

### Select Vulnerability Category

Identify the type of vulnerability present in the code and select it from the options below.  
The vulnerable files and lines of code have been marked with ✖.

File Upload Vulnerability

Unrestricted File Upload

Vulnerable Components

Using Components From Untrusted Source

Injection Flaws

SQL Injection

Insufficient Transport Layer Protection

Unprotected Transport of Credentials

Hint

Submit

### Debug Information

Challenge: 5eecd3a136ea0171cf25eea2

✖ gradebook

✖ api

dev

test

\_\_init\_\_.py

admin.py

apps.py

models.py

pagination.py

permissions.py

serializers.py

tests.py

urls.py

✖ views.py

app

comments

grades

users

manage.py

.gitignore

requirements.txt

```
212 API endpoint for viewing public courses.
213 """
214 queryset = Course.objects.filter(is_active=True).order_by('start_date')
215 serializer_class = CoursePublicSerializer
216 permission_classes = (AllowAny, )
217
218 def get_object(self):
219     """
220     Returns the object the view is displaying.
221     """
222     queryset = self.filter_queryset(self.get_queryset())
223
224     # Perform the lookup filtering.
225     lookup_url_kwarg = self.lookup_url_kwarg or self.lookup_field
226
227     assert lookup_url_kwarg in self.kwargs, (
228         'Expected view %s to be called with a URL keyword argument \'
229         'named "%s". Fix your URL conf, or set the `lookup_field` '
230         'attribute on the view correctly.' %
231         (self.__class__.__name__, lookup_url_kwarg)
232     )
233
234     filter_kwargs = {self.lookup_field: self.kwargs[lookup_url_kwarg]}
235     results = list(queryset.raw(
236         'SELECT * FROM grades_course WHERE id=%s' % filter_kwargs.get('pk')
237     ))
238
239     if not results:
240         raise NotFound
241
242     obj = results.pop()
243
244     # May raise a permission denied
245     self.check_object_permissions(self.request, obj)
246
247     return obj
248
249 def retrieve(self, request, *args, **kwargs):
250     instance = self.get_object()
251     serializer = self.get_serializer(instance)
252     return Response(serializer.data)
253
254
```