TODO REST API DOCUMENTATION

Table of Contents

User-Routes

> ### Register

- URL: `/auth/register`
- Method: `PUT`
- URL Params: `none`
- Success Response:
  - Code: `201`
  - Content: `{ "_id": "[id]", "accessToken": "[JWT TOKEN]" }`
- Error Response 1:
  - Code: `409`
  - Content: `{ "error": "Username already exists." }`
- Error Response 2:
  - Code: `406`
  - Content: `{ "error": Username and password must be specified." }`
- Error Response 3:
  - Code: `400`
- Sample call:

```
axios.put('http://localhost:4000/auth/register', {
  username: 'berke',
  password: 'pass',
});
```

> ### Login

- URL: `/auth/login`
- Method: `POST`
- URL Params: `none`
- Success Response:
  - Code: `200`
  - Content: `{ "accessToken": "[JWT TOKEN]" }`
- Error Response 1:
  - Code: `403`
  - Content: `{ "error": "User not found." }`
- Error Response 2:
  - Code: `403`
  - Content: `{ "error": "Invalid credentials." }`
- Error Response 3:
  - Code: `406`
  - Content: `{ "error": Username and password must be specified." }`
- Sample call:

```
axios.post('http://localhost:4000/auth/login', {
  username: 'berke',
  password: 'pass',
});
```

Get Profile

- URL: `/auth/me`
- Method: `POST`
- URL Params: `none`
- Query Params: `?expand_todo=[true/1]`
- Headers: `Authorization: "Bearer [TOKEN]"`
- Success Response:
  - Code: `200`
  - Content: `{ "username": "berkegokmen1", "createdAt": "XXX", todolist: [] }`
- Sample call:

```
axios.get(
  'http://localhost:3000/auth/me?expand_todo=1',
  {},
  {
    headers: { Authorization: `Bearer [TOKEN]` },
  }
);
```

Logout

- URL: `/auth/logout`

- Method: `POST`
- URL Params: `none`
- Headers: `Authorization: "Bearer [TOKEN]"`
- Success Response:
  - Code: `200`
  - Content: `{ "message": "Logout successful" }`
- Error Response 1:
  - Code: `401`
  - Content: `{ "error": "jwt expired" }`
- Error Response 2:
  - Code: `400`
  - Content: `{ "error": "Invalid format for the token." }`
- Sample call:

```
axios.post(
  'http://localhost:3000/auth/logout',
  {},
  {
    headers: { Authorization: `Bearer [TOKEN]` },
  }
);
```

Logout All

- URL: `/auth/logoutall`
- Method: `POST`
- URL Params: `none`
- Headers: `Authorization: "Bearer [TOKEN]"`
- Success Response:
  - Code: `200`
  - Content: `{ "message": "Logout from all successful" }`
- Error Response 1:
  - Code: `401`
  - Content: `{ "error": "jwt expired" }`
- Sample call:

```
axios.post(
  'http://localhost:3000/auth/logoutall',
  {},
  {
    headers: { Authorization: `Bearer [TOKEN]` },
  }
);
```

Todo-Routes

Create

- URL: `/todo/me`
- Method: `PUT`
- URL Params: `none`
- Headers: `Authorization: "Bearer [TOKEN]"`
- Success Response:
  - Code: `201`
  - Content: `{ "_id": "[id]" }`
- Error Response 1:
  - Code: `400`
  - Content: `{ "error": "Title must be specified." }`
- Sample call:

```
axios.put(
  'http://localhost:4000/todo/me',
  {
    title: 'create rest api documantation',
    description: ':)',
  },
  {
    headers: { Authorization: `Bearer [TOKEN]` },
  }
);
```

Get Todos

- URL: `/todo/me`
- Method: `GET`
- URL Params: `none`
- Headers: `Authorization: "Bearer [TOKEN]"`
- Success Response:
  - Code: `200`
  - Content: `{ "todos" : [] }`
- Sample call:

```
axios.get(
  'http://localhost:4000/todo/me',
  {},
  {
    headers: { Authorization: `Bearer [TOKEN]` },
  }
);
```

Get Todo by ID

- URL: `/todo/:id`
- Method: `GET`
- URL Params: `id`
- Headers: `Authorization: "Bearer [TOKEN]"`
- Success Response:
  - Code: `200`
  - Content: `{ "_id": "[id]", "title": "create documentation", "description": "for textoni-todo-rest" }`
- Sample call:

```
axios.get(
  'http://localhost:4000/todo/me',
  {},
  {
    headers: { Authorization: `Bearer [TOKEN]` },
  }
);
```

Search

- URL: `/todo/me/search`
- Method: `PUT`
- URL Params: `none`
- Query Params: `?q=[QUERY STRING]`
- Headers: `Authorization: "Bearer [TOKEN]"`
- Success Response:
  - Code: `200`
  - Content: `{ "todos": [] }`
- Error Response 1:
  - Code: `400`
  - Content: `{ "error": "Please provide a query string." }`
- Error Response 2:
  - Code: `400`
  - Content: `{ "error": "Query string cannot be longer than 16 characters." }`
- Sample call:

```
axios.get(
  'http://localhost:4000/todo/me/search?q=rest',
  {},
  {
    headers: { Authorization: `Bearer [TOKEN]` },
  }
);
```

Update Todo

- URL: `/todo/:id`
- Method: `POST`
- URL Params: `id`
- Headers: `Authorization: "Bearer [TOKEN]"`
- Success Response:
  - Code: `200`
  - Content: `{ "message": "Todo updated" }`
- Error Response 1:
  - Code: `400`
  - Content: `{ "error": "New title or description must be specified." }`
- Error Response 2:
  - Code: `404`
  - Content: `{ "error": "Todo not found." }`
- Sample call:

```
axios.post(
  'http://localhost:4000/todo/626bf65c90cdc47981018724',
  {
    title: 'new title',
    description: 'new desc',
  },
  {},
  {
    headers: { Authorization: `Bearer [TOKEN]` },
  }
);
```

Delete Todo

- URL: `/todo/:id`
- Method: `POST`
- URL Params: `id`
- Headers: `Authorization: "Bearer [TOKEN]"`
- Success Response:
  - Code: `200`
  - Content: `{ "message": "Todo removed." }`
- Error Response 1:
  - Code: `404`
  - Content: `{ "error": "Todo not found." }`
- Sample call:

```
axios.delete(
  'http://localhost:4000/todo/626bf65c90cdc47981018724',
  {},
  {
    headers: { Authorization: `Bearer [TOKEN]` },
    }
  );
```