

Stack Overflow is a community of 4.7 million programmers, just like you, helping each other.

Join them; it only takes a minute:

Sign up

Join the Stack Overflow community to:



Ask programming questions



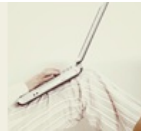
Answer and help your peers



Get recognized for your expertise

check if `.one()` is empty sqlalchemy

Work on work you love. From home.



stackoverflow

I am running a query based off of other ids for the query. The problem i have is that sometimes the query won't find a result. Instead of having the entire program crash, how can I check to see if the result will be None?

This is the query I have:

```
sub_report_id =
DBSession.query(TSubReport.ixSubReport).filter(and_(TSubReport.ixSection==sectionID[0],
TSubReport.ixReport== reportID[0])).one()
```

When the code gets executed and no results are found, I get a `NoResultFound` exception

```
NoResultFound: No row was found for one()
```

is there a way to just skip the query if there is not going to be a result?

Found the solution on SO(couldn't find it before) [Getting first row from sqlalchemy](#)

python sqlalchemy pyramid

asked Jul 27 '14 at 21:48



john
675 ● 1 ● 7 ● 23

3 Answers

Use `first()` function instead of `one()`. It will return None if there is no results.

```
sub_report_id =
DBSession.query(TSubReport.ixSubReport).filter(and_(TSubReport.ixSection==sectionID[0],
TSubReport.ixReport== reportID[0])).first()
```

see documentation [here](#)

answered Jul 27 '14 at 21:52



Lynch
4,253 ● 1 ● 12 ● 26

I ended up using a try catch but this seems more clean – [john](#) Jul 28 '14 at 14:05

1 @john Yes its cleaner. The API throws a exception to warn you that something went wrong. Basically `.one()` should only be used getting only one result is mandatory for the rest of the method. (good example is when you retrieve an element by id and want to perform an operation on it. In this case you know the execution should stop if the element does not exists.) Remember exception thrown by API are supposed to help you, they should get in your way. – [Lynch](#) Jul 28 '14 at 19:58

2 Note that there's a subtle difference between `.one()` and `.first()` - if there are *more than one result*, `.one()` will also fail, while `.first()` will just return the first result and omit the rest. May lead to strange

behaviour when you accidentally get two sub-reports in the same section – [Sergey](#) Jul 28 '14 at 20:35

that's what I was thinking but I funny enough I ended up changing a few other things which let me avoid this situation – [john](#) Jul 29 '14 at 13:25



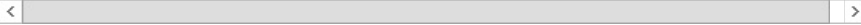
Did you find this question interesting? Try our newsletter

Sign up for our newsletter and get our top new questions delivered to your inbox ([see an example](#)).

How would SQLAlchemy know there wasn't going to be a result without doing the query?

You should catch the exception and handle it then:

```
try:
    sub_report_id =
    DBSession.query(TSubReport.ixSubReport).filter(and_(TSubReport.ixSection==sectionID[0],
    TSubReport.ixReport== reportID[0])).one()
except NoResultFound:
    sub_report_id = [] # or however you need to handle it
```



answered Jul 27 '14 at 21:51



lanAuld

4,626 ● 1 ● 19 ● 35

You can also use `one_or_none()`, this returns `None` when there's no result found and is syntactically clearer than `first()`. No error handling required.

ref: [one_or_none\(\)](#)

answered Dec 4 '15 at 8:59



Yth

56 ● 3