

Part 1: Conceptual

Part 2: Whose Currency is it Anyway?

Tests

Submitting

Solution

Assessment: Flask 1

[Download code](#)

Warning: Assessments

Remember, assessments are meant to be completed **by you**, not as a shared exercise with friends or other members of your cohort.

All code submitted should be **written by you**. If you incorporate code from elsewhere, it must be clearly specified.

Use a private GitHub repo to submit this assignment. Add your mentor as a collaborator. Please do not put your assessment on public GitHub repo.

Part 1: Conceptual

Answer the following questions inside the ***conceptual.md*** file.

Part 2: Whose Currency is it Anyway?

Note: Don't use AJAX!

This entire application should be backend Python and Flask. Do not use any Javascript for this.

You're going to build a small currency converter! This will be done using the ***forex-python*** module which you can install in your virtual environment using `pip install forex-python`.

You can [read its documentation](#).

Make sure your application has the following:

- A form with three inputs:
 - One input to type in a three letter currency code to convert from
 - One input to type in a three letter currency code to convert to
 - Another input to type in a number amount to convert

This could look like this:

Converting from

USD

Converting to

GBP

Amount:

50

Convert

- Once the form is submitted, make sure that a valid currency code (eg, USD, EUR, JPY) has been input for the currency you are converting from and the currency you are converting to. If the currency code is invalid, you should display a friendly error message to the user letting them know that the currency code is invalid. If the amount is not a valid number, note that, too:

Not a valid amount.

Not a valid code: ZZZ

Not a valid code: YYY

Converting from

ZZZ

Converting to

YYY

Amount:

Convert

(**Hint:** To handle the error of an invalid currency code, it may help to check what type of error is raised in the ***forex-python*** module and handle it gracefully.)

- If the currency code is valid, you should display a message to the user letting them know the value in the currency they are trying to convert into. When you display the converted currency, it should have an accompanying currency symbol and be rounded to two decimal places. For example, if we were to convert 100 USD to EUR, we would expect a reuslt like “€ 85.16”

This could look like this:

The result is £ 39.51.

Home

- Refactor your code** to move any logic that's isn't about the route itself to functions in other Python file(s). Remember, *separate your concerns* — if you end up with a function that, say, checks if a currency code is valid, you might want this later in a non-Flask app, so move it out of ***app.py***.

Tests

Write tests for your Flask routes and any other functions you write.

It will take a bit of creative thinking on how to test this — after all, coversion rates between currencies change, so it wouldn't be a good idea to have a code that makes sure \$100 becomes €85.16. Hover below for a hint.

Hover to reveal

Submitting

Warning: Do not include your “venv” in the submitted ZIP file!

Delete the `venv` folder before you zip & upload your assessment. You will be unable to submit this assessment if you include your ***venv*** folder since it is so large (and it would be pointless, anyway—a venv folder won't work except on the computer where it was made)

Solution

You can view [Our Solution](#) (password required)