

# Testing document for TODO APP

*Author: Srikanth Amudala*

## Table of Contents

<b>Section 1: User registration and Testing .....</b>	<b>2</b>
Step 1: Initial setup .....	2
Step 2: Auth tokens are generated using a registration API .....	2
Step 3: (Testing).....	2
<b>Section 2: Accessing Admin dashboard: .....</b>	<b>3</b>
Step 1: creating superuser .....	3
Admin Dashboard .....	4
Auth tokens: .....	5
Task details.....	6
<b>Section 3: Verifying all API's through postman .....</b>	<b>6</b>
Add TODO: .....	6
List all TODO's.....	7
Applying filters to list TODO: .....	8
Filtering through multiple fields .....	9
Update TODO: .....	10
Delete TODO:.....	10

## Section 1: User registration and Testing

### Step 1: Initial setup

Open terminal in the root folder and execute the following commands.

```
docker-compose run web python manage.py migrate
```

```
docker-compose up --build
```

### Step 2: Auth tokens are generated using a registration API

Register a user and get the token from the response as below

#### ***User Registration***

url: `http://localhost:8000/account/register`

body: username, email, password, password2

\*password2, password should match each other

Returns the auth token in the json format as shown below

The screenshot shows a REST client interface with a POST request to `http://localhost:8000/account/register`. The request body is in x-www-form-urlencoded format with the following data:

KEY	VALUE	DESCRIPTION
username	srikanth	
email	srikanth@gmail.com	
password2	srikanth	
password	srikanth	

The response status is 200 OK. The response body is in JSON format:

```
{  "email": "srikanth@gmail.com",  "token": "cd00d9dad0aa06655a742681a3cfff7a1c760ae5",  "response": "successfully registered new user.",  "username": "srikanth",  "pk": 2}
```

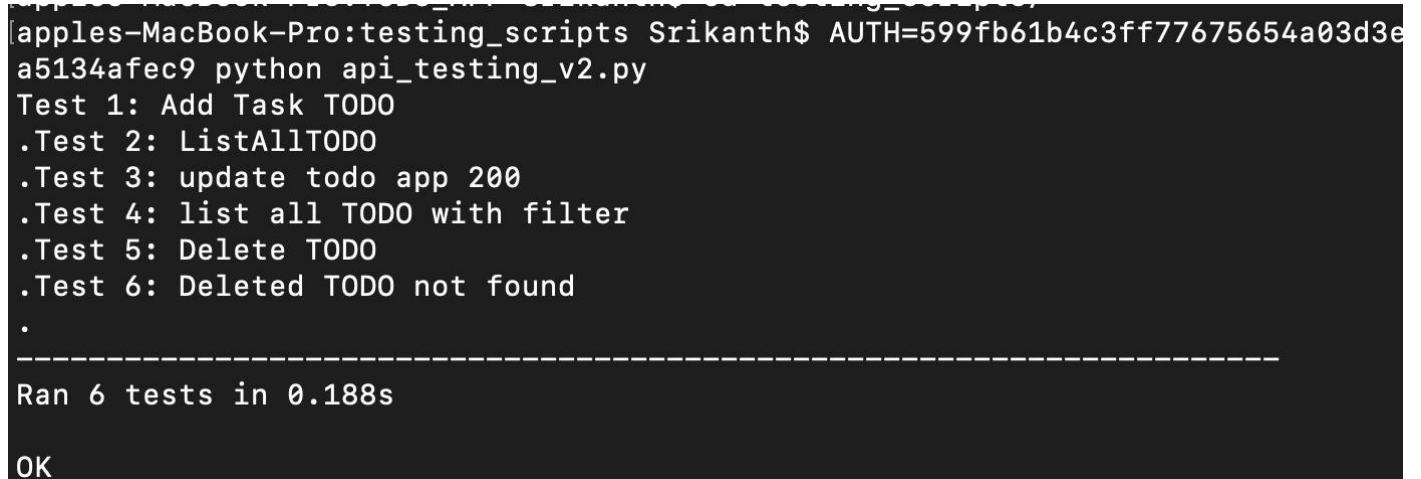
### Step 3: (Testing)

Execute the following command for testing in the terminal

CD -> TODO\_APP/testing\_scripts

AUTH=<token> python3 api\_testing\_v2.py

Following screen shot shows the tests performed



```
apples-MacBook-Pro:testing_scripts Srikanth$ AUTH=599fb61b4c3ff77675654a03d3e
a5134afec9 python api_testing_v2.py
Test 1: Add Task TODO
.Test 2: ListAllTODO
.Test 3: update todo app 200
.Test 4: list all TODO with filter
.Test 5: Delete TODO
.Test 6: Deleted TODO not found
.
-----
Ran 6 tests in 0.188s

OK
```

Test 1: Adds a task to TODO List for the specific user using the token provided.

Test 2: Lists all the TODO tasks with filter sortby: title

Test 3: Updates the above added Task description and status

Test 4: Lists all TODO with filter using title

Test 5: Deletes the above added task

Test 6: Verifies if the above deleted Task is present in listallTODO

## Section 2: Accessing Admin dashboard:

### Step 1: creating superuser

To access the admin dashboard, open terminal and execute the following command

*`docker-compose run web python manage.py createsuperuser`*

provide the email id and password.

```

/00020/0121kanth/Desktop/11002100/assignment/1000_HM1_2/1000_HM1
apples-MacBook-Pro:TODO_APP Srikanth$ docker-compose run web python manage.py createsuperuser
Starting todo_app2_db_1 ... done
Email: root@root.com
Password:
Password (again):
The password is too similar to the email.
This password is too short. It must contain at least 8 characters.
This password is too common.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
apples-MacBook-Pro:TODO_APP Srikanth$

```

We can now access the admin dashboard using the above created super user credentials.

Admin Dashboard <http://localhost/admin/>

TODO Server APP | Admin Dashboard

WELCOME, **ROOT@ROOT.COM**. [VIEW](#)

Site administration

ACCOUNT

Accounts + Add Change

AUTH TOKEN

Tokens + Add Change

SAMPLEAPP

Tasks + Add Change

Recent actions

My actions

Task 2

Task

Task 1

Task

Task 2

Task

Task 6

Task

Task 9

Task

Task 10

Task

test1

Task

Users are registered through registration API, which automatically generations auth token and returns to the user as explained earlier.

Select account to change

ADD ACCOUNT +

Action: -----  0 of 9 selected

<input type="checkbox"/>	PK	EMAIL	USERNAME	DATE JOINED	LAST LOGIN	IS ADMIN	IS STAFF
<input type="checkbox"/>	3	prem@gmail.com	prem	July 2, 2020, 7:53 a.m.	July 2, 2020, 7:53 a.m.	✖	✖
<input type="checkbox"/>	1	root@root.com	root	July 2, 2020, 6:55 a.m.	July 3, 2020, 1:32 a.m.	✔	✔
<input type="checkbox"/>	2	srikanth@gmail.com	srikanth	July 2, 2020, 7:07 a.m.	July 2, 2020, 7:07 a.m.	✖	✖
<input type="checkbox"/>	4	sunny@gmail.com	sunny	July 2, 2020, 9:07 a.m.	July 2, 2020, 9:07 a.m.	✖	✖
<input type="checkbox"/>	5	user2@gmail.com	user2	July 2, 2020, 10:26 p.m.	July 2, 2020, 10:26 p.m.	✖	✖
<input type="checkbox"/>	6	user3@gmail.com	user3	July 2, 2020, 10:29 p.m.	July 2, 2020, 10:29 p.m.	✖	✖
<input type="checkbox"/>	7	user4@gmail.com	user4	July 2, 2020, 10:32 p.m.	July 2, 2020, 10:32 p.m.	✖	✖
<input type="checkbox"/>	8	user8@gmail.com	user8	July 3, 2020, 12:31 a.m.	July 3, 2020, 12:31 a.m.	✖	✖
<input type="checkbox"/>	9	user9@gmail.com	user9	July 3, 2020, 1:11 a.m.	July 3, 2020, 1:11 a.m.	✖	✖

9 accounts

Auth tokens:

## TODO Server APP | Admin Dashboard

WELCOME, **ROOT@ROOT.COM** [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Home > Auth Token > Tokens

Select Token to change

ADD TOKEN +

Action: -----  0 of 9 selected

<input type="checkbox"/>	KEY	USER	CREATED
<input type="checkbox"/>	843779e3cb34cc1374a73aa675f5fafd1468811c	user9@gmail.com	July 3, 2020, 1:11 a.m.
<input type="checkbox"/>	24752390a93e37ad2046c1b9b5229ae749e51716	user8@gmail.com	July 3, 2020, 12:31 a.m.
<input type="checkbox"/>	276ad2d64306663bf97a010c26db7ce5652a0335	user4@gmail.com	July 2, 2020, 10:32 p.m.
<input type="checkbox"/>	694a180f250d04f7725486ece3682050c6ad901e	user3@gmail.com	July 2, 2020, 10:29 p.m.
<input type="checkbox"/>	23d638bb9bb16d60cf3a8410cf8101a4ad3a4282	user2@gmail.com	July 2, 2020, 10:26 p.m.
<input type="checkbox"/>	dd974e3e09cf590136638230b308b3ae30a1744	sunny@gmail.com	July 2, 2020, 9:07 a.m.
<input type="checkbox"/>	9a772d4eb14fc1a2c8daaf51d0d02411a49c8613	prem@gmail.com	July 2, 2020, 7:53 a.m.
<input type="checkbox"/>	cd00d9dad0aa06655a742681a3cfff7a1c760ae5	srikanth@gmail.com	July 2, 2020, 7:07 a.m.
<input type="checkbox"/>	738e8434e2c8ab3b29c3477d047c08adc235d6ad	root@root.com	July 2, 2020, 6:55 a.m.

9 Tokens

All tasks added by user, can be filtered from the side bar according to the status

TODO Server APP | Admin Dashboard

WELCOME, **ROOT@ROOT.COM** / VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Sampleapp > Tasks

Select task to change

Action: -----

0 of 7 selected

<input type="checkbox"/>	TITLE	STATUS
<input type="checkbox"/>	Task 7	Done
<input type="checkbox"/>	Task 4	In Progress
<input type="checkbox"/>	Task 2	TODO
<input type="checkbox"/>	Task 6	DONE
<input type="checkbox"/>	Task 3	TODO
<input type="checkbox"/>	Task 1	TODO
<input type="checkbox"/>	test1	TODO

7 tasks

FILTER

By status

All

DONE

Done

In Progress

TODO

## Task details in read only mode

TODO Server APP | Admin Dashboard


WELCOME, **ROOT@ROOT.COM** / VIEW SITE / CHANGE PASSWORD / LOG OUT


Home > Sampleapp > Tasks > Task 7

Change task

UserID:

user2@gmail.com





Title:

Task 7

Status:

Done

TaskDueDate:

April 23, 2019

Desc:

By user 2

## Section 3: Verifying all API's through postman

\* for now all tasks in the db should be unique

Add TODO:

URL : <http://localhost:8000/todo/addTODO/>

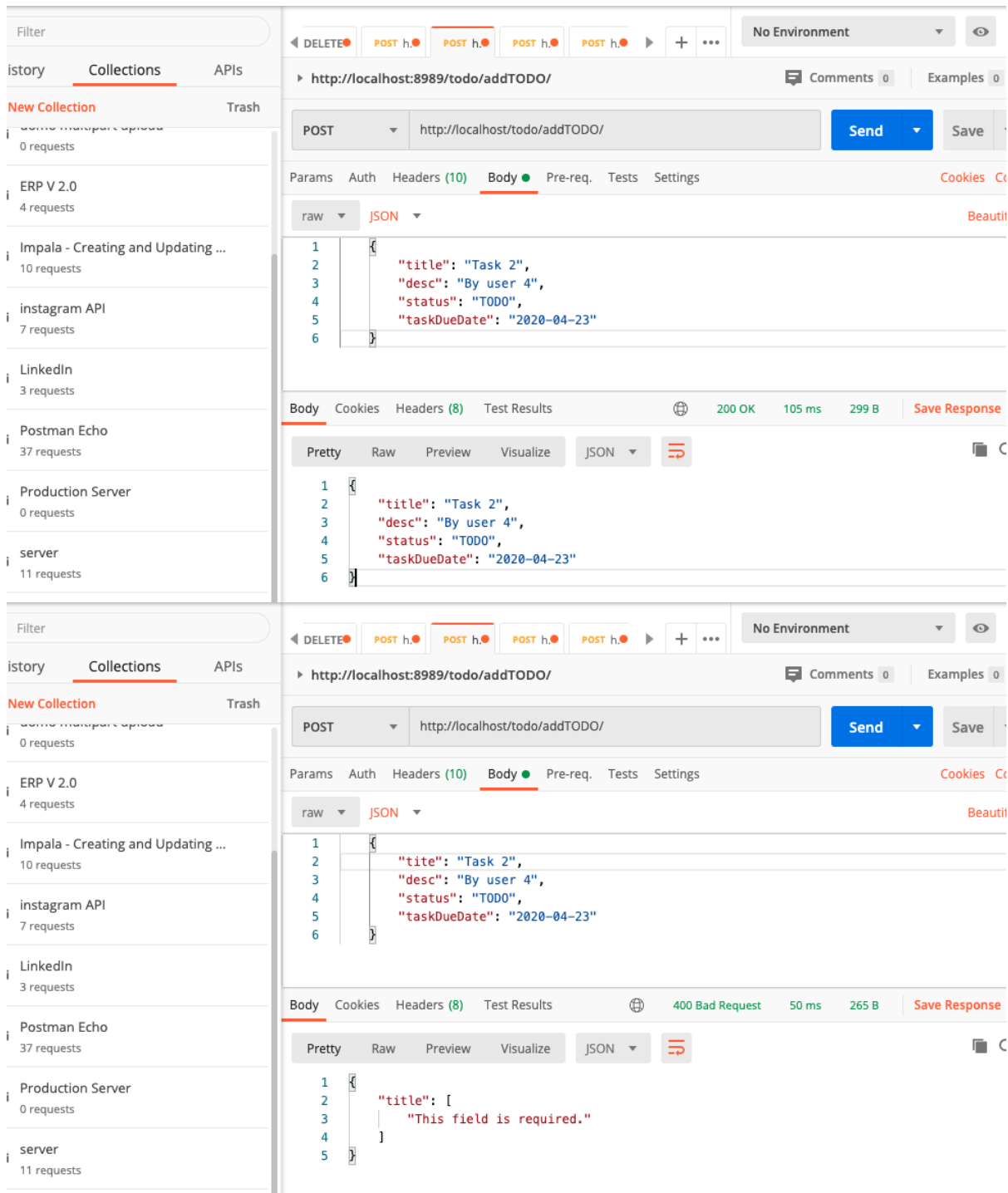
While making the call, please pass Headers filed while requesting the url as below  
 Authorization Token <tokenValue>

Returns: added task

Error handling:

If a field is missed, returns the field title stating, field is required.

Following screen shot shows the output from Registered users



## Tasks added

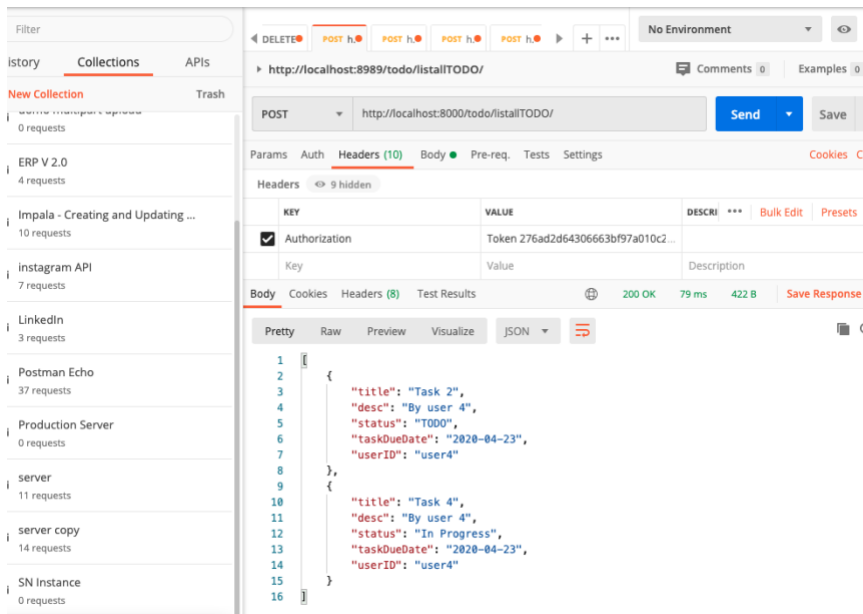
1. `user4`: Task 4, Task 2
2. `user2`: Task 5, Task 7

## List all TODO's

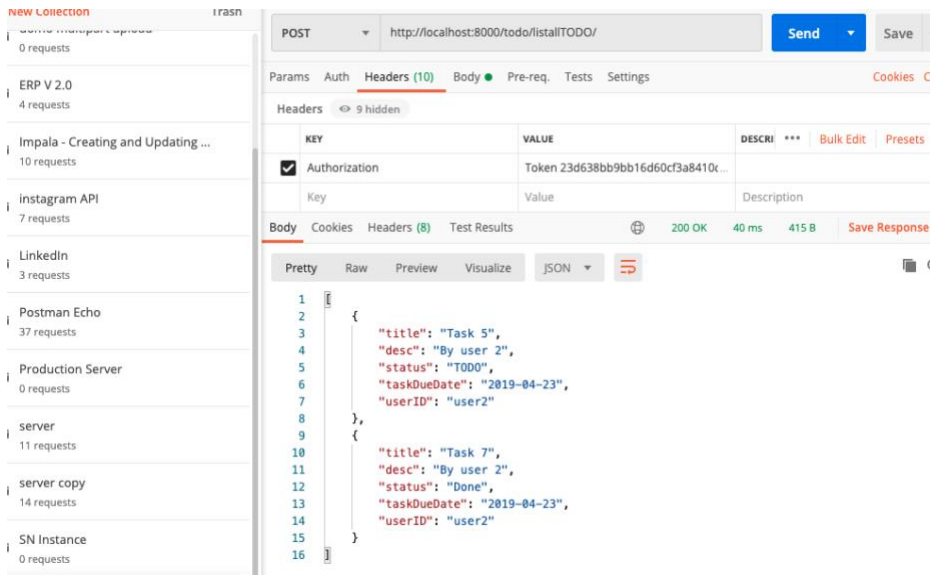
URL: `http://localhost:8000/todo/listallTODO/`

returns the tasks of the specific user using the access token

Following screenshot shows the result, when `user4` token has been used in the auth header.



Result, when inputted user2 token



Applying filters to list TODO:

In the body, input the fields you want to filter as form data.

Example filter with title: Task 5 for user2

\*filters the data only from the specific user, given an unknown task name, returns empty list.



POST http://localhost:8000/todo/listallTODO/ Send Save

Params Auth Headers (10) **Body** Pre-req. Tests Settings Cookies C

form-data

	KEY	VALUE	DESCRIPTION	*** Bulk E
<input type="checkbox"/>	userID	prem@gmail.com		
<input checked="" type="checkbox"/>	title	Task 5		
<input type="checkbox"/>	status	TODO		
<input type="checkbox"/>	taskDueDate	2020-04-23		
<input type="checkbox"/>	sortby	title		
	Key	Value	Description	

**Body** Cookies Headers (8) Test Results 200 OK 49 ms 316 B Save Response

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "title": "Task 5",
4     "desc": "By user 2",
5     "status": "TODO",
6     "taskDueDate": "2019-04-23",
7     "userID": "user2"
8   }
9 ]
```

### Filtering through multiple fields

For example, user2 has two tasks with status TODO and DONE with the same taskDueDate. Lets find the task with TODO on that specific date.

Following image shows the result

POST http://localhost:8000/todo/listallTODO/ Send Save

Params Auth Headers (10) Body Pre-req. Tests Settings Cookies

form-data

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input type="checkbox"/>	userID	prem@gmail.com			
<input type="checkbox"/>	title	Task 8			
<input checked="" type="checkbox"/>	status	TODO			
<input checked="" type="checkbox"/>	taskDueDate	2019-04-23			
<input type="checkbox"/>	sortBy	title			
	Key	Value	Description		

Body Cookies Headers (8) Test Results 200 OK 58 ms 316 B Save Response

Pretty Raw Preview Visualize JSON

```

1 [
2   {
3     "title": "Task 5",
4     "desc": "By user 2",
5     "status": "TODO",
6     "taskDueDate": "2019-04-23",
7     "userID": "user2"
8   }
9 ]

```

## Update TODO:

Let's update the status of Task 5 to DONE for user2 and make some desc updates

POST http://localhost:8000/todo/updateTODO/ Send Save

Params Auth Headers (10) Body Pre-req. Tests Settings Cookies

raw JSON

```

1 {
2   "title": "Task 5",
3   "desc": "update user 2 data",
4   "status": "DONE",
5   "taskDueDate": "2019-04-23"
6 }

```

Body Cookies Headers (8) Test Results 200 OK 76 ms 306 B Save Response

Pretty Raw Preview Visualize JSON

```

1 {
2   "title": "Task 5",
3   "desc": "update user 2 data",
4   "status": "DONE",
5   "taskDueDate": "2019-04-23"
6 }

```

\*For time being we have to pass all the key fields while updating the data.

\*If the user tries to update a task created by another user, the server returns unauthorized error.

## Delete TODO:

URL: <http://localhost:8000/todo/deleteTODO/>

Takes title as input body field

Let us try to delete Task 5 from user2, returns a Boolean True, if successful else error reason

POST

http://localhost:8000/todo/deleteTODO/

Send

Save

Params

Auth

Headers (10)

Body

Pre-req.

Tests

Settings

Cookies

Co

form-data

	KEY	VALUE	DESCRIPTION	...	Bulk E
<input checked="" type="checkbox"/>	title	Task 5			
	Key	Value	Description		

Body

Cookies

Headers (8)

Test Results

200 OK

88 ms

220 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

1

true

\*If the user tries to delete a task created by another user, the server returns unauthorized error.