Topic: 7. Class-Based Views Basics & Retrieving All Records

Speaker: Personal | Notebook: API Development using Django Framework

---

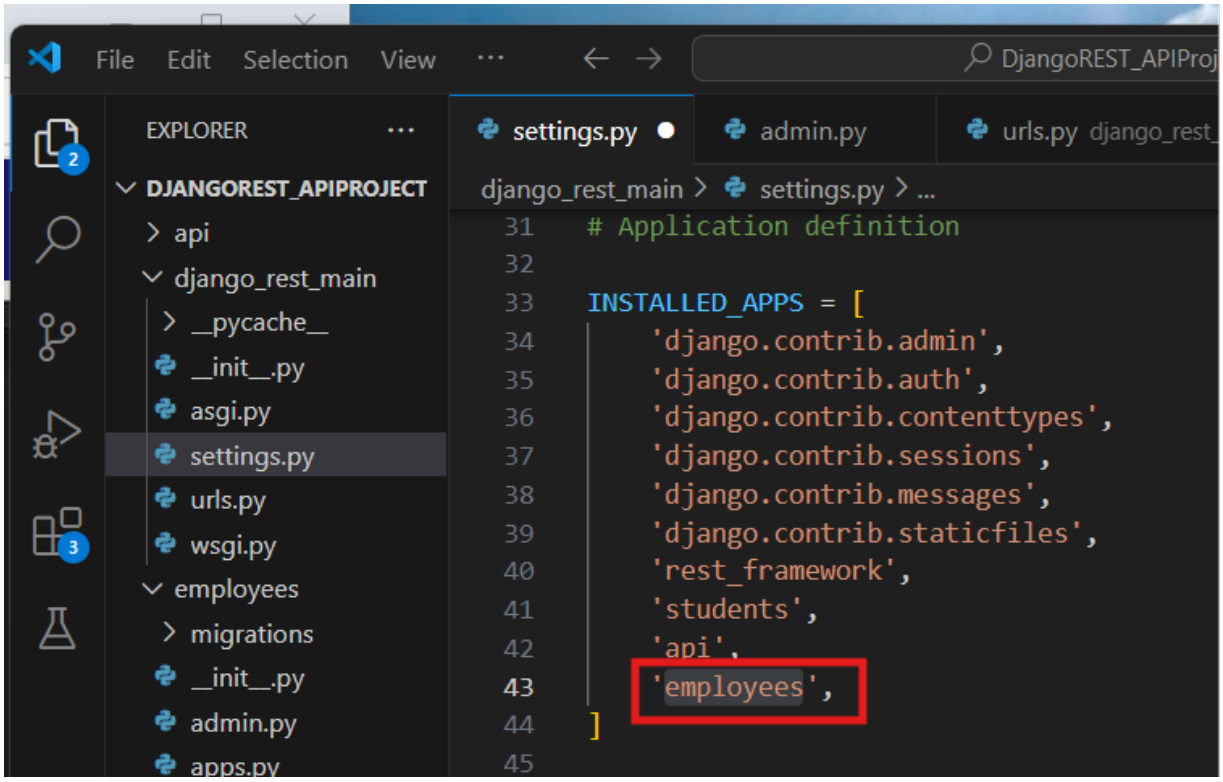
Class-based views follow the principles of object-oriented programming. These views are used for reusability and code efficiency.

For this example, we need to create a new app called EMPLOYEES.

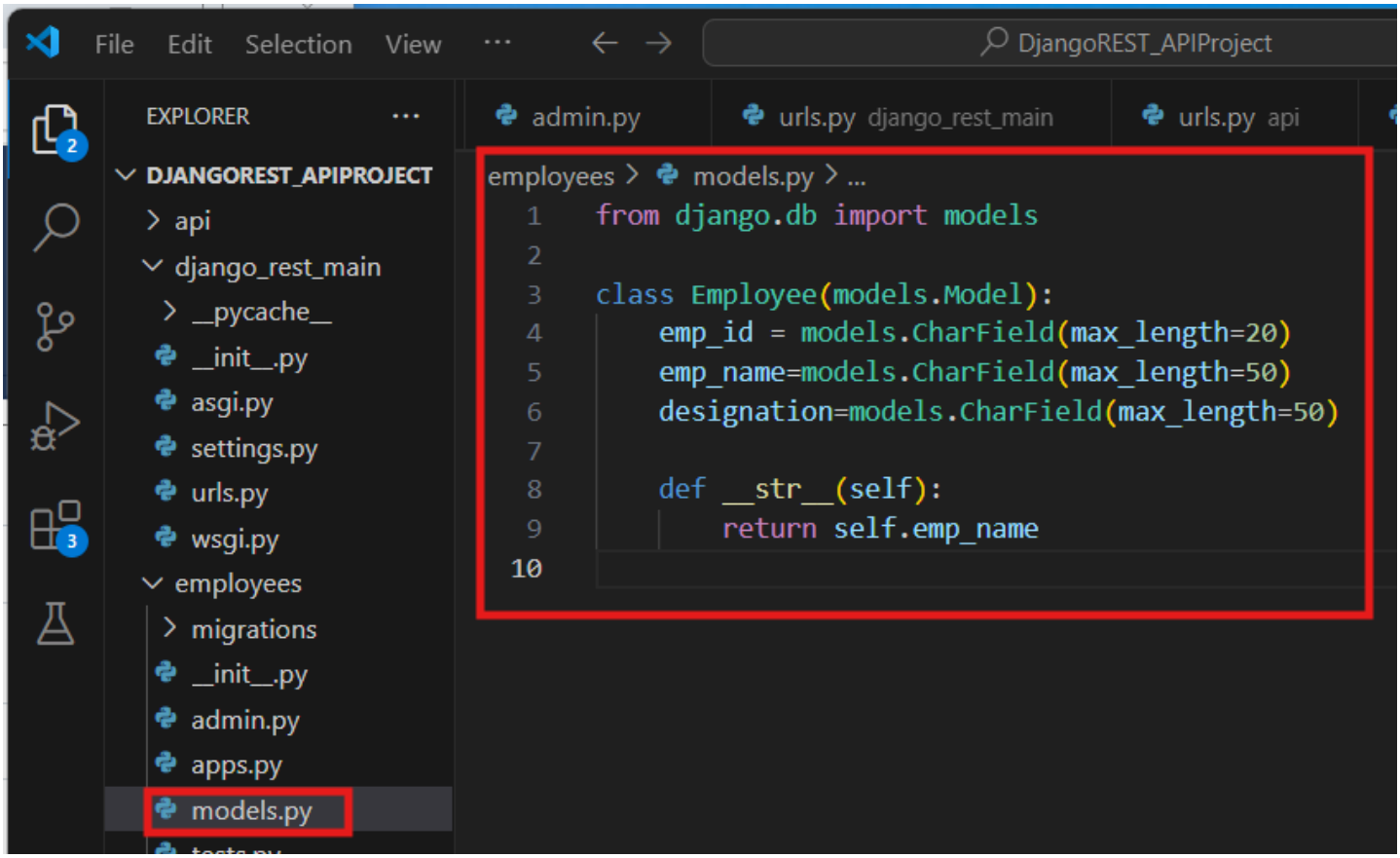
1. In the terminal, create a new app:

```
$ python manage.py startapp employees
```

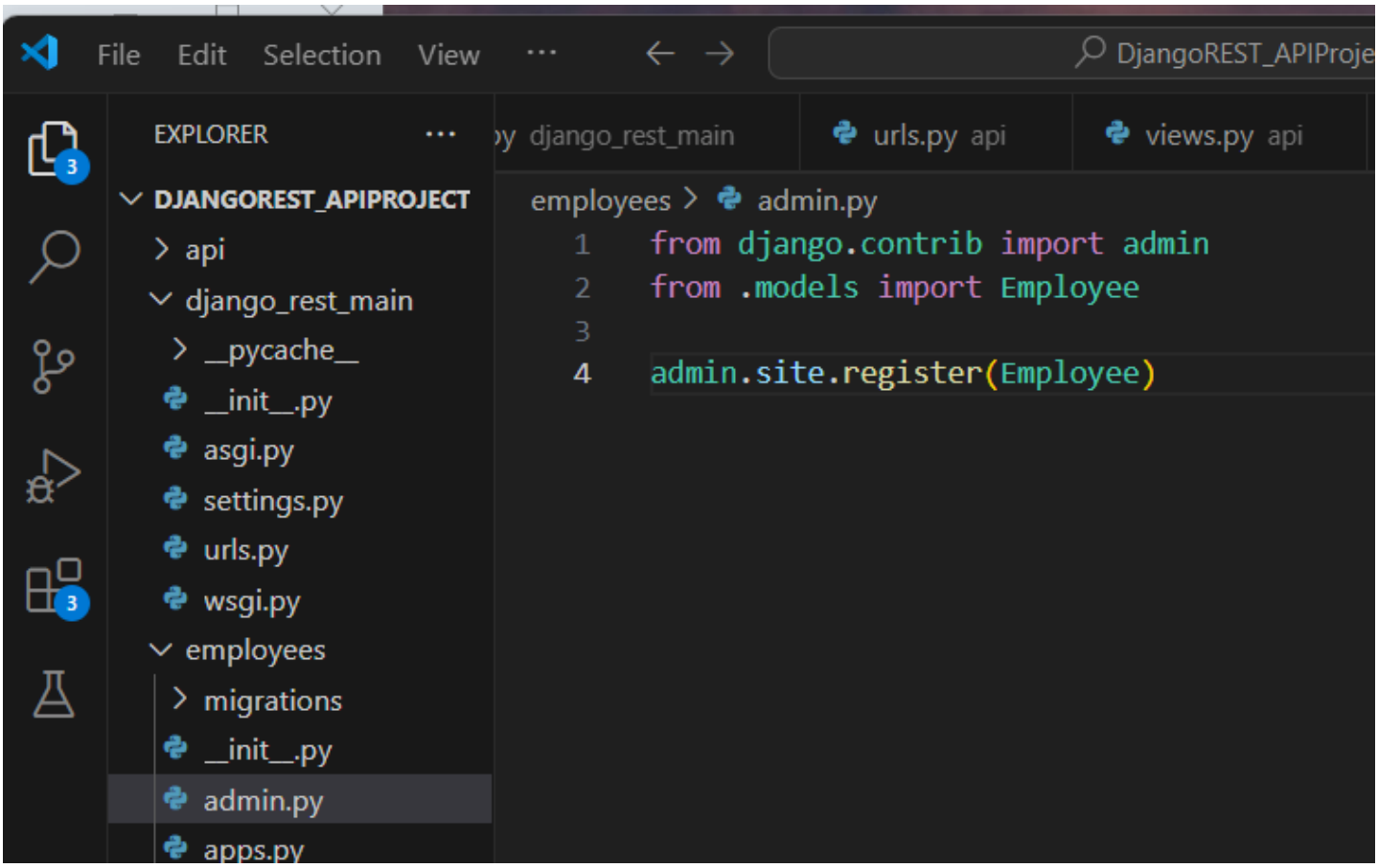
2. Register this new app in the SETTINGS.PY



3. Create the new model in MODELS.PY



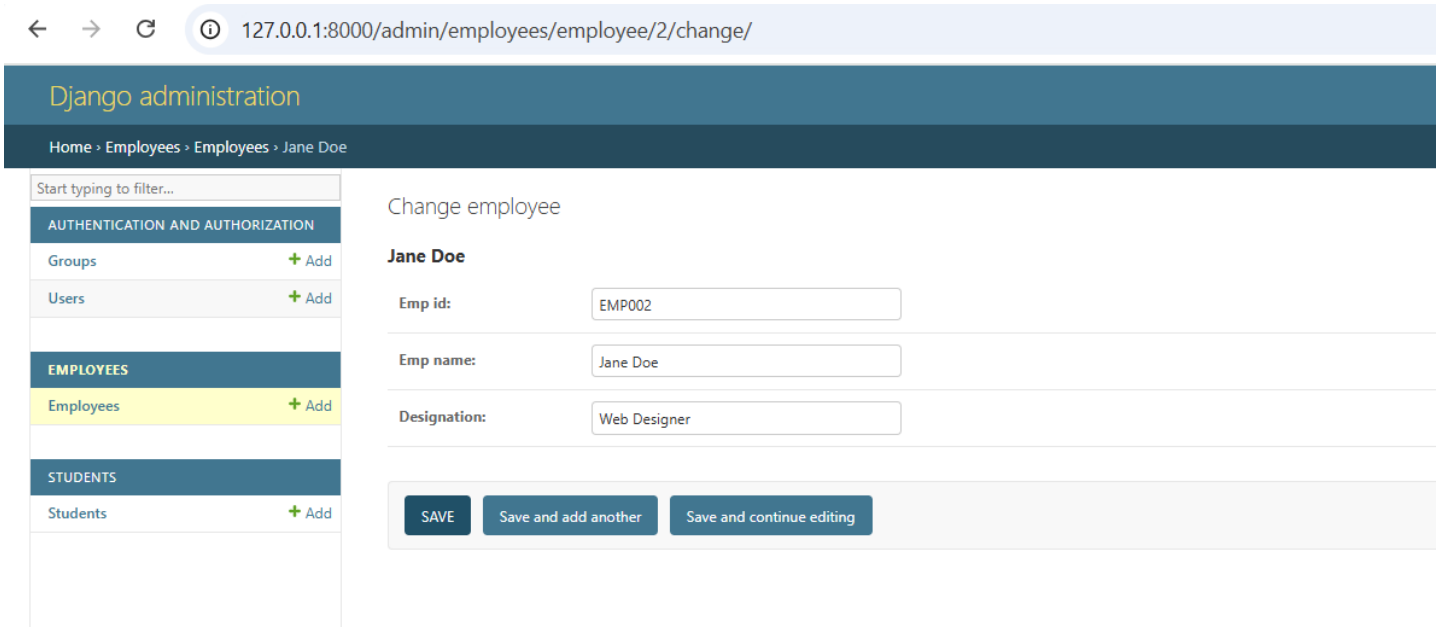
4. Update the ADMIN.PY



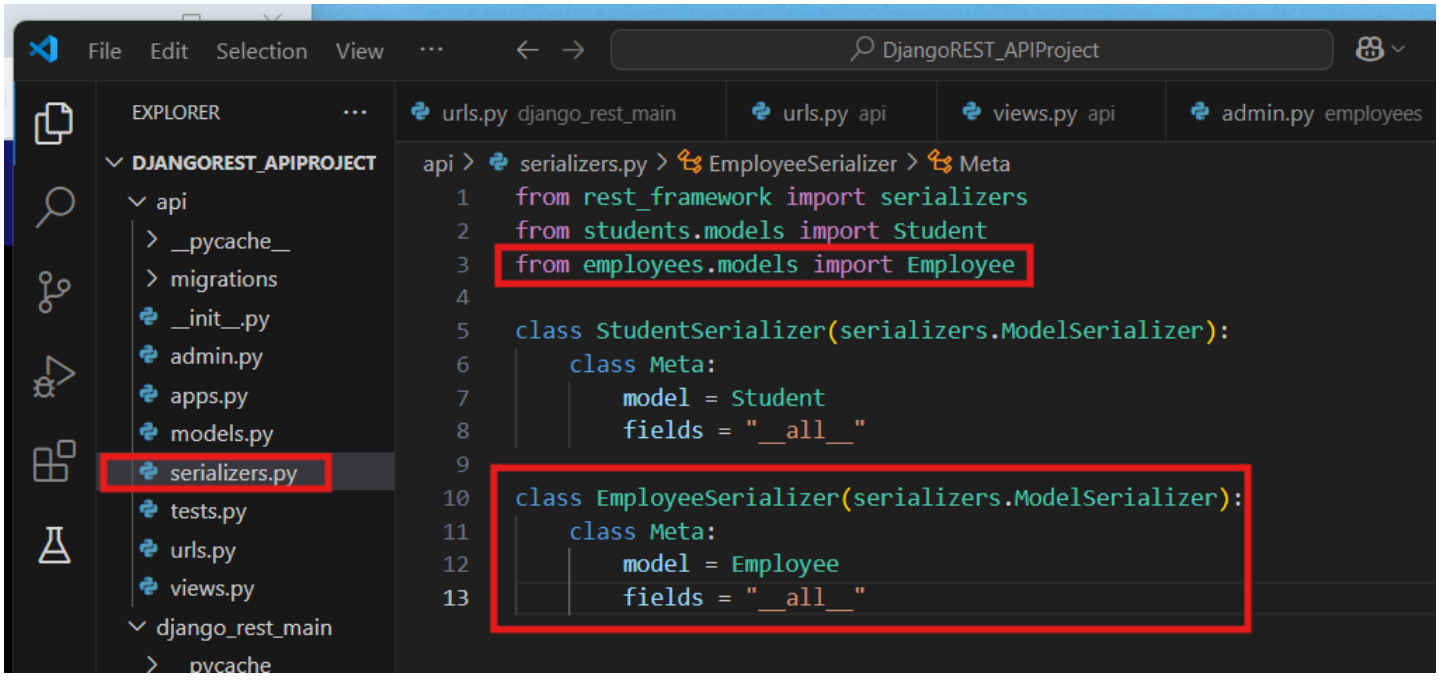
5. Make the migrations. Be sure you are in the correct folder to see the migrations happen.

```
rosil@LearnCodeRepeat MINGW64 /c/Users/rosil/OneDrive/Documents/MyCodingCareer/Django Projects/API Devt/Resources/DjangoREST_APIProject
$ python manage.py makemigrations
Migrations for 'employees':
  employees\migrations\0001_initial.py
    + Create model Employee
(env)
rosil@LearnCodeRepeat MINGW64 /c/Users/rosil/OneDrive/Documents/MyCodingCareer/Django Projects/API Devt/Resources/DjangoREST_APIProject
$ python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, employees, sessions, students
Running migrations:
  Applying employees.0001_initial... OK
(env)
rosil@LearnCodeRepeat MINGW64 /c/Users/rosil/OneDrive/Documents/MyCodingCareer/Django Projects/API Devt/Resources/DjangoREST_APIProject
$
```

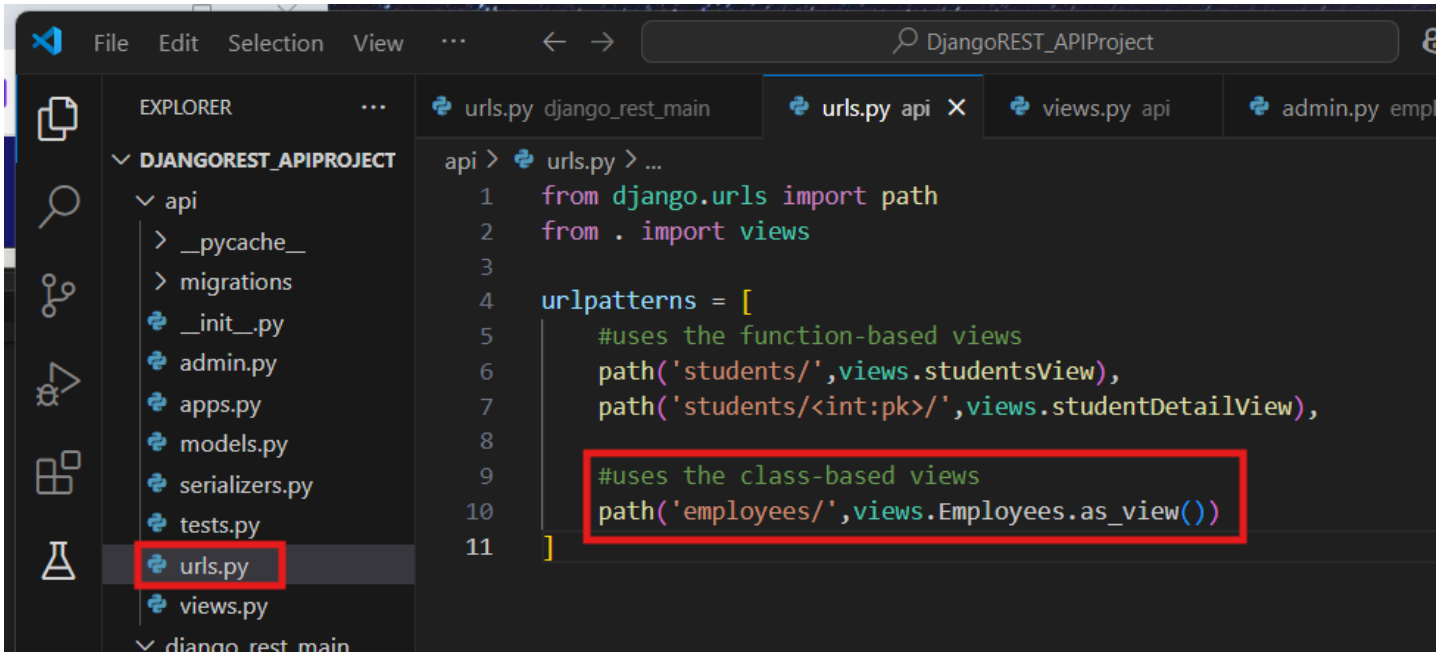
6. Check the admin panel and create new records for the Employees model:



7. Create a serializer for the Employees model. Go to API\SERIALIZER.PY

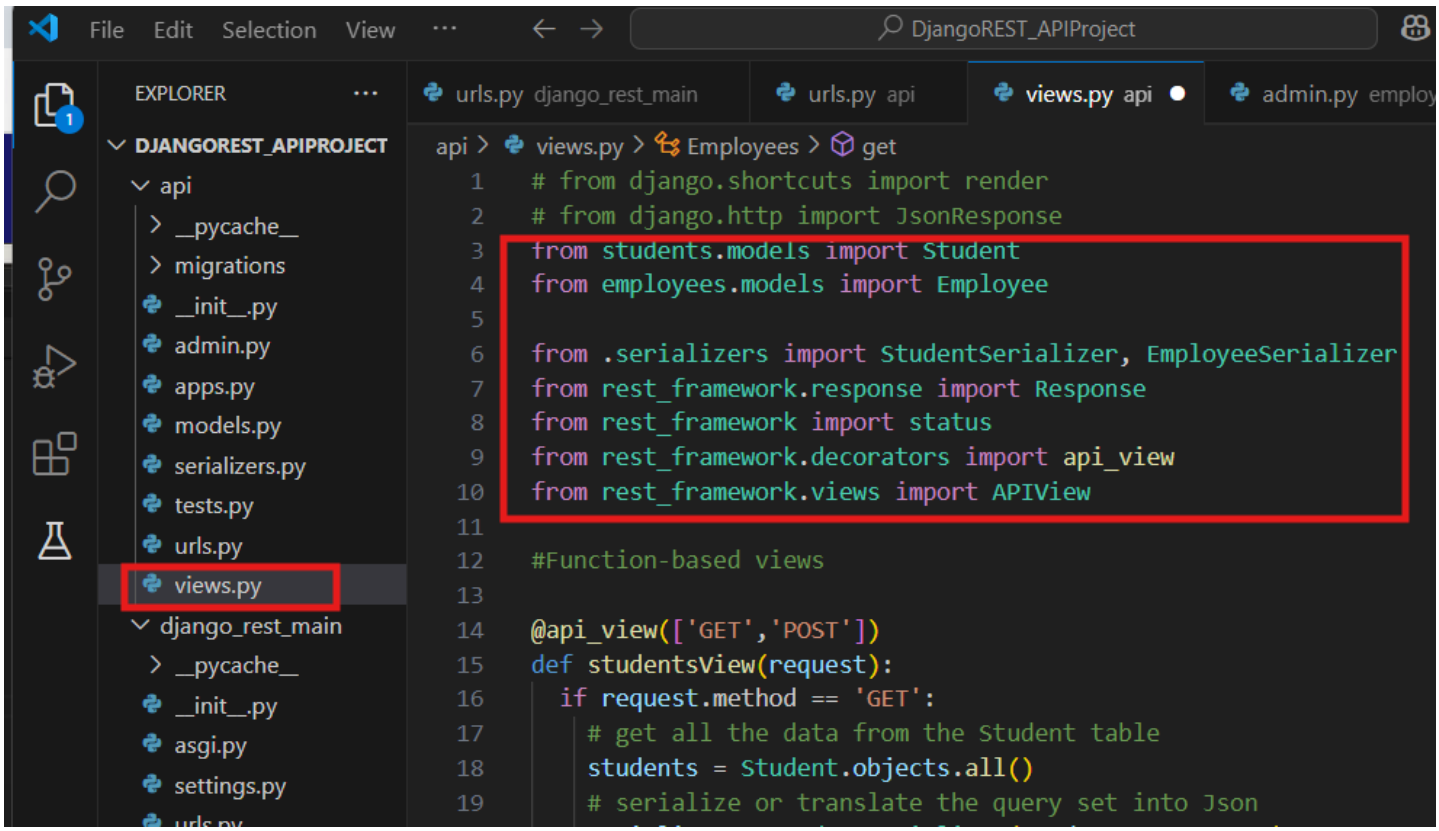


8. To retrieve all the records, go to the API\URLS.PY and update:

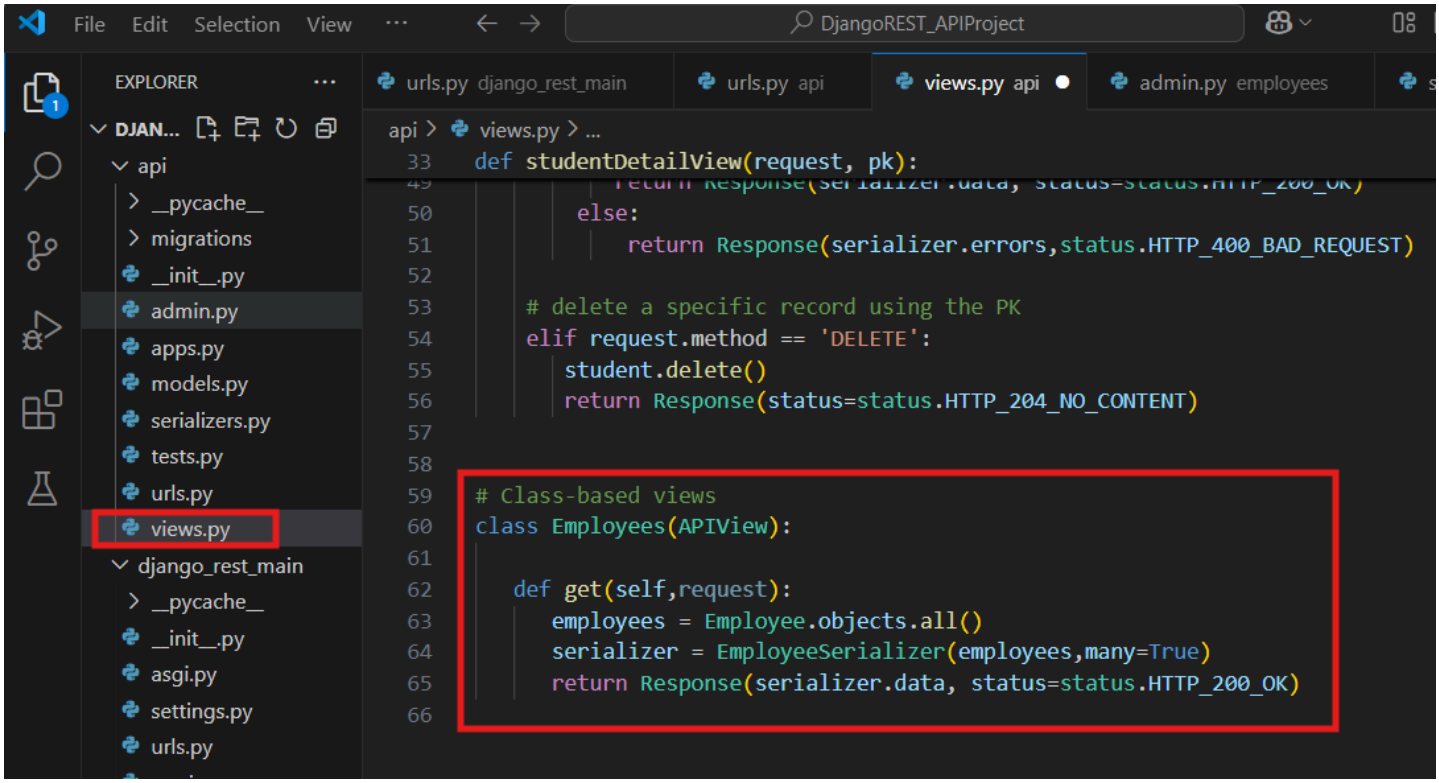


9. Create the class-based views. Go to the API\VIEWS.PY:

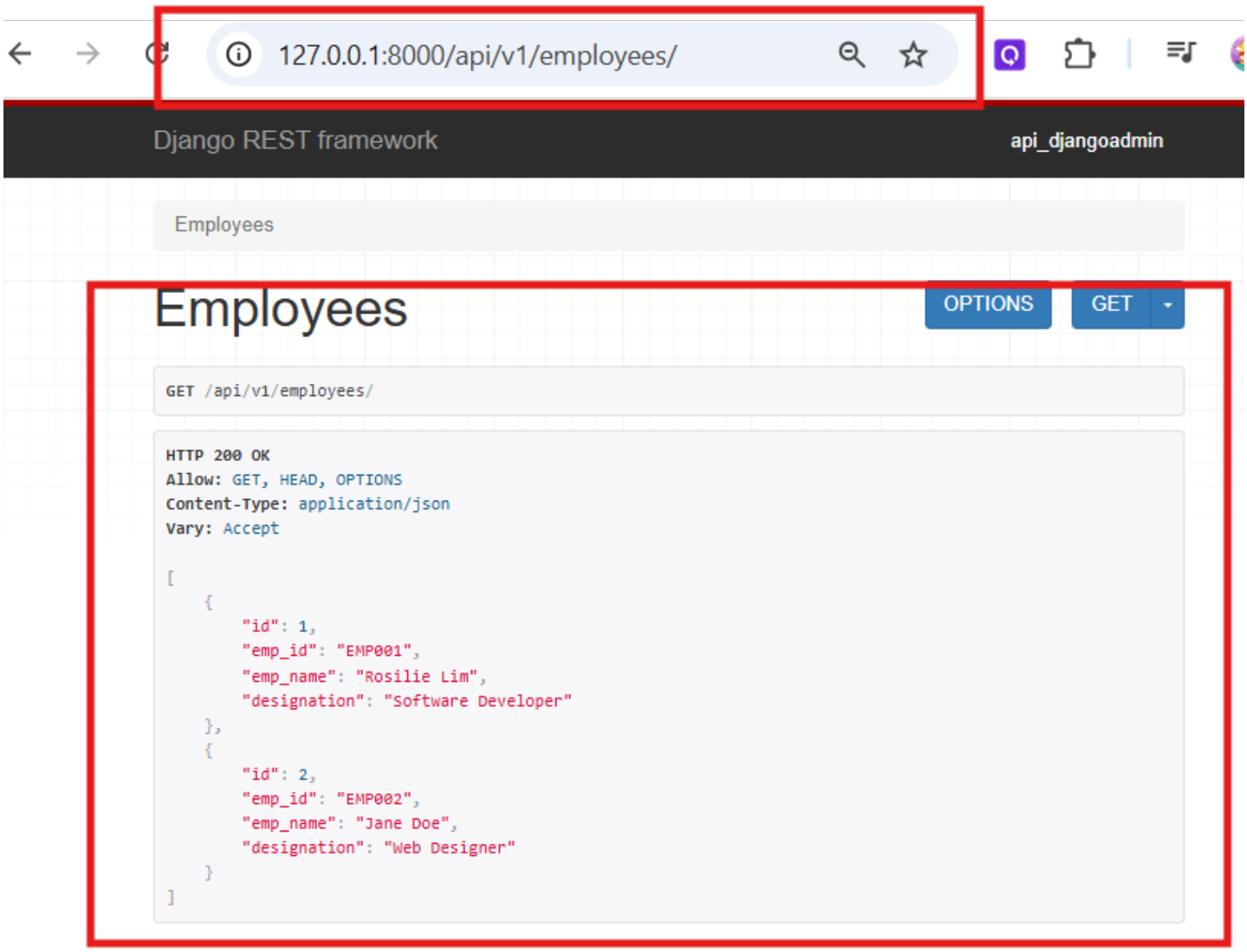
Add the necessary libraries:



Create the class Employee and its methods:



10. To test, use the URL `http://127.0.0.1:8000/api/v1/employees/`



11.