESPONSES

- Unauthorized (401): If client hasn’t logged in yet.
- Forbidden (403): If client tries to see other user’s profile.

b. **POST**

/users

Allow client to register.

REQUEST

- **Parameters**: No content
- **Header**:

Key	**Value**
Content-Type	application/json

- **Body**:

Name	**Type**	**Description**	**Example**
email	String	Email of the user	admin@admin.com
password	String	Password of the user	123456

SUCCESSFUL RESPONSE

- **HTTP code: 200 OK**
- **Headers**:

Key	**Value**
Content-Type	application/json

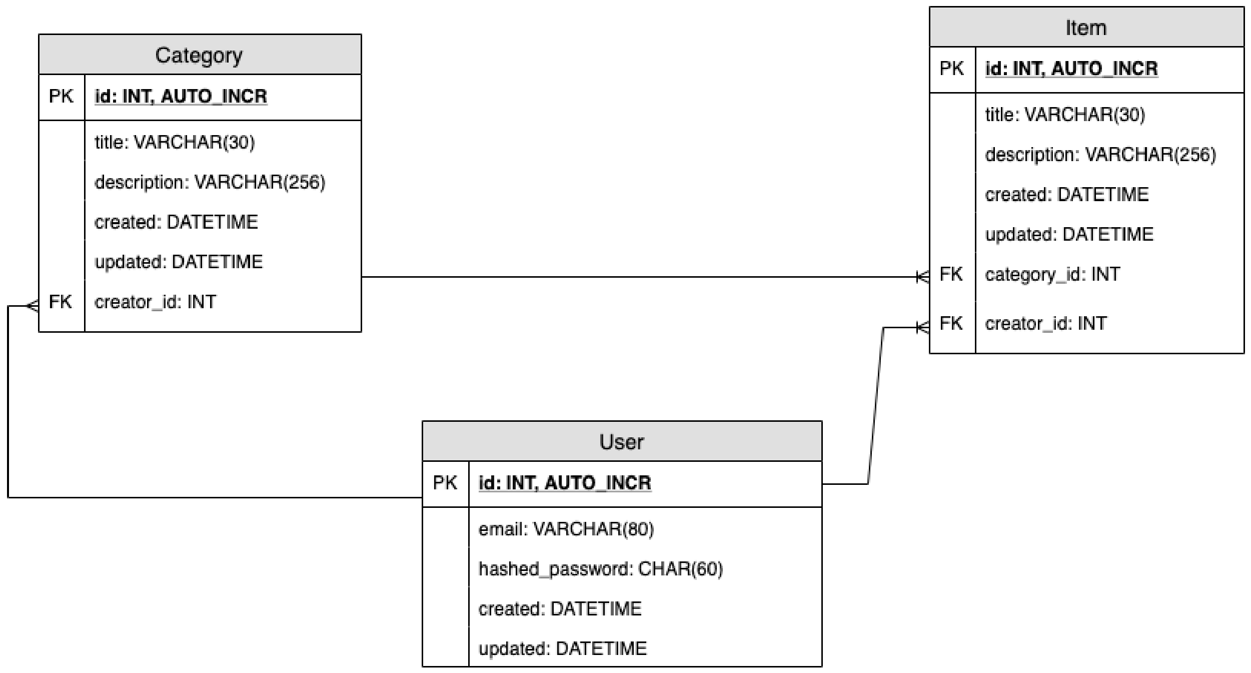
- **Response example**:

```
{{
   "access_token":
"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpYXQiOjE1NzEyODAxMTEsIm5iZiI6MTU3MTI4MDExMS
wianRpIjoiMzU3YWJjNTYtNjRjOC00MzAzLWE3YmMtMDIxNmNmYWVhYmI4IiwiZXhwIjoxNTcxMjgxMDExLC
JpZGVudGl0eSI6MSwiZnJlc2giOmZhbHNlLCJ0eXBlIjoiYWNjZXNzIn0.M_r1meDSt_jpSW3SPjI5kVwyCQ
mgPKoBoPM-59NYJVU",
   "id": 1
}
```

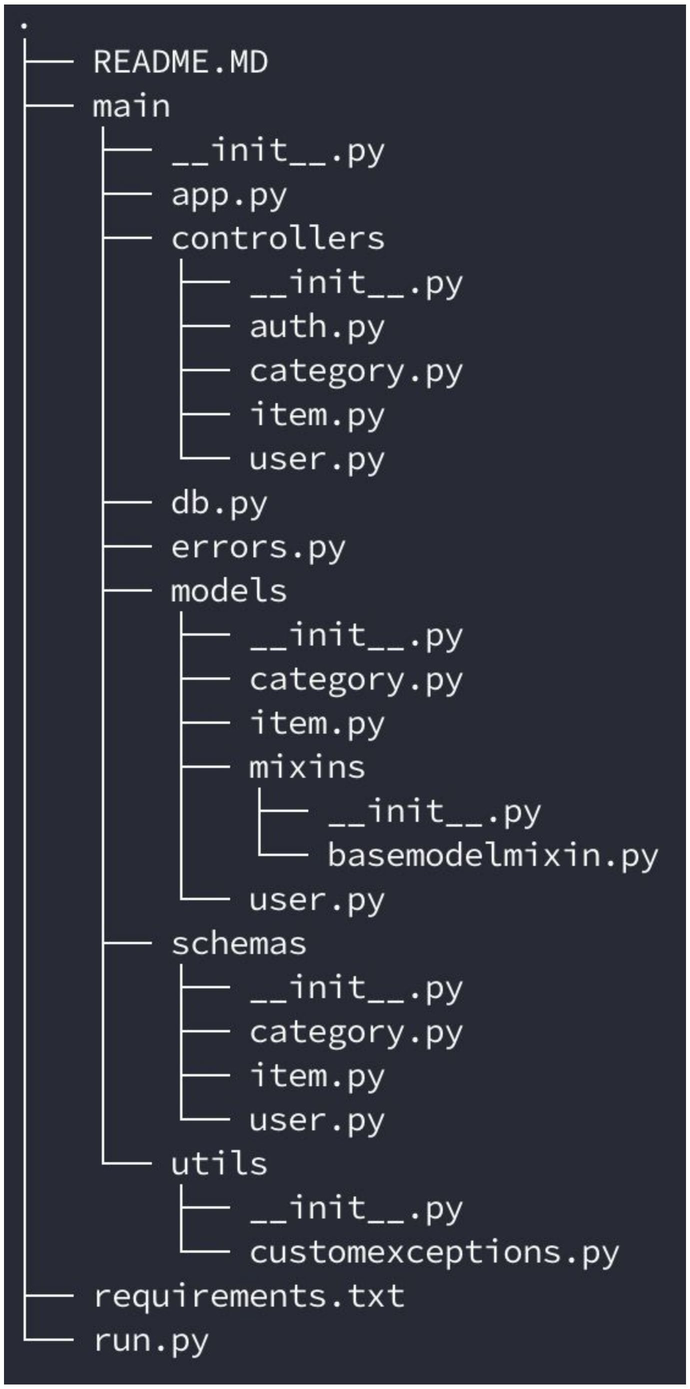
ERROR RESPONSES

- Validation Error (400): If email/password doesn't pass the validation test.
- Duplicate Entity (404): If client tries to create a user with the email that has already used by another user.

II. Entity Relationship Diagram



III. Folder Structure



```
.
├── README.MD
├── main
│   ├── __init__.py
│   ├── app.py
│   ├── controllers
│   │   ├── __init__.py
│   │   ├── auth.py
│   │   ├── category.py
│   │   ├── item.py
│   │   └── user.py
│   ├── db.py
│   ├── errors.py
│   ├── models
│   │   ├── __init__.py
│   │   ├── category.py
│   │   ├── item.py
│   │   ├── mixins
│   │   │   ├── __init__.py
│   │   │   └── basemodelmixin.py
│   │   └── user.py
│   ├── schemas
│   │   ├── __init__.py
│   │   ├── category.py
│   │   ├── item.py
│   │   └── user.py
│   └── utils
│       ├── __init__.py
│       └── customexceptions.py
├── requirements.txt
└── run.py
```