

Django Banking App Endpoints

Endpoint: `/accounts/register/`

Methods: `GET`, `POST`

Fields/payload: `username`, `password1`, `password2`, `email`, `first_name`, `last_name`

Success URL: `/accounts/login/`

Validation errors:

- `The two password fields didn't match`
- `A user with that username already exists`
- `This password is too short. It must contain at least 8 characters`
- `Enter a valid email address`
- Username, password, and repeat password are required. If empty, the following error is given above the problematic field(s): `This field is required`

Notes:

- All fields can be at most 100 characters long.

Endpoint: `/accounts/login/`

Methods: `GET`, `POST`

Fields/payload: `username`, `password`

Success URL: `/accounts/profile/view/`

Invalid inputs:

- `Username or password is invalid`

Notes: If the credentials are valid, Django's session authentication is used to log the user in.

Endpoint: `/accounts/logout/`

Methods: `GET`

Description: Log the user's session out of Django. If the user is unauthenticated, simply ignore the request. In either case, return a redirect to `/accounts/login/`

Endpoint: `/accounts/profile/view/`

Methods: `GET`

Fields: `id`, `username`, `email`, `first_name`, `last_name`

Description: Return a JSON response with the above fields, disclosing data of the logged-in user

Example response:

```
{"id": 1, "username": "Demon", "email": "demon@gmail.com", "first_name": "", "last_name": ""}
```

Additional notes:

- If the request is unauthenticated (user is not logged in), HTTP 401 UNAUTHORIZED is return.
-

Endpoint: `/accounts/profile/edit/`

Methods: `GET`, `POST`

Fields/payload: `first_name`, `last_name`, `email`, `password1`, `password2`

Success URL: `/accounts/profile/view/`

Validation errors:

- `The two password fields didn't match`
- `This password is too short. It must contain at least 8 characters`
- `Enter a valid email address`

Additional notes:

- At GET requests, the form (except for the password fields) are pre-filled with the current logged in user's values.
 - At the POST requests, no field is required. If any field is empty or missing, the corresponding value is updated to empty, with the only exception of passwords. If `password1` and `password2` are empty, it means that the user did not want to change its password; so, no changes will be made to the user's password.
 - The redirect shows the user's updated info.
 - If the request is unauthenticated (user is not logged in), HTTP 401 UNAUTHORIZED is return.
-

Endpoint: `/banks/add/`

Methods: `GET`, `POST`

Fields/payload: `name`, `description`, `inst_num`, `swift_code`

Success URL: `/banks/<bank_id>/details/`

Validation errors:

- If a field's input has more than 100 characters, the following error is placed above the field: `Ensure this value has at most 100 characters (it has <X>)`, where `<X>` is the current input's length.
- All fields are mandatory and cannot be left blank. If so, the following error is shown above the problematic field(s): `This field is required`

Additional notes:

- The owner of the new bank is the current logged-in user.
 - `bank_id` is the ID of the newly-created bank.
 - If the request is unauthenticated (user is not logged in), HTTP 401 UNAUTHORIZED is returned.
-

Endpoint: `/banks/<bank_id>/branches/add/`

Methods: `GET`, `POST`

Fields/payload: `name`, `transit_num`, `address`, `email`, `capacity`

Success URL: `/banks/branch/<branch_id>/details/`

Validation errors:

- `This field is required` (for all fields except for capacity)
- If a field's input has more than 100 characters, the following error is placed above the field: `Ensure this value has at most 100 characters (it has <X>)`, where `<X>` is the current input's length. This applies to all fields except for email and capacity.
- If capacity is negative, `Ensure this value is greater than or equal to 0` is placed above it.
- `Enter a valid email address`

Additional notes:

- If the request is unauthenticated (user is not logged in), HTTP 401 UNAUTHORIZED is returned.
- The form prefills the email input with the default value. The user is able to edit it.
- The bank to which the new branch belongs to is `bank_id`. If no bank exists with such ID, both GET and POST return 404 NOT FOUND.

- The user must be the **owner** of the corresponding bank; otherwise, a 403 FORBIDDEN is returned (both GET and POST). In other words, users are not able to add a branch to someone else's bank.
- The `branch_id` in the success URL is the ID of the newly-created branch.

Endpoint: `/banks/all/`

Methods: `GET`

Description: An HTML response is returned that lists all banks. The page displays the **name** and **institution number** of each bank in an unordered list, where each name links to the details page of the corresponding bank.

Example Response:

- [TD](#) Institution Number: 1
- [BMO](#) Institution Number: 2
- [RBC](#) Institution Number: 3

Endpoint: `/banks/<bank_id>/details/`

Methods: `GET`

Description: An HTML response is returned that contains the information of a specific bank (or a 404 NOT FOUND if it does not exist) and some info about its branches. The bank information shown is the bank **name**, **description**, **swift code**, **institution number**, and its **branches**. The branch information is displayed in a table, where each row contains a branch's **name**, **transit number**, and **address**. If the bank does not have any branches, `No branch found` is returned in the branches table.

Example response:

BMO

Bank of montreal

Swift code: BOFMCAM2

Inst number: 001

Branches:

name	transit number	address
King & University	24182	980 University Ave
Yonge & Queen	24872	29 Queen St W

Endpoint: `/banks/branch/<branch_id>/details/`

Methods: `GET`

Description: A JSON response that returns the detailed info of a branch (or a 404 NOT FOUND if it does not exist).

Fields: `id`, `name`, `transit_num`, `address`, `email`, `capacity`, `last_modified`

Example response:

```
{ "id": 1, "name": "jkl", "transit_num": "12", "address": "jlk", "email": "admin@utoronto.ca", "capacity": 12, "last_modified": "2022-01-08T22:30:40.419Z" }
```

Additional notes:

- If the request is unauthenticated (user is not logged in), HTTP 401 UNAUTHORIZED is return.
-

Endpoint: `/banks/branch/<branch_id>/edit/`

Methods: `GET`, `POST`

Fields/payload: `name`, `transit_num`, `address`, `email`, `capacity`

Success URL: `/banks/branch/<branch_id>/details/`

Validation errors:

- `This field is required` (for all fields except for capacity)
- If a field's input has more than 100 characters, the following error is displayed above the field: `Ensure this value has at most 100 characters (it has <X>)`, where `<X>` is the current input's length. This applies to all fields except for email and capacity.
- If capacity is negative, `Ensure this value is greater than or equal to 0` is displayed above it.
- `Enter a valid email address`

Additional notes:

- If the request is unauthenticated (user is not logged in), HTTP 401 UNAUTHORIZED is return.
- The form is pre-filled with the current values of the branch instance.
- The branch's bank is not changeable; thus, it is not part of the form.
- The user must be the **owner** of the corresponding bank; otherwise, a 403 FORBIDDEN response is returned (both GET and POST). In other words, users must not be able to modify someone else's branches

