

BuyNothing is an application which allows users to transact gifts, send asks and give thanks to other users on the platform. It essentially works like a CashApp or Venmo, the only difference is money doesn't get traded. My task was to build the backend server that will support the application infrastructure. This will consist of creating, updating, and deleting accounts on the app, generating gift transactions that can be seen from both users, allowing users to make ask requests, creating thank you messages, and notes that can be stored for the users convenience. Additionally, there will be an admin point of view of all the data involved. As admin, you can see all the accounts, gifts, thanks, and notes.

The framework of the application is built on RESTful APIs. When the server is up, users can complete actions through http requests with additional information such as path and body parameters. The API was built using the Java programming language on the Eclipse IDE. Here are some things one can do with this backend server:

ACCOUNTS:

Creating Accounts:

The screenshot shows a REST client interface with a POST request to `http://www.localhost:8080/rest-BN/api/accounts`. The request body is in JSON format, containing user details. The interface includes tabs for Params, Auth, Headers (8), Body (selected), Pre-req., Tests, and Settings. On the right, there are buttons for 'Send', 'Cookies', and 'Beautify'. The JSON body is as follows:

```
1 {
2   "uid" : "62",
3   "zip" : "60073",
4   "name" : "tabor",
5   "street" : "3433 south",
6   "phone" : "224-403-9894",
7   "isActive" : "false"
```

Getting Accounts:

GET

▼

http://www.localhost:8080/rest-BN/api/accounts

Send

▼

ParamsAuthHeaders (8)Body ●Pre-req. TestsSettings

raw ▼JSON ▼

Body ▼

200 OK25 ms476 BSave Response ▼

PrettyRawPreviewVisualizeJSON ▼

```
1  [
2    {
3      "uid": "62",
4      "isActive": false,
5      "userAsks": [],
6      "userGives": [],
7      "userThanks": [],
8      "thanks_to_this_account": [],
9      "userNotes": [],
10     "name": "tabor",
11     "phone": "224-403-9894",
12     "picture": " ",
13     "zip": "60073",
14     "street": "3433 south",
15     "dateCreated": "2023-03-15"
16   }
```

Activate Account:

PUT ▼ http://www.localhost:8080/rest-BN/api/accounts/62/activate Send ▼

Params Auth Headers (8) **Body** ● Pre-req. Tests Settings

raw ▼ **JSON** ▼ Cookies Beautify

Response Body:

Body ▼ 200 OK 11 ms 123 B Save Response ▼

Pretty Raw Preview Visualize Text ▼ 🔍

1

Body ▼ 200 OK 9 ms 475 B Save Response ▼

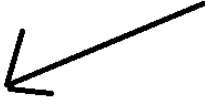
Pretty Raw Preview Visualize **JSON** ▼ 🔍

```
1 {
2   "uid": "62",
3   "isActive": true,
4   "userAsks": [],
5   "userGives": [],
6   "userThanks": [],
7   "thanks_to_this_account": [],
8   "userNotes": [],
9   "name": "tabor",
10  "phone": "224-403-9894",
11  "picture": " ",
12  "zip": "60073",
13  "street": "3433 south",
14  "dateCreated": "2023-03-16"
15 }
16
```

Body ▼ 200 OK 9 ms 475 B Save Response ▼

Pretty Raw Preview Visualize **JSON** ▼ 🔍

```
1 {
2   "uid": "62",
3   "isActive": true,
4   "userAsks": [],
5   "userGives": [],
6   "userThanks": [],
7   "thanks_to_this_account": [],
8   "userNotes": [],
9   "name": "tabor",
10  "phone": "224-403-9894",
11  "picture": " ",
12  "zip": "60073",
13  "street": "3433 south",
14  "dateCreated": "2023-03-16"
15 }
16
```



- Admin can deactivate accounts by using “deactivate” in the path params instead of “activate”

Delete Account:

GET

http://www.localhost:8080/rest-BN/api/accounts/62

Send

Params

Auth

Headers (8)

Body

Pre-req.

Tests

Settings

raw

JSON

Cookies

Beautify

All accounts in current environment:

GET

http://www.localhost:8080/rest-BN/api/accounts/62

Send

Params

Auth

Headers (8)

Body

Pre-req.

Tests

Settings

raw

JSON

Cookies

Beautify

Body

200 OK 6 ms 157 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

1

[]

Account is Gone!

Updating Account:

PUT

http://www.localhost:8080/rest-BN/api/accounts/62

Send

Params

Auth

Headers (8)

Body

Pre-req.

Tests

Settings

Cookies

raw

JSON

Beautify

```
1 {
2   "uid": "62",
3   "zip": "60616",
4   "name": "tabor",
5   "street": "123 first street",
6   "phone": "224-403-9894",
7   "isActive": false
}
```

- Changing zip and street

API response:

Body

204 No Content 22 ms 112 B

Save Response

Pretty

Raw

Preview

Visualize

Text

Copy

Search

```
1
```

Update:

Body

200 OK 23 ms 482 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

Copy

Search

```
1 {
2   "uid": "62",
3   "isActive": false,
4   "userAsks": [],
5   "userGives": [],
6   "userThanks": [],
7   "thanks_to_this_account": [],
8   "userNotes": [],
9   "name": "tabor",
10  "phone": "224-403-9894",
11  "picture": "",
12  "zip": "60616",
13  "street": "123 first street",
14  "dateCreated": "2023-03-16"
15 }
16
```

ASKS:

- This allows a user to send an ask request for a favor/gift
- Consists of a uid (user id) of the user sending the ask, type (type of the message), and the description of the ask

Create Ask:

POST

http://www.localhost:8080/rest-BN/api/accounts/62/asks

Send

Params

Auth

Headers (8)

Body

Pre-req.

Tests

Settings

Cookies

raw

JSON

Beautify

```
6  {
7    "uid": "62",
8    "type": "ask",
9    "description": "Can you send me a couple gifts?"
10 }
11
12
```

Getting User Asks:

- aid is a unique ask id for every ask
- Start date is when the ask request is active (default is current date) and date created is when the ask was constructed

GET	▼	http://www.localhost:8080/rest-BN/api/asks	Send ▼
Params	Auth	Headers (8)	Body ●
Pre-req. Tests Settings			Cookies
raw ▼	JSON ▼	Beautify	

```
[
  {
    "uid": "62",
    "aid": "6ed7b1b1-171c-4775-b704-3b279a6ed751",
    "type": "ask",
    "description": "Can you send me a couple gifts?",
    "startDate": "2023-03-16",
    "endDate": "",
    "extra_zip": [],
    "is_active": true,
    "date_created": "2023-03-16"
  },
]
```

Getting All Asks in Application:



GET ▼ http://www.localhost:8080/rest-BN/api/asks Send ▼

Params Auth Headers (8) Body ● Pre-req. Tests Settings Cookies

Body ▼ 200 OK 7 ms 1.51 KB Save Response ▼

```
35 {
36   "uid": "62",
37   "aid": "f1a1b55d-100e-4ac6-b231-8593edc6b62d",
38   "type": "ask",
39   "description": "Can you send me a couple gifts?",
40   "startDate": "2023-03-16",
41   "endDate": "",
42   "extra_zip": [],
43   "is_active": true,
44   "date_created": "2023-03-16"
45 },
46 {
47   "uid": "80",
48   "aid": "3a0d4da6-c4ad-4cee-a494-60009cc59c25",
49   "type": "ask",
50   "description": "Can you send me a couple gifts?",
51   "startDate": "2023-03-16",
52   "endDate": "",
53   "extra_zip": [],
54   "is_active": true,
55   "date_created": "2023-03-16"
56 }
```

- Asks are now stored in the user accounts information (“userAsks”)!



```
{
  "uid": "80",
  "isActive": false,
  "userAsks": [
    {
      "uid": "80",
      "aid": "3a0d4da6-c4ad-4cee-a494-60009cc59c25",
      "type": "ask",
      "description": "Can you send me a couple gifts?",
      "startDate": "2023-03-16",
      "endDate": "",
      "extra_zip": [],
      "is_active": true,
      "date_created": "2023-03-16"
    }
  ]
}
```

For more functions and information check out the complete API:

<http://www.cs.iit.edu/~virgil/cs445/mail.spring2022/project/project-api.html>