Topic: 9. CRUD Operations using Mixins

Speaker: Personal | Notebook: API Development using Django Framework

---

Mixins are a way to allow the reusability of methods. In object-oriented programming, mixin is a class with methods from other classes.

With mixins, we can reuse certain CRUD operations. These are the following:

LISTMODELMIXIN - method used is list()

CREATMODELMIXIN - method used is create()

RETRIEVEMODELMIXIN - - method used is retrieve()

UPDATEMODELMIXIN - method used is update()

DESTROYMODELMIXIN - method used is destroy()

These are used with GenericAPIView. See the documentation [here](#).

1. To view all the records of Employees using Mixins, we update the API\VIEWS.PY and comment out the classes EMPLOYEE AND EMPLOYEE DETAIL.

2. Import the needed library.

3. Create the new classes and test the URL.

```
112   # uses Mixins for CRUD
113   class Employees(mixins.ListModelMixin,mixins.CreateModelMixin, generics.GenericAPIView):
114       queryset = Employee.objects.all()
115       serializer_class = EmployeeSerializer
116
117       def get(self, request):
118           return self.list(request)
119
120       def post(self, request):
121           return self.create(request)
122
123   class EmployeeDetail(generics.GenericAPIView):
124       pass
125
```

4. Testing the URL: `http://127.0.0.1:8000/api/v1/employees/`

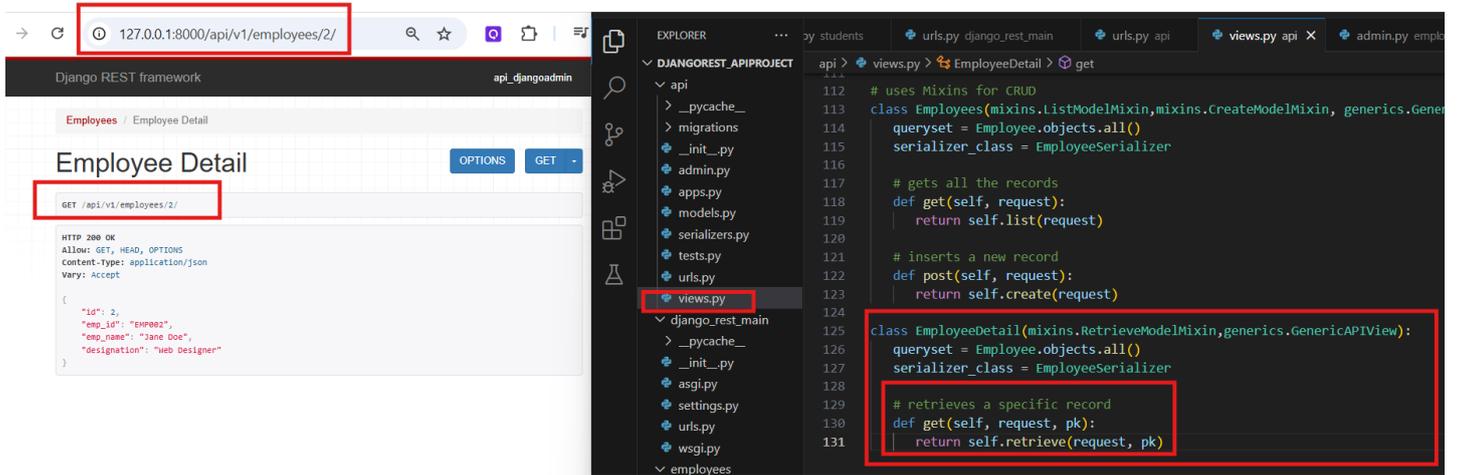This will result to having a form where we can input employee details and when submitted, this will be added to our Employee model.

5. For single - record Read/Update/Delete operations, we update our API\APPS.PY as :

6. To update a specific record, we use the UPDATEMODELMIXIN.

To test:

GET /api/v1/employees/2/

HTTP 200 OK
Allow: GET, PUT, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "id": 2,
    "emp_id": "EMP002",
    "emp_name": "Jane Doe",
    "designation": "Web Designer"
}

Emp id	EMP002
Emp name	Janet Doe
Designation	Web Designer

This will update the record to:

Django REST framework   api_djangoadmin
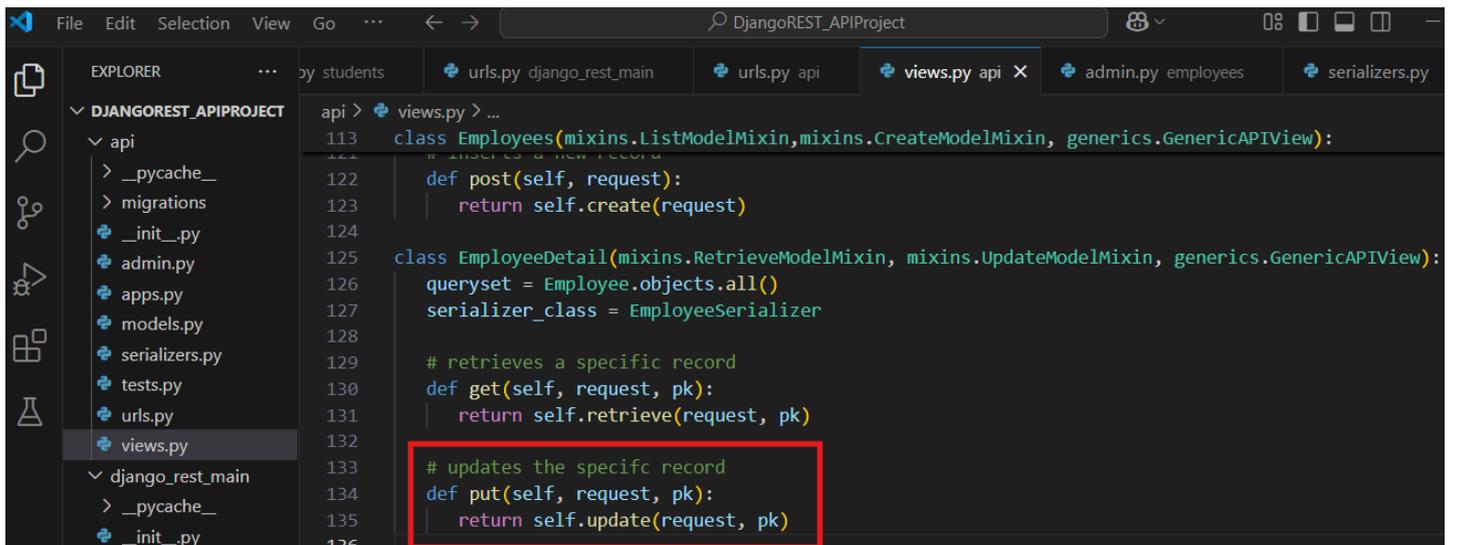
Employees / Employee Detail

Employee Detail   OPTIONS   GET ▾

```
PUT /api/v1/employees/2/
```

```
HTTP 200 OK
Allow: GET, PUT, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "id": 2,
    "emp_id": "EMP002",
    "emp_name": "Janet Doe",
    "designation": "Web Designer"
}
```

Raw data   HTML form

Emp id	EMP002
Emp name	Janet Doe
Designation	Web Designer

PUT

7. To delete a specific record.

```
      # uses Mixins for CRUD
112   class Employees(mixins.ListModelMixin,mixins.CreateModelMixin, generics.GenericAPIView):
114       queryset = Employee.objects.all()
115       serializer_class = EmployeeSerializer
116
117       # gets all the records
118       def get(self, request):
119           return self.list(request)
120
121       # inserts a new record
122       def post(self, request):
123           return self.create(request)
124
125   class EmployeeDetail(mixins.RetrieveModelMixin, mixins.UpdateModelMixin, mixins.DestroyModelMixin, generics.GenericAPIView):
126       queryset = Employee.objects.all()
127       serializer_class = EmployeeSerializer
128
129       # retrieves a specific record
130       def get(self, request, pk):
131           return self.retrieve(request, pk)
132
133       # updates the specific record
134       def put(self, request, pk):
135           return self.update(request, pk)
136
137       # deletes/destroys a specific record
138       def delete(self, request, pk):
139           return self.destroy(request, pk)
```

8. To test and delete record 2,

→ C ⓘ 127.0.0.1:8000/api/v1/employees/2/ 🔍 ☆ ⬛ ⤵ | ☰♪ 🦄

Django REST framework   **api_djangoadmin**

Employees / Employee Detail

Employee Detail   DELETE   OPTIONS   GET ▾

```
GET /api/v1/employees/2/
```

```
HTTP 200 OK
Allow: GET, PUT, DELETE, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "id": 2,
    "emp_id": "EMP002",
    "emp_name": "Janet Doe",
    "designation": "Web Designer"
}
```

                                              Raw data   HTML form

Emp id   EMP002

Emp name   Janet Doe

Designation   Web Designer

                                                            PUT

9.