Topic: 10. CRUD using Generics

Speaker: Personal | Notebook: API Development using Django Framework

---



Generics according to Django Documentation:

Django's generic views... were developed as a shortcut for common usage patterns... They take certain common idioms and patterns found in view development and abstract them so that you can quickly write common views of data without having to repeat yourself.

GenericAPIView

This class extends the REST framework's `APIView` class, adding commonly required behavior for standard list and detail views.

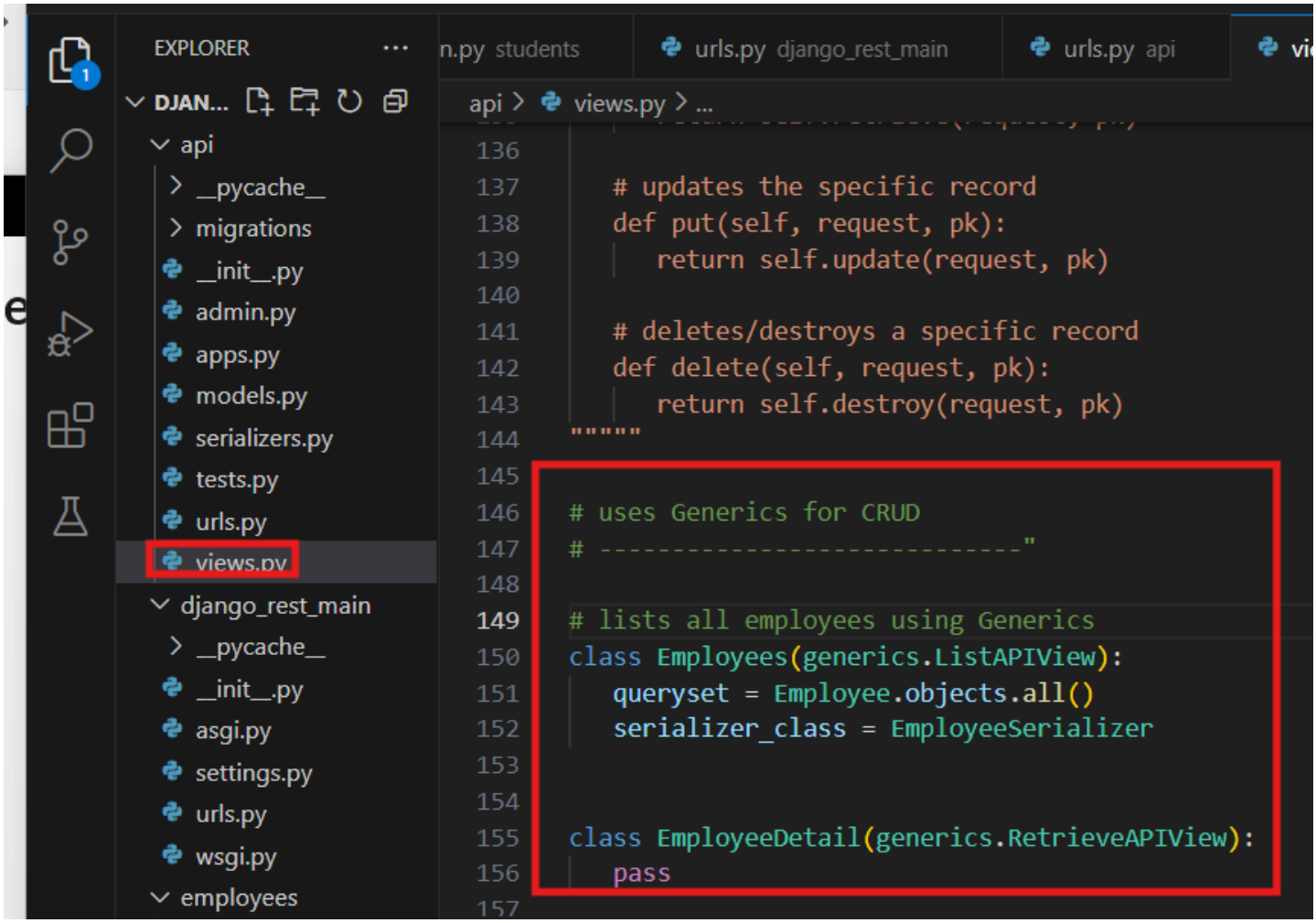
Each of the concrete generic views provided is built by combining `GenericAPIView`, with one or more mixin classes.

To read more about Generics, use this link to Medium Digest

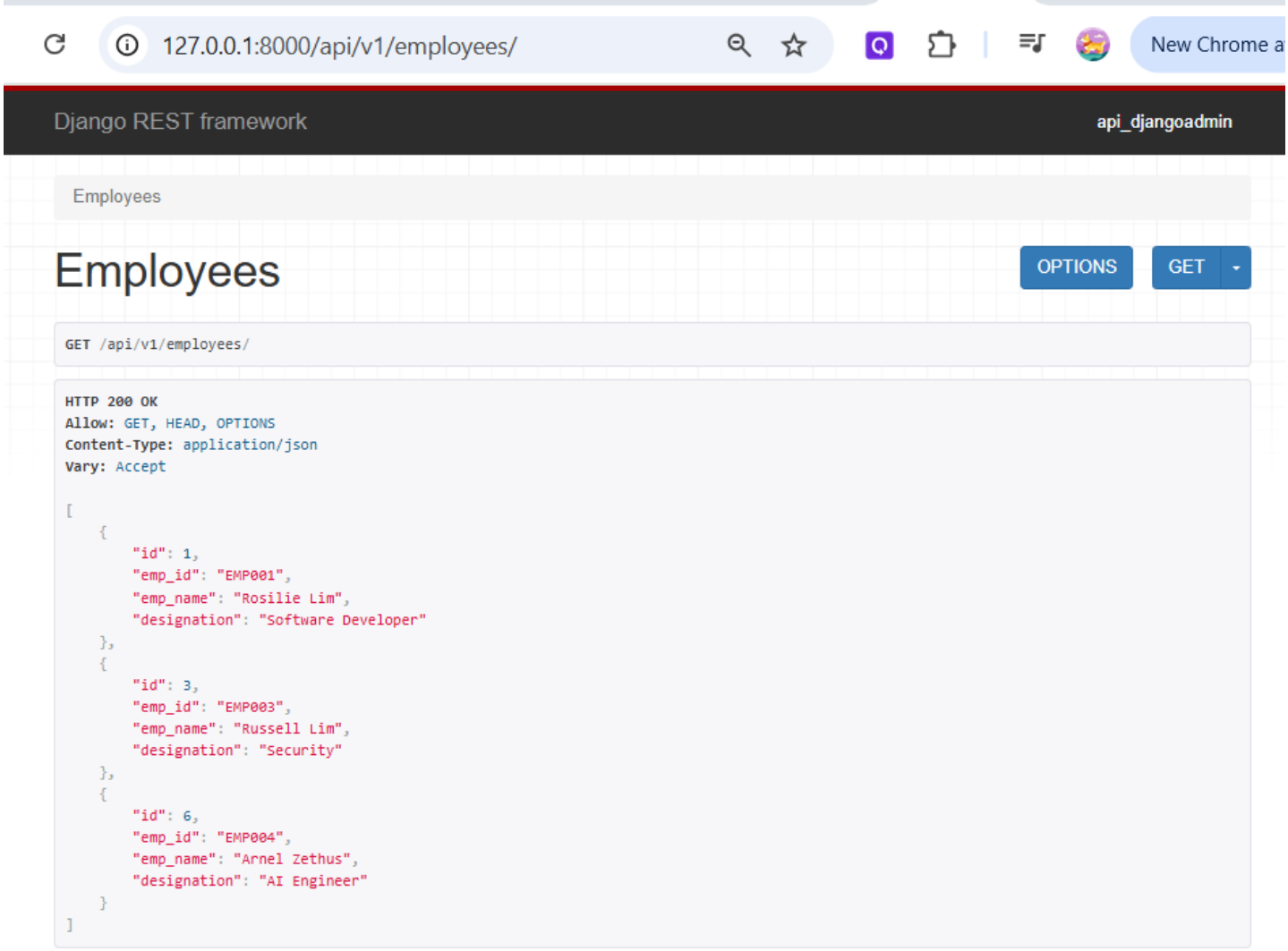
The API views are:

`ListAPIView`	`RetrieveAPIView`
`CreateAPIView`	`DestroyAPIView`
`UpdateAPIView`	`ListAPICreateView`
`RetrieveUpdateAPIView`	`RetrieveUpdateDestroyAPIView`

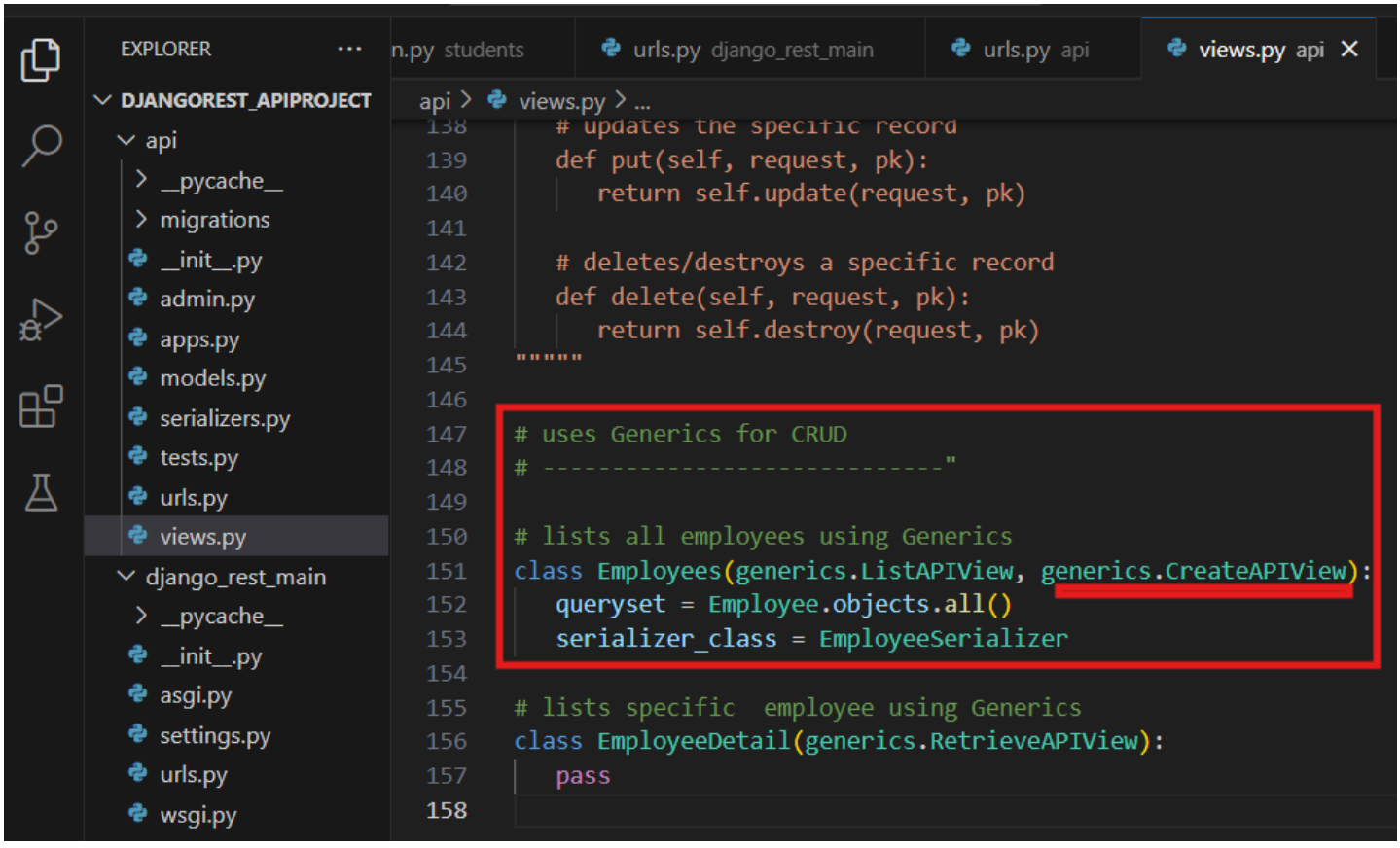
1. To apply CRUD on Employee model using Generics. Comment out the code using Mixins and add:

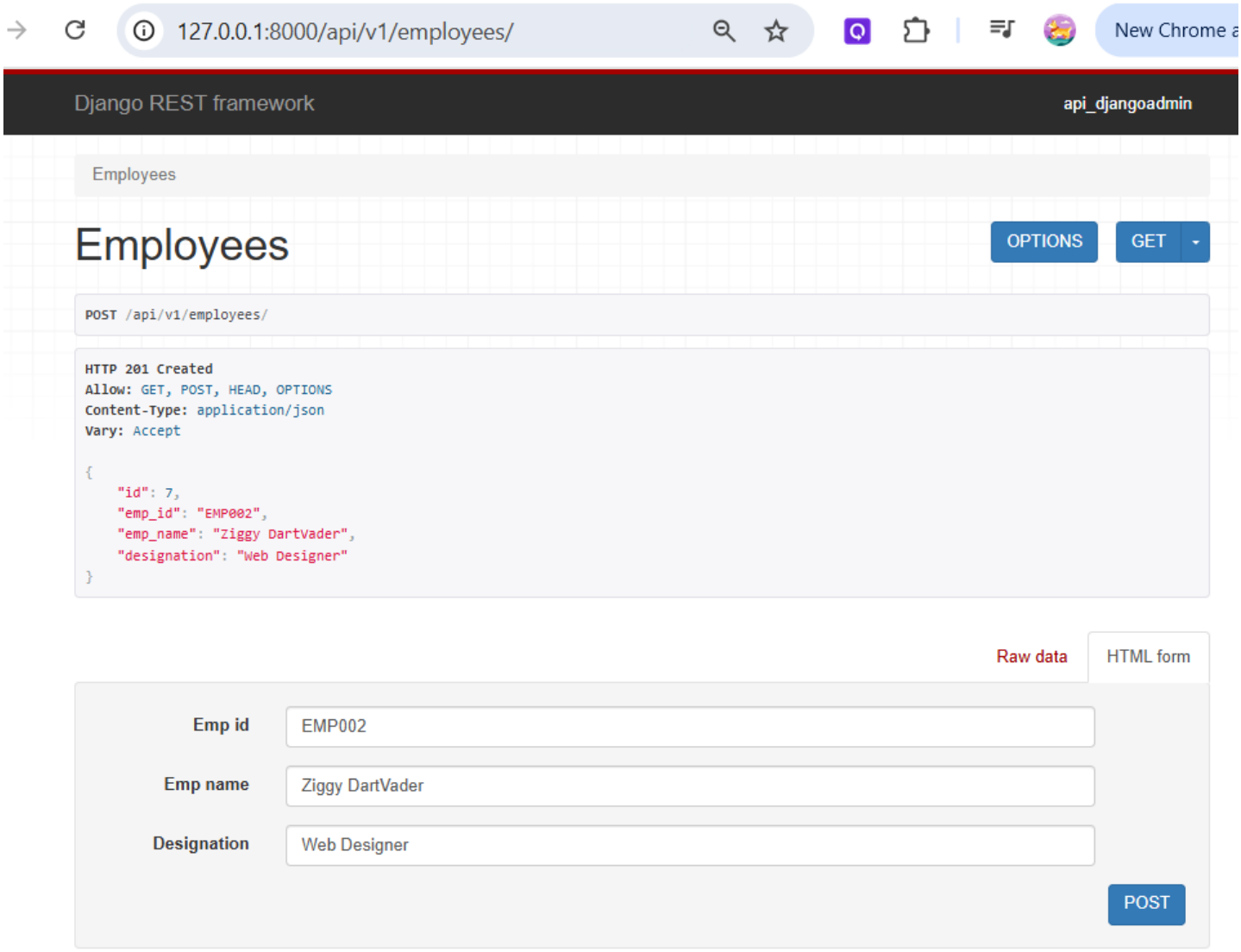


Running the URL path for listing the employees:

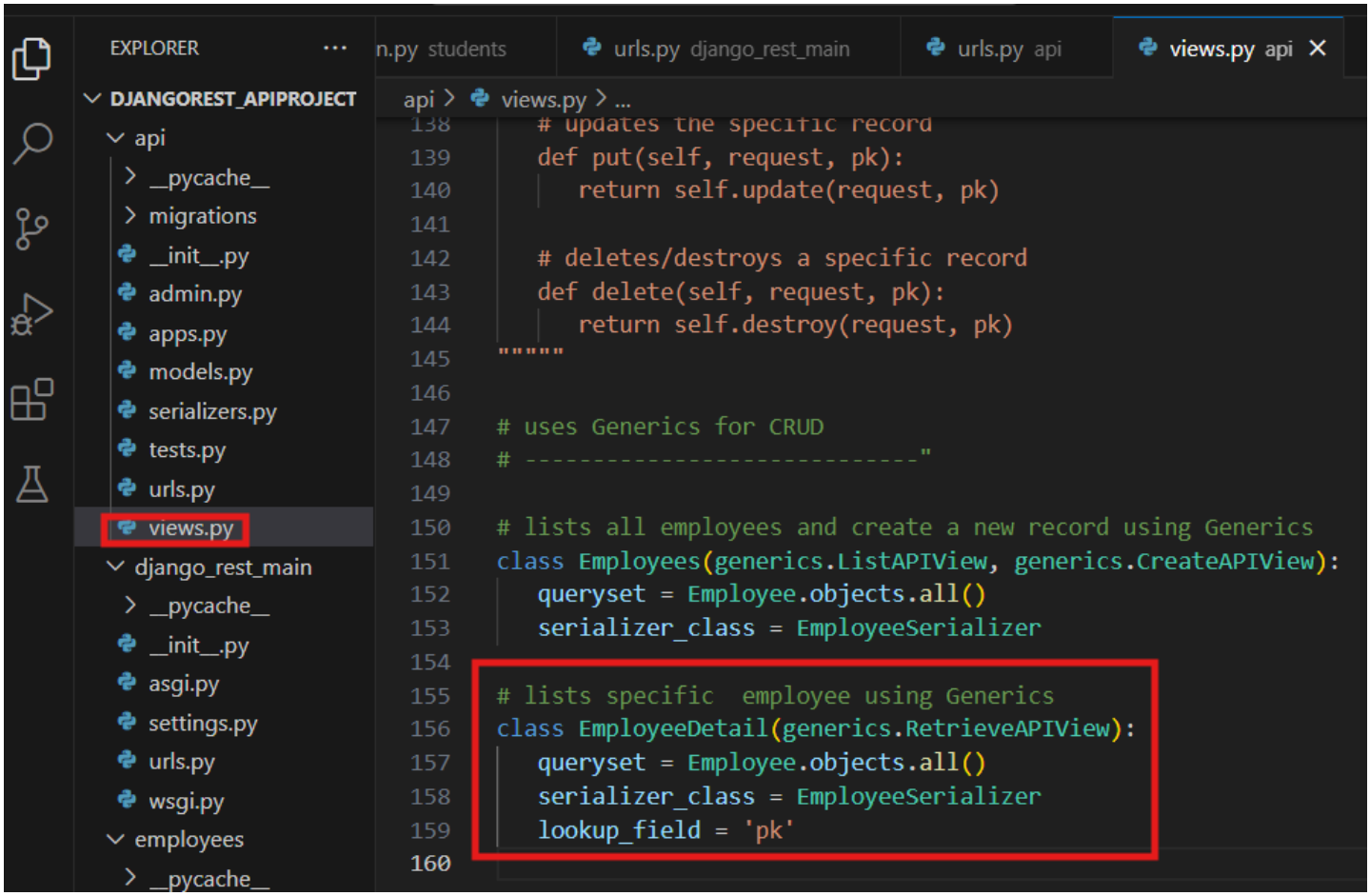


To allow a FORM for creating an employee, we add a `generics.CreateAPIView` to inherit.

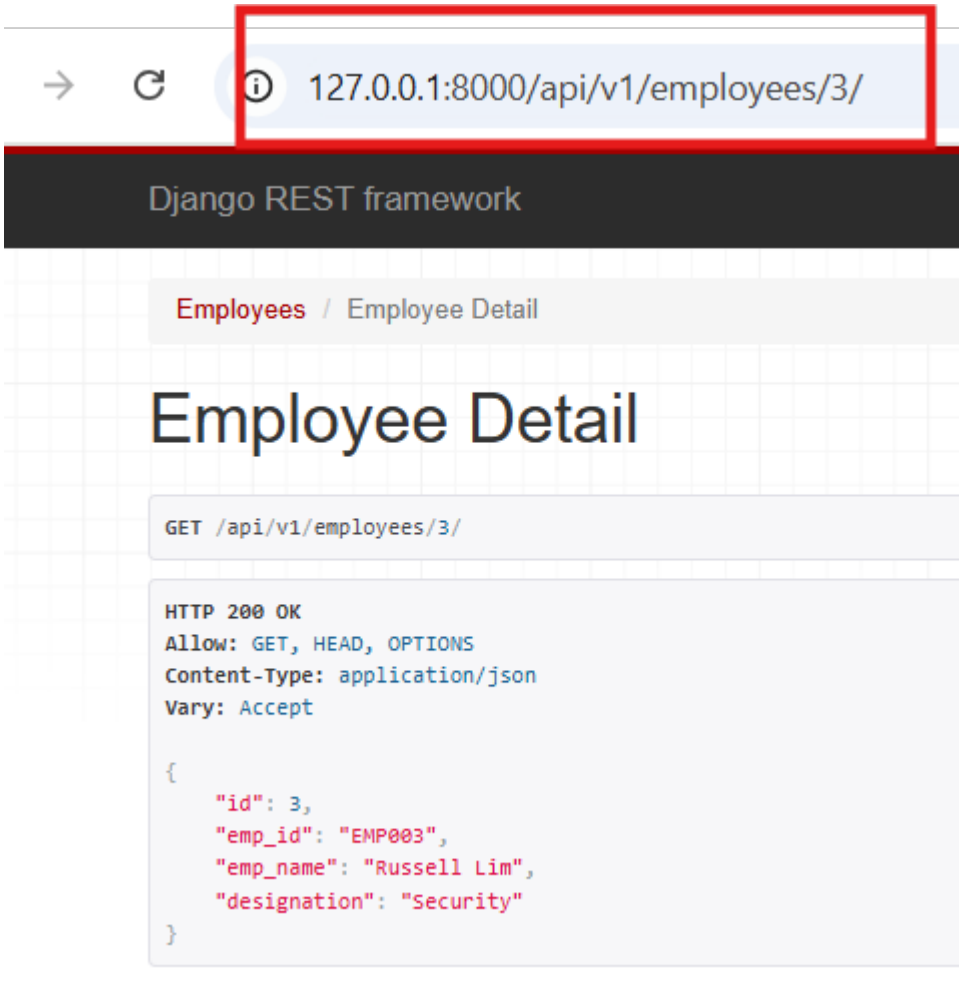




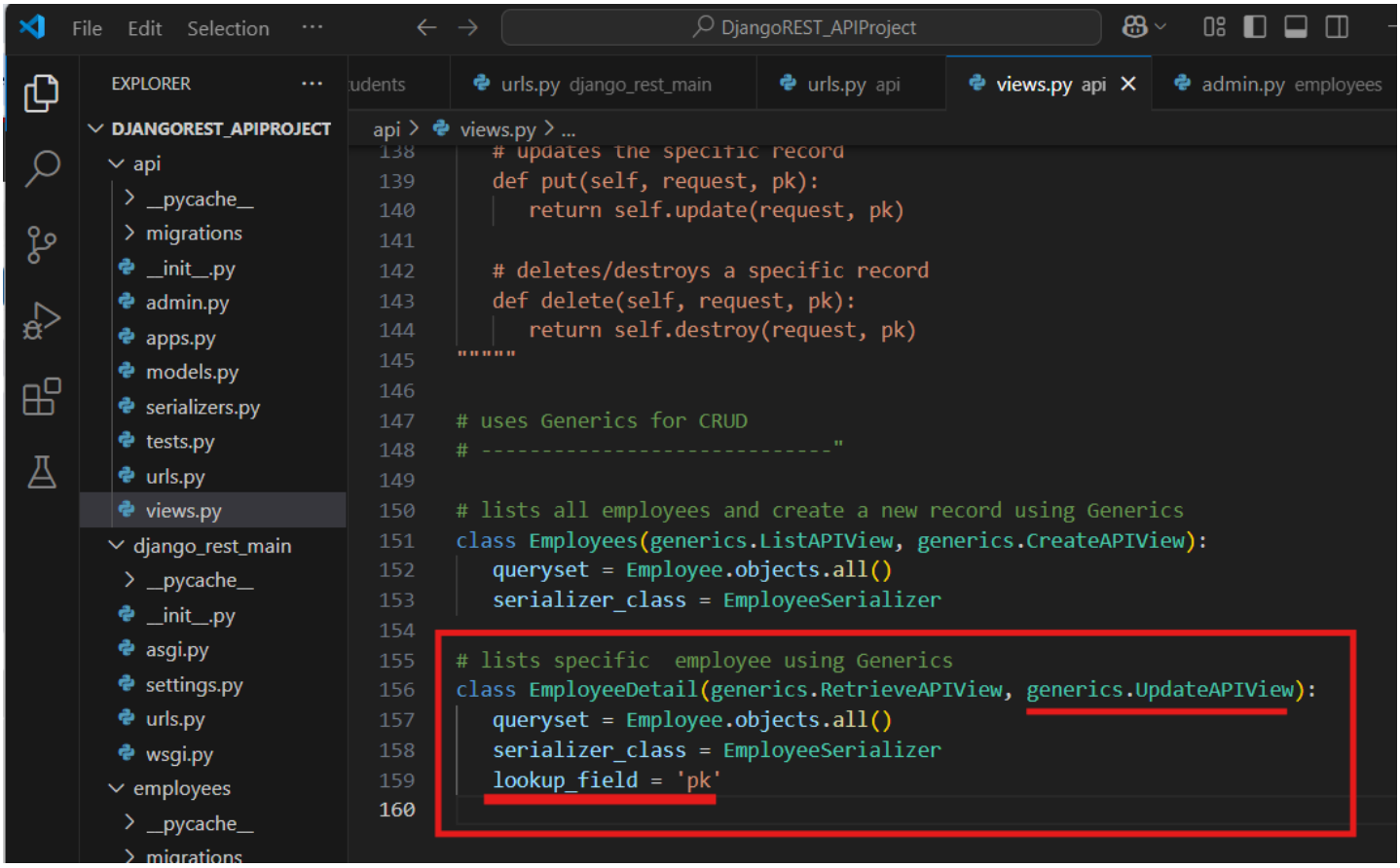
2. To list the Employee Detail, we use the generics.RetrieveAPIView:



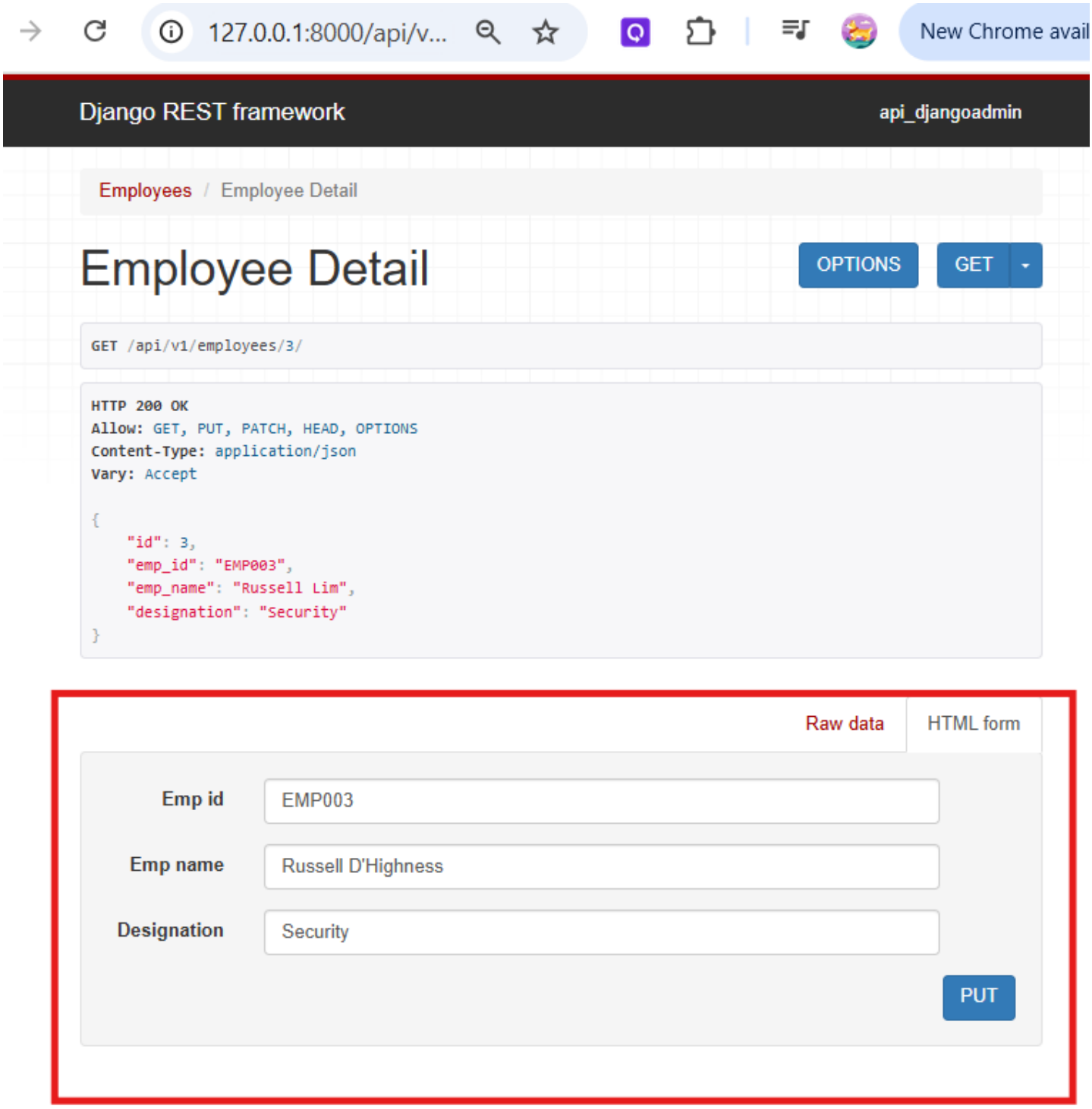
To view the specific record:



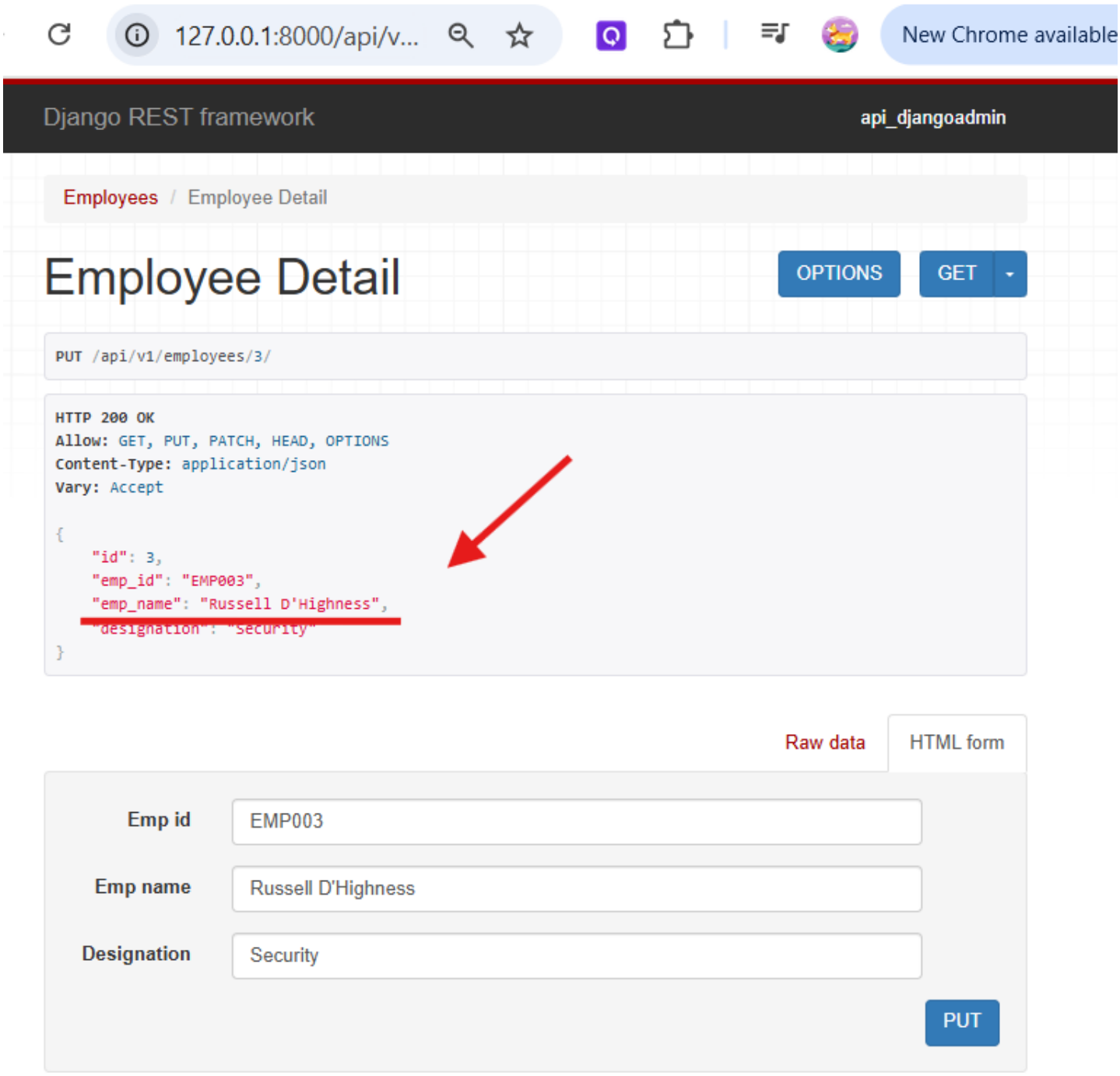
3. To update the record, we use the generics.UpdateAPIView:



Updating the form:



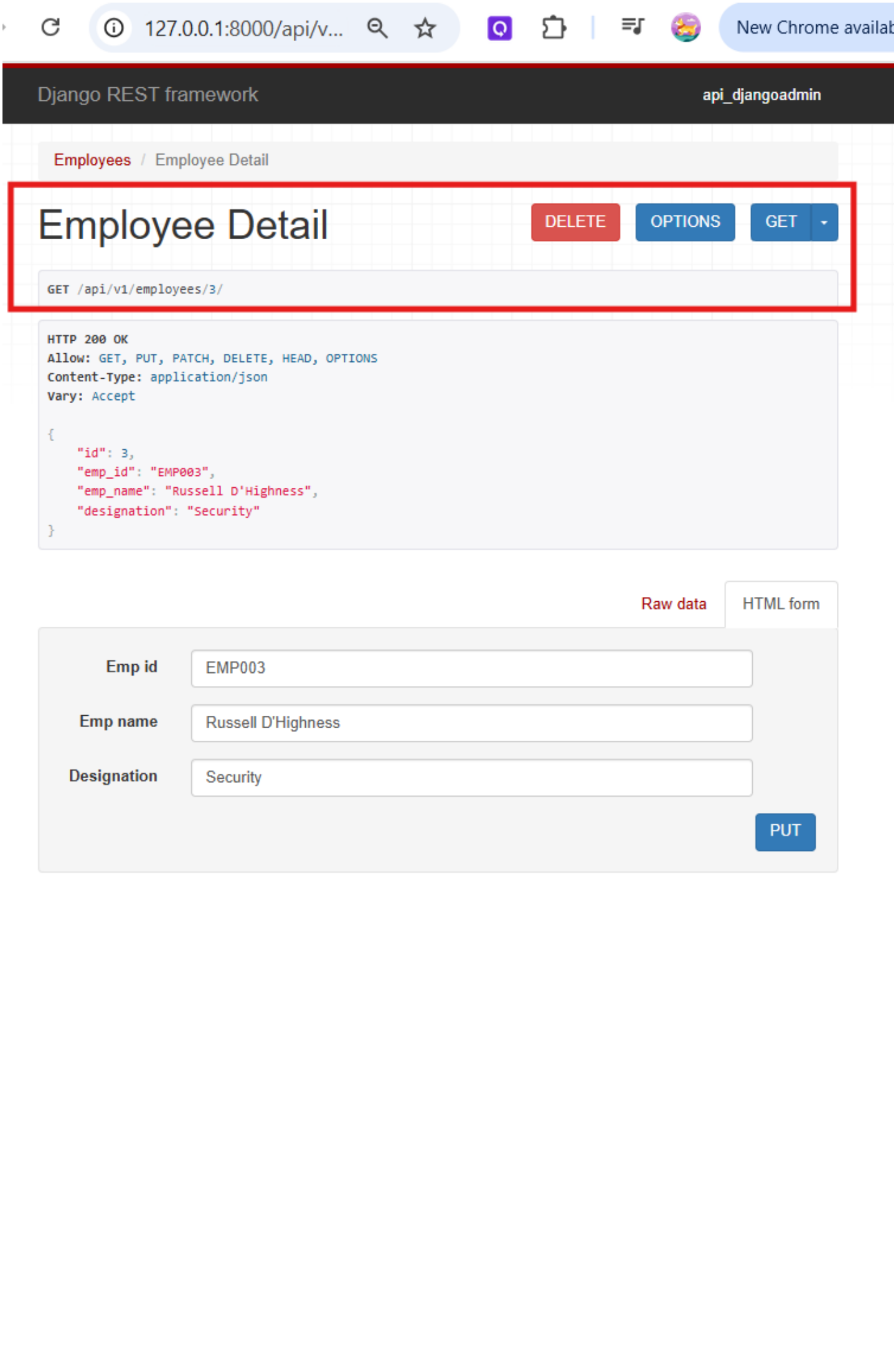
After the PUT submission:

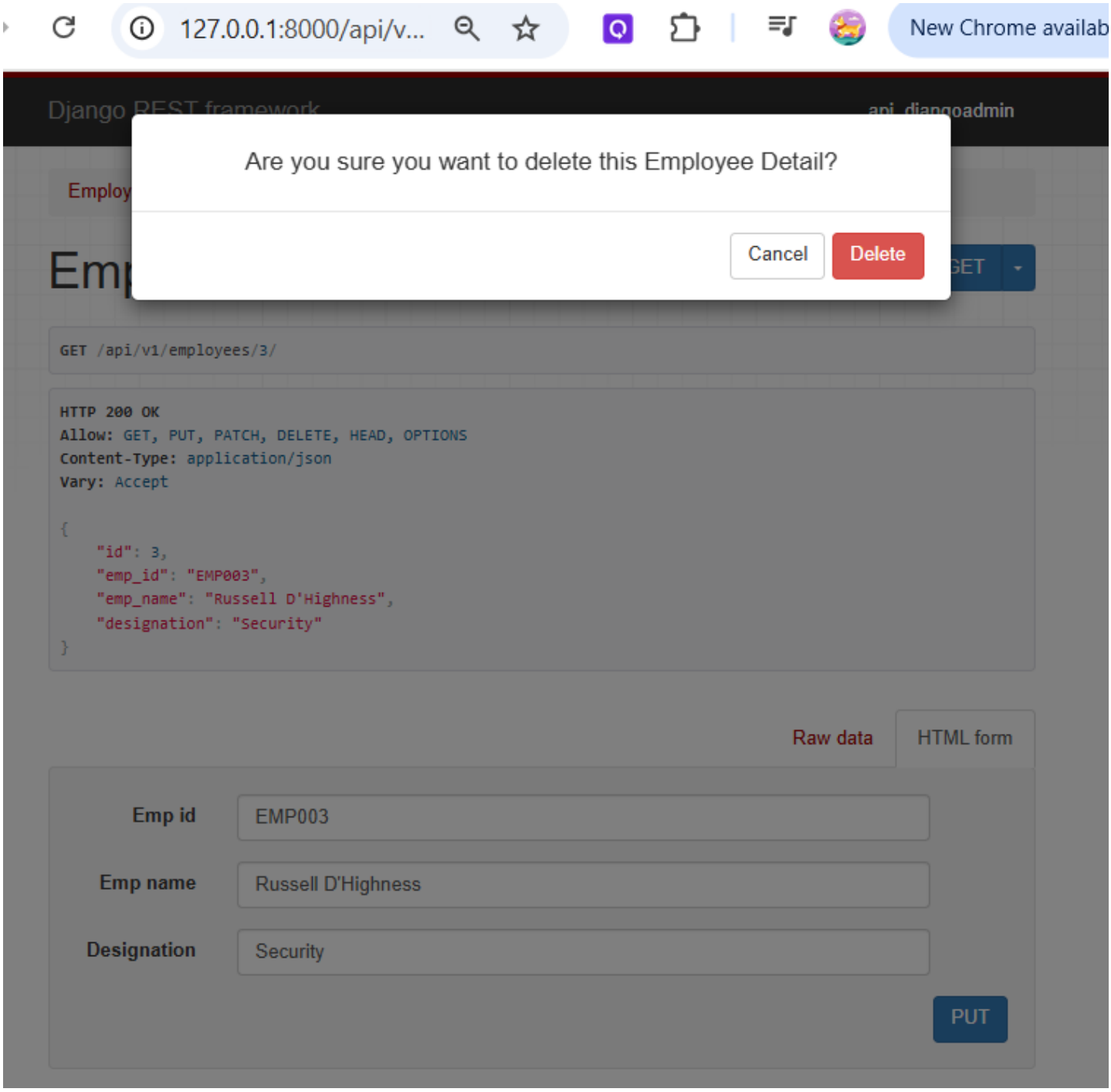


4. To delete a record, we add the generics.DestroyAPIView:

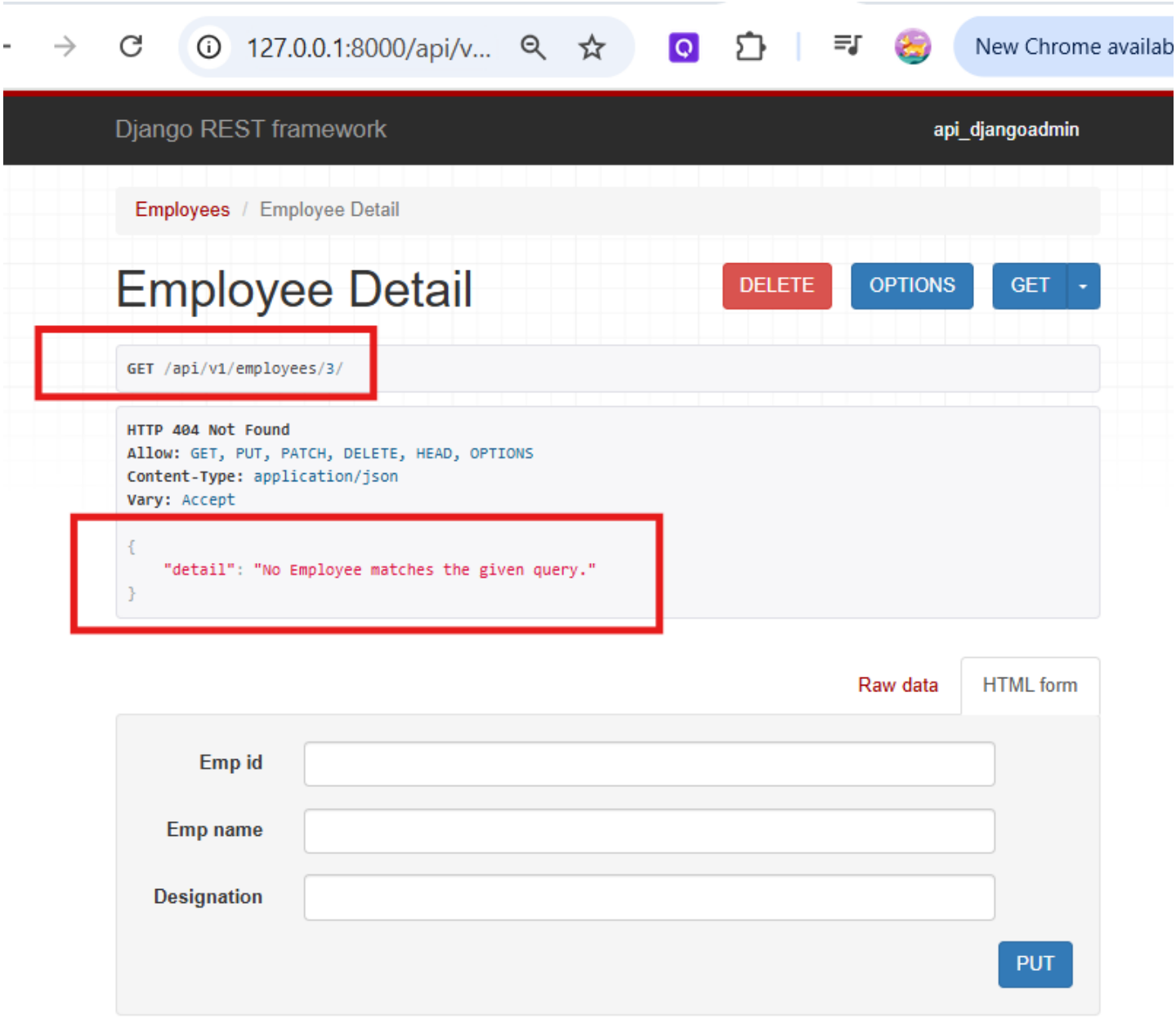
```
# lists/updates/deletes specific  employee using Generics
class EmployeeDetail(generics.RetrieveAPIView, generics.UpdateAPIView, generics.DestroyAPIView):
    queryset = Employee.objects.all()
    serializer_class = EmployeeSerializer
    lookup_field = 'pk'
```

Before DELETE action:


127.0.0.1:8000/api/v...
New Chrome availab
Django REST framework
api_djangoadmin
Employees / Employee Detail
Employee Detail
DELETE
OPTIONS
GET
GET /api/v1/employees/3/
HTTP 200 OK
Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept
{
"id": 3,
"emp_id": "EMP003",
"emp_name": "Russell D'Highness",
"designation": "Security"
}
Raw data
HTML form
Emp id
EMP003
Emp name
Russell D'Highness
Designation
Security
PUT



AFTER deleting record # 3:

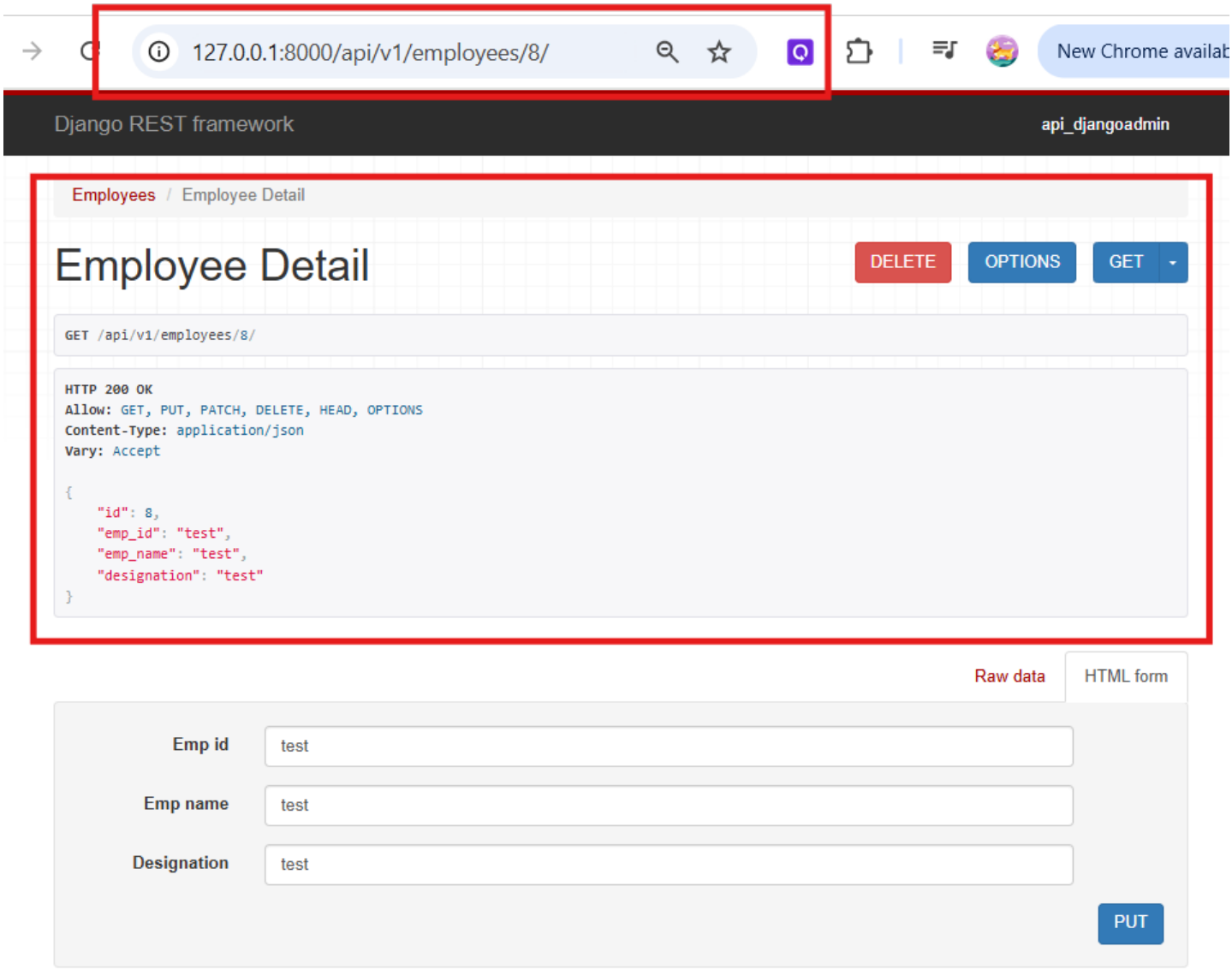


5. To delete/update/list a specific record using only `generics.RetrieveUpdateDestroyAPIView`.

```python
# lists/updates/deletes specific  employee using Generics using the long way
# class EmployeeDetail(generics.RetrieveAPIView, generics.UpdateAPIView, generics.DestroyAPIView):
#     queryset = Employee.objects.all()
#     serializer_class = EmployeeSerializer
#     lookup_field = 'pk'

VS

# lists/updates/deletes specific  employee using Generics the faster way
class EmployeeDetail(generics.RetrieveUpdateDestroyAPIView):
    queryset = Employee.objects.all()
    serializer_class = EmployeeSerializer
    lookup_field = 'pk'
```



In summary, Generics views are easier and more efficient than using other ways like function-based views or mixins.

Copyright © Personal Digital Notebooks | By Rosilie | Date Printed: Nov. 5, 2026, 8:59 a.m.