

CSC309 Group 431

The APIs are categorized by the apps in Django:

- accounts
- recipes
- userdata
- userfunction

accounts

Accounts APIs are responsible for login, signup, update user profiles.

POST Sign Up

```
http://127.0.0.1:8000/accounts/signup/
```

Sign Up requires the user to enter a unique username and email with passwords of at least 8 characters.

The response will return a JSON object with filed username and email.

Note:

- Password will not be returned.
- When signing up, users cannot add their phone and avatar.

Example Payload:

Plain Text

```
payload={'username': 'test',  
'email': 'test@gmail.com',  
'password': 'qwertyuiop32392',  
'password2': 'qwertyuiop32392'}
```

Body formdata

username	test
	Username must be unique
email	test@gmail.com
	Email must be unique

password	qwertyuiop32392
	Password must be at least 8 characters
password2	qwertyuiop32392
	Password must be at least 8 characters

POST Log In

http://127.0.0.1:8000/accounts/login/

The response will return user's information except for password.

Example Payload:

Plain Text

```
payload={ 'username': 'test',  
  'password': 'qwertyuiop32392' }
```

Body formdata

username	test
password	qwertyuiop32392

POST Log Out



http://127.0.0.1:8000/accounts/logout/

Log Out does not take any user inputs but it will ask the user to be logged in first in order to log out.

AUTHORIZATION Bearer Token

Token	{{token}}
-------	-----------

PUT Edit Profile



http://127.0.0.1:8000/accounts/profile/edit/

Edit Profile allows users to add phone numbers and avatars.

Note: When signing up, users cannot add their phone and avatar.

Example Payload:

Plain Text

```
payload={
  'username': 'test',
  'email': 'test@gmail.com',
  'password': 'ehwrqjklhejklqw',
  'password2': 'ehwrqjklhejklqw',
  'phone': '12345678'
}
files=[
  ('avatar', ('01_A_1.jpg', open('/Users/jingwenshi/Downloads/asl_data/test/A/01_A_1.jpg', 'rb')
)]
```

AUTHORIZATION Bearer Token

Token {{token}}

Body formdata

username	test
	Username must be unique
email	test@gmail.com
	Email must be unique
password	ehwrqjklhejklqw
	Password must be at least 8 characters long
password2	ehwrqjklhejklqw
	Password must be at least 8 characters long
phone	12345678
	This field is not required
avatar	This field is not required

GET Get Profile

🔒

http://127.0.0.1:8000/accounts/profile/details/

Returns a JSON response with all fields of the current user (except for password).

AUTHORIZATION Bearer Token

Token {{token}}

recipes

recipe APIs contains all recipe related methods:

- Create, Update, Delete Recipe/RecipeIngredient/Step
- Create Ingredient
- Get Recipe

Note:

- Ingredient and RecipeIngredient are two different models. Ingredient is a model that contains unique ingredient names. And Recipe Ingredient is a model with 2 FK pointed to Recipe and Ingredient and 2 attributes of amount and unit.
- Step is a model contains a description of the current step and a FK pointed the Recipe model.
- For more details, please refer to the Model descriptions.

POST Create Recipe

🔒

http://127.0.0.1:8000/recipes/create-recipe/

Example Payload:

Plain Text

```
payload={'title': 'How to make Soup Bao',
'description': 'A delicious dish!',
'time': '2',
'time_unit': 'hours',
'cuisine': 'Chinese',
'diet': 'Others'}
files=[
    ('picture',('01_A_1.jpg',open('/Users/jingwenshi/Downloads/asl_data/test/A/01_A_1.jpg','rt
    ]
```

AUTHORIZATION Bearer Token

Token

{{token}}

Body formdata

title	How to make Soup Bao This field is required
description	A delicious dish! This field is required
picture	This field is required

time	2
	This field is required
time_unit	hours
	This field is required
cuisine	Chinese
	This field is required
diet	Others
	This field is required

PUT Update Recipe



http://127.0.0.1:8000/recipes/recipe=1/RUD-recipe/

A recipe id must be passed to the endpoint url (after recipe=).

The payload is the same as Create Recipe

AUTHORIZATION Bearer Token

Token	{{token}}
-------	-----------

Body formdata

title	How to make Soup Dumplings
	This field is required
description	This is a traditional chinese dish!
	This field is required
picture	This field is required
time	3
	This field is required
time_unit	hours
	This field is required
cuisine	Chinese
	This field is required
diet	Others
	This field is required

DELETE Delete Recipe



```
http://127.0.0.1:8000/recipes/recipe=1/RUD-recipe/
```

Delete a specific recipe from the current user.

A recipe id must be passed to the endpoint url (after recipe=).

AUTHORIZATION Bearer Token

Token	{{token}}
-------	-----------

POST Create Ingredient



```
http://127.0.0.1:8000/recipes/recipe=1/create-ingredient/
```

Note: Once an ingredient is created, it cannot be deleted. But a RecipeIngredient can be deleted from a user's recipe.

Example Payload:

Plain Text

```
payload={ 'name': 'Pork' }
```

AUTHORIZATION Bearer Token

Token	{{token}}
-------	-----------

Body formdata

name	Pork
	This field is required

POST Create Recipe Ingredient



```
http://127.0.0.1:8000/recipes/recipe=1&ingredient=1/create-recipe-ingredient/
```

A recipe id and ingredient id must be passed to the endpoint url after the equal sign.

Note: The name of the ingredient should be passed in as an id in the endpoint. For more details, please refer to the model descriptions.

Example Payload:

Plain Text

```
payload={ 'amount': '500',  
'unit': 'grams' }
```

AUTHORIZATION Bearer Token

Token {{token}}

Body formdata

amount	500
	This field is required
unit	grams
	This field is required

PUT Update Recipe Ingredient



```
http://127.0.0.1:8000/recipes/recipe=1&ingredient=1/RUD-recipe-ingredient/
```

A recipe id and ingredient id must be passed to the endpoint url after the equal sign.

The payload is the same as Create Recipe Ingredient.

AUTHORIZATION Bearer Token

Token {{token}}

Body formdata

amount	400
	This field is required
unit	grams
	This field is required

DELETE Delete Recipe Ingredient



```
http://127.0.0.1:8000/recipes/recipe=1&ingredient=1/RUD-recipe-ingredient/
```

A recipe id and ingredient id must be passed to the endpoint url after the equal sign

A recipe id and ingredient id must be passed to the endpoint url after the equal sign.

Delete a RecipeIngredient from the current user's specific recipe.

AUTHORIZATION Bearer Token

Token	{{token}}
-------	-----------

POST Create Step



http://127.0.0.1:8000/recipes/recipe=1/create-step/

A recipe id must be passed to the endpoint url after the equal sign.

Note:

The number represents the step number and it is not unique. That is, a recipe can have multiple step 2. In the front end, the developer only simply displays the step in non-decreasing order and is the user's responsibility to organize their steps.

Example Payload:

Plain Text

```
payload={'number': '1',
'description': 'Mix pork with soy sauce'}
files=[
  ('picture',('02_A_1.jpg',open('/Users/jingwenshi/Downloads/asl_data/test/A/02_A_1.jpg','rt
  ]
```

AUTHORIZATION Bearer Token

Token	{{token}}
-------	-----------

Body formdata

number	1
	This field is required
description	Mix pork with soy sauce
	This field is required
picture	This field is required

PUT Update Step



http://127.0.0.1:8000/recipes/recipe=1&step=1/RUD-step/

`http://127.0.0.1:8000/recipes/recipe=1/step=1/RUD-step/`

A recipe id and step id must be passed to the endpoint url after the equeal sign.

The payload is the same as Create Step.

AUTHORIZATION Bearer Token

Token {{token}}

Body formdata

number	1
description	Mix pork with salt for 3 mins
picture	

DELETE Delete Step



`http://127.0.0.1:8000/recipes/recipe=1&step=1/RUD-step/`

A recipe id and step id must be passed to the endpoint url after the equeal sign.

AUTHORIZATION Bearer Token

Token {{token}}

GET View/Get Recipe

`http://127.0.0.1:8000/recipes/recipe=1/details/`

A recipe id must be passed to the endpoint url after the equeal sign.

userdata

userdata APIs contains all method to manipulate and modify user's data

POST Create Rating



`http://127.0.0.1:8000/userdata/recipe=1/create-rating/`

A recipe id must be passed to the endpoint url after the equeal sign.

Example Payload:

Plain Text

```
payload={'score': '5'}
```

AUTHORIZATION Bearer Token

Token {{token}}

Body formdata

score 5

This field is required

PUT Edit Rating

🔒

```
http://127.0.0.1:8000/userdata/rating=1/RU-rating/
```

A recipe id must be passed to the endpoint url after the equeal sign.

The payload is the same as Create Rating.

AUTHORIZATION Bearer Token

Token {{token}}

Body formdata

score 4

POST Create Comment

🔒

```
http://127.0.0.1:8000/userdata/recipe=1/create-comment/
```

A recipe id must be passed to the endpoint url after the equeal sign.

AUTHORIZATION Bearer Token

Token {{token}}

Body formdata

message	This recipe is pretty good!
	This field is required
file	This field is required

PUT Edit Comment



```
http://127.0.0.1:8000/userdata/recipe=1&comment=1/RUD-comment/
```

AUTHORIZATION Bearer Token

Token	{{token}}
-------	-----------

Body formdata

message	This recipe is good, but each step could be more clear.
file	

DELETE Delete Comment



```
http://127.0.0.1:8000/userdata/recipe=1&comment=1/RUD-comment/
```

AUTHORIZATION Bearer Token

Token	{{token}}
-------	-----------

POST Like Recipe



```
http://127.0.0.1:8000/userdata/recipe=1/create-liked-recipe/
```

AUTHORIZATION Bearer Token

Token	{{token}}
-------	-----------

DELETE Delete Like Recipe



```
http://127.0.0.1:8000/userdata/liked-recipe=1/RD-liked-recipe/
```

AUTHORIZATION Bearer Token

Token {{token}}

POST Create Browsed Recipe



http://127.0.0.1:8000/userdata/recipe=1/create-browsed-recipe/

AUTHORIZATION Bearer Token

Token {{token}}

DELETE Delete Browsed Recipe



http://127.0.0.1:8000/userdata/browsed-recipe=1/RD-browsed-recipe/

AUTHORIZATION Bearer Token

Token {{token}}

POST Create Shopping Lst



http://127.0.0.1:8000/userdata/recipe=1/create-shopping-lst/

AUTHORIZATION Bearer Token

Token {{token}}

DELETE Delete Shopping Lst



http://127.0.0.1:8000/userdata/shopping-lst=1/RD-shopping-lst/

AUTHORIZATION Bearer Token

Token {{token}}

POST Favorite Recipe



http://127.0.0.1:8000/userdata/recipe=1/create-liked-recipe/

http://127.0.0.1:8000/userdata/recipe=1/create-liked-recipe/

AUTHORIZATION Bearer Token

Token {{token}}

DELETE Delete Favorite Recipe



http://127.0.0.1:8000/userdata/liked-recipe=1/RD-liked-recipe/

AUTHORIZATION Bearer Token

Token {{token}}

User Function

GET Ingredient Autocomplete



http://127.0.0.1:8000/userfunction/autocomplete/?ingredient=Po

AUTHORIZATION Bearer Token

Token {{token}}

PARAMS

ingredient Po

GET Popular Recipes

http://127.0.0.1:8000/userfunction/popular-recipes/

GET My Recipe



http://127.0.0.1:8000/userfunction/my-recipe/

AUTHORIZATION Bearer Token

Token {{token}}

GET Search by Name

```
http://127.0.0.1:8000/userfunction/search/name/?name=Soup&cuisine=Chinese&diet=Others&time=3&unit=hours
```

PARAMS

name	Soup
cuisine	Chinese
diet	Others
time	3
unit	hours

GET Search by Creator

```
http://127.0.0.1:8000/userfunction/search/creator/?creator=test&cuisine=Chinese&diet=Others&time=3&unit=hours
```

PARAMS

creator	test
cuisine	Chinese
diet	Others
time	3
unit	hours

GET Search by Ingredient

```
http://127.0.0.1:8000/userfunction/search/ingredient/?ingredient=Pork&cuisine=Chinese&diet=Others&time=3&unit=hours
```

PARAMS

ingredient	Pork
------------	------

Signup

Endpoint: /accounts/signup/

Payload:

Methods: Post

Description: signs up and creates a user

Login

Endpoint: /accounts/login/

Payload:

Methods: Post

Description: logs the user in with token auth

View Profile

Endpoint: /accounts/profile/details/

Methods: Get

Description: displays the request user's information

Login required

Edit Profile

Endpoint: /accounts/profile/edit/

Methods: Put

Description: updates the request user's information

Login required

Create Recipe

Endpoint: recipes/create-recipe/

Payload:

Methods: Post

Description: creates a recipe under the request user

Login required

Update Recipe

Endpoint: recipes/recipe=<rid>/RUD-recipe/

Payload:

Methods: Put

Login required

Delete Recipe

Endpoint: recipes/recipe=<rid>/RUD-recipe/

Methods: Del

Description: delete a recipe

Login required

Create Ingredient

Endpoint: recipes/recipe=<rid>/create-ingredient/

Payload:

Methods: Post

Description: create a raw ingredient

Create Recipe Ingredient

Endpoint: recipes/recipe=<rid>&ingredient=<iid>/create-recipe-ingredient/

Payload:

Methods: Post

Description: create a recipe ingredient

Update Recipe Ingredient

Endpoint: recipes/recipe=<rid>&ingredient=<iid>/RUD-recipe-ingredient/

Payload:

Methods: Put

Description: update a recipe ingredient

Delete Recipe Ingredient

Endpoint: recipes/recipe=<rid>&ingredient=<iid>/RUD-recipe-ingredient/

Methods: Del

Description: delete a recipe ingredient

Create Step

Endpoint: recipes/recipe=<rid>/create-step

Payload:

Methods: Post

Description: create a step for a recipe

Update Step

Endpoint: recipes/recipe=<rid>&step=<sid>/RUD-step/

Payload:

Method: Put

Description: update a step for a recipe

Delete Step

Endpoint: recipes/recipe=<rid>&step=<sid>/RUD-step/

Method: Del

Description: delete a step for a recipe

View Recipe

Endpoint: recipes/recipe=<rid>/details/

Method: Get

Description: display details of a recipe

Create Rating

Endpoint: userdata/recipe=<rid>/create-rating

Method: Post

Description: create a rating by the user for a recipe

Delete Shopping List

Endpoint: userdata/recipe=<rid>/RD-rating

Method: Post

Description: removes a shopping list for the user

Create Comment

Endpoint: userdata/recipe=<rid>/create-comment

Method: Post

Description: create a comment by the user for a recipe

Liked Recipe

Endpoint: userdata/recipe=<rid>/create-liked-recipe

Method: Post

Description: allows the user to like a recipe

Favorite Recipe

Endpoint: userdata/recipe=<rid>/create-favorite-recipe

Method: Post

Description: allows the user to favorite a recipe

Create Shopping List

Endpoint: userdata/recipe=<rid>/create-shopping-list

Method: Post

Description: creates a shopping list for the user

Delete Shopping List

Endpoint: userdata/recipe=<rid>/RD-shopping-list

Method: Post

Description: removes a shopping list for the user

cuisine	Chinese
diet	Others
time	3
unit	hours

GET **Get Shopping List**



http://127.0.0.1:8000/userfunction/shopping-list/

AUTHORIZATION Bearer Token

Token {{token}}