



Quickswitch



Jump



1

Locate Vulnerability



Challenge Complete

Locate Vulnerability

Identify and select the code blocks that cause the vulnerability listed below by clicking the ⚠️ next to the line numbers in the code viewer.

Files containing selectable code blocks have been marked with ⚠️.

Vulnerability Category

Injection Flaws - SQL Injection

Submit Your Answer

There is 1 vulnerable block in the source code that you need to locate.

You have selected 0 code blocks

Debug Information

Challenge: 5eecdd3a136ea0171cf25eea2

Code Blocks

gradebook/api/views.py:L222-232 - 5eecdd137a569f4e34450a75

gradebook/api/views.py:L214-214 - 5eecdd137a569f4e34450a74

gradebook/api/views.py:L235-241 - 5eecdd137a569f4e34450a72

gradebook/api/urls.py:L13-13 - 5eecdd137a569f4e34450a76

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)
```