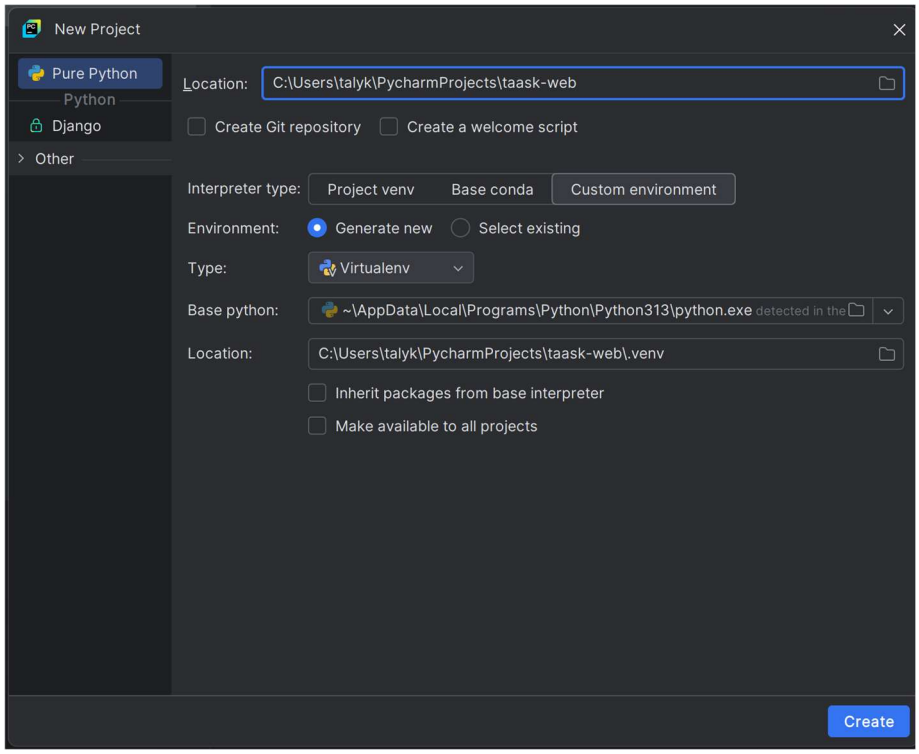
Оглавление

Разработка http сервиса

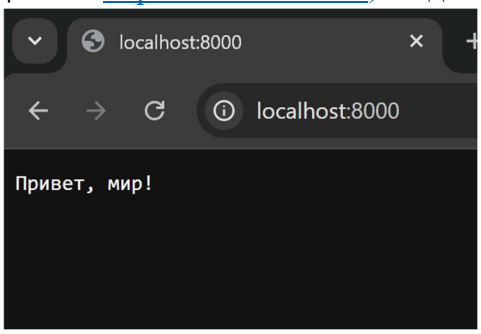
Сегодня вы будете командой разработчиков, которая создаёт web сервисы. Весь процесс максимально приближен к тому, как это происходит в современных IT компаниях. Вы получите требования к сервису (этот файл), напишите и протестируете свой код. Если будут вопросы – не стесняйтесь задавать.

Настройка проекта

1. Скачайте проект https://github.com/talykovnikita/quadratic-equation-web/archive/refs/heads/main.zip
2. Создайте новый проект в pycharm с инициализацией виртуального окружения (venv)



3. Добавьте в проект скаченные файлы из шага 1
4. В консоле выполните команду pip install -r requirements.txt
5. Запустите main.py
6. Откройте браузер и перейдите на http://localhost:8000, вы должны увидеть:



7. Если всё получилось, настройка прошла успешно. Если нет – поднимите руки и попросите помочь

Задание

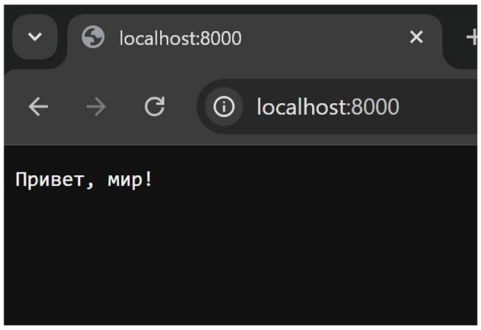
Вам предстоит разработать бизнес-логику веб-сервиса, который умеет решать квадратные уравнения и протестировать его работу. Часть кода уже написана, это вам поможет.

В проекте уже есть следующие файлы:

1. **main.py** - содержит код запуска http сервиса. Править его не требуется.

Для запуска сервиса - запускаем файл main.py и в браузере открываем http://localhost:8000

Вы должны увидеть такое сообщение:



Вы уже сделали это на этапе настройки проекта.

2. HttpHandler.py – содержит шаблон методов-обработчиков для разных типов запросов.

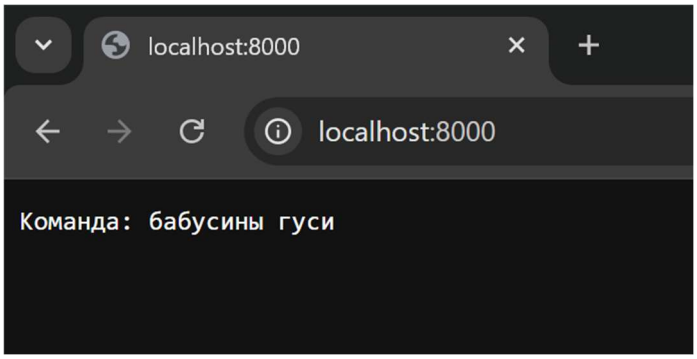
- do_GET() – для GET запросов.
- do_POST() – для POST запросов.

Нужно реализовать логику именно этих методов.

3. test_response.py – файл содержит тесты, которые помогут протестировать ваш код. Тесты написаны с помощью фреймворка pytest. Дальше есть инструкция по их запуску.

Требования к сервису

1. Придумать название команды и реализовать логику обработки GET запросов в методе do_GET, чтобы ваш сервер отвечал строкой «Команда: *название вашей команды*» и статус кодом 200.



2. Реализовать логику обработки POST запросов в методе do_POST, который на вход в теле запроса получает JSON с параметрами квадратного уравнения:

```
{
   "a": 1,
   "b": -5,
   "c": 4,
}
```

а на выходе отвечает JSON с корнями этого уравнения

```
{
   "x1": 4,
   "x2": 1
}
```

Важно:

- Следите за форматом выходных данных, в реальных системах это очень важно.
- Дробные числа округлять до 2-ух знаков после запятой.
- Целые числа отправлять без дробной части.
- В ответе в x1 записывать тот корень, который получается при **прибавлении корня из D.**

Требования к ответам сервиса

Когда отвечать	Статус код	Формат ответа

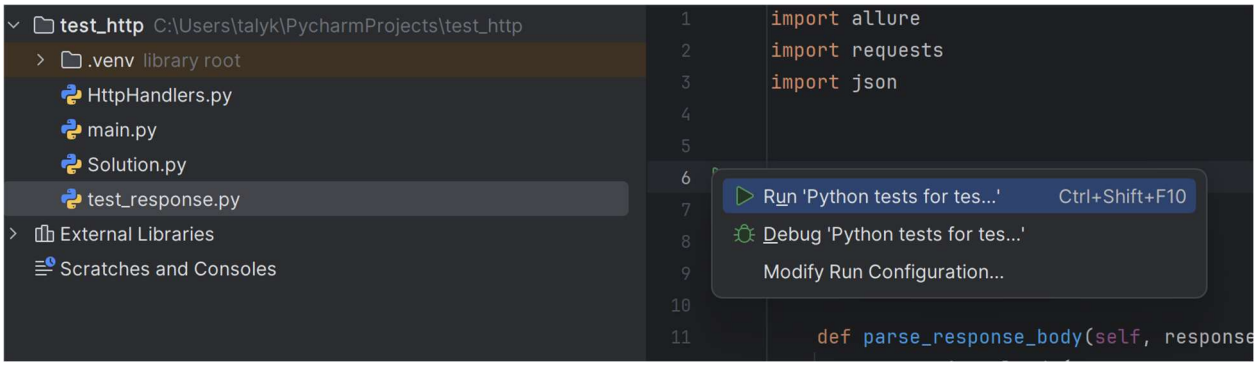
Удалось решить уравнение, независимо от количества корней	200	{
		"x1": 4,
		"x2": 1
		}

D < 0	403	{
		“error”: “уравнение не имеет решений, дискриминант меньше нуля”
		}

В теле запроса отсутствует один из параметров a, b или c	422	{ “error”: "переданы не все параметры" }
Любые другие ошибки	500	{ “error”: "произошла ошибка на стороне сервиса" }

Тестирование

Чтобы запустить проверку, нужно в файле test_response.py нажать на “Play” возле класса с тестами и выбрать опцию “Run …”.



Когда ваш сервис будет готов, можете запускать тесты. Сначала запустите файл main.py, чтобы запустить сервис. Потом запускайте тесты.

Успешным успехом считается, когда все тесты зелёные.