



# my collectn

## testing

### GET /store

http://127.0.0.1:5000/store

### BODY formdata

Example Request

/store

```
curl --location --request GET 'http://127.0.0.1:5000/store'
```

### GET /store/<name>

http://127.0.0.1:5000/store/Online Shopping Store

Documentation Settings

## BODY formdata

Example Request

/store/<name>

```
curl --location --request GET 'http://127.0.0.1:5000/store/Online Shopping Store'
```

## POST /store

http://127.0.0.1:5000/store

## HEADERS

### Content-Type

application/json

## BODY raw

```
{  
  "name" : "Another Store"  
}
```

Example Request

/store



```
curl --location --request POST 'http://127.0.0.1:5000/store' \  
--header 'Content-Type: application/json' \  
--data-raw '{  
  "name" : "Another Store"  
}'
```

## POST /store/<name>/item

http://127.0.0.1:5000/store/Online Shopping Store/item

### HEADERS

#### Content-Type

application/json

### BODY raw

```
{  
  "name" : "Another Store",  
  "price": 10.87  
}
```

Example Request

/store/<name>/item



```
curl --location --request POST 'http://127.0.0.1:5000/store/Online Shopping Store/item' \  
--header 'Content-Type: application/json' \  
--data-raw '{  
  "name" : "Another Store",  
  "price": 10.87  
'
```

# test\_first api design

## GET /items

http://127.0.0.1:5000/items

this should return a list of items, each in json format.

## BODY formdata

Example Request

/items

```
curl --location --request GET 'http://127.0.0.1:5000/items'
```



## GET /item/<name>

http://127.0.0.1:5000/item/ball

this should return one specific item, uniquely identified by its name.No 2 items may have the same name.

## HEADERS

### Authorization

JWT

eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJleHAiOiJlOTk5OTQsImIhdCI6MTYzMjk5MDc5NCwibmJmljoxNjMyOTkwNzk0LCJpZGVudGl0eSI6MX0.WnAWNjliOewDWz6gbwvGP5TgQXnh16T4205irHj8y2M

### BODY formdata

Example Request

/item/<name>

```
curl --location --request GET 'http://127.0.0.1:5000/item/piano' \  
--header 'Authorization: JWT eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJleHAiOiJlOTk5OTQsImIhdCI6MTYzMjk5MDc5NCwibmJmljoxNjMyOTkwNzk0LCJpZGVudGl0eSI6MX0.WnAWNjliOewDWz6gbwvGP5TgQXnh16T4205irHj8y2M'
```

Example Response

500 INTERNAL SERVER ERROR

Body    Header (5)



```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>TypeError: wrapper() got an unexpected keyword argument 'name' // Werkzeug Debugger</title>
    <link rel="stylesheet" href="?__debugger__=yes&cmd=resource&f=style.css"
        type="text/css">
    <!-- We need to make sure this has a favicon so that the debugger does
        not by accident trigger a request to /favicon.ico which might
        change the application state. -->
```

[View More](#)

## POST /item/<name>

```
http://127.0.0.1:5000/item/ball
```

this should create a new item. If the item already exists, it will fail.

## HEADERS

### Content-Type

application/json

## BODY raw

```
{
  "name" : "piano",
  "price" : 15.88
}
```



## Example Request

/item/&lt;name&gt;

```
curl --location --request POST 'http://127.0.0.1:5000/item/ball' \  
--header 'Content-Type: application/json' \  
--data-raw '{  
  "name" : "piano",  
  "price" : 15.88  
'
```

**PUT /item/<name>**

this should create a new item or update an existing item.

**HEADERS****Content-Type**

application/json

**BODY** raw

```
{  
  "price" : 167.89  
}
```

 Example Request/item/<name>  
Documentation Settings ▼

```
curl --location --request PUT 'http://127.0.0.1:5000/item/sofa' \  
--header 'Content-Type: application/json' \  
--data-raw '{  
  "price" : 167.89  
'
```

## DEL /item/<name>

this will delete an item, uniquely identified by its name.

## HEADERS

### Content-Type

application/json

Example Request

/item/&lt;name&gt;

```
curl --location --request DELETE 'http://127.0.0.1:5000/item/ukulele' \  
--header 'Content-Type: application/json' \  
--data-raw ''
```



## POST /auth



Documentation Settings ▼

```
http://127.0.0.1:5000/auth
```

### HEADERS

#### Content-Type

```
application/json
```

### BODY raw

```
{
  "username" : "Sukanya",
  "password" : "bert"
}
```

Example Request

/auth

```
curl --location --request POST 'http://127.0.0.1:5000/auth' \
--header 'Content-Type: application/json' \
--data-raw '{
  "username" : "Sukanya",
  "password" : "bert"
}'
```