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

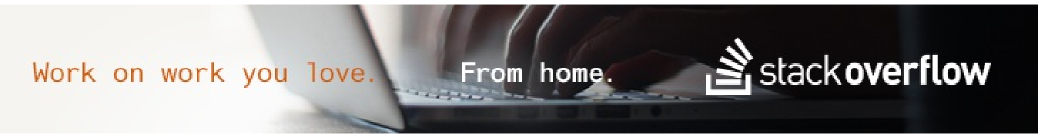
Join the Stack Overflow community to:

Ask programming questions

Answer and help your peers

Get recognized for your expertise

How to elegantly check the existence of an object/instance/variable and simultaneously assign it to variable if it exists in python?



A question for the pythonistas beneath you. If this question makes no sense, please feel free to correct me. I for myself hope, that there is an answer to it, that will make my life easier. :-)

I am using SQLAlchemy to populate a database and often I need to check if a orm object exists in a database before processing. This may be an unconventional question, but I found myself encountering this pattern often:

```
my_object = session.query(SomeObject).filter(some_fiter).first()
if my_object: # Mostly in databases...
    # Juchee it exists
    # process
else:
    # It does not exist. :-(
    my_object = SomeObject()
    # process
```

What I am **dreaming** of would be something like:

```
if my_object = session.query(someObject).blabla.first():
    # if my_object is None this scope is left alone
    # if my_object is not None I can work with my_object here...
```

I know, that this syntax is wrong, but I wanted to explain, what I mean by this example. Any equivalent way would make me happy.

Is there an elegant python approach for this pattern? This question aims not only at SQLAlchemy, but to each equivalent scenario.

closing my eyes hitting "Post your question" and waiting for the smart people and pythonistas by heart to hunt me down for asking something mayhaps inappropriate ;-)

python sqlalchemy

edited Jul 5 '11 at 19:54
Joe Holloway
14.6k ●9 ●57 ●80

asked Jul 5 '11 at 19:33
Aufwind
5,534 ●15 ●57 ●116

1 Explicit is always better than implicit, says Python... and we get this. It frustrated me at first too. So inelegant. Oh well. – arkigos Jul 8 '11 at 7:03

6 Answers

You want to execute a Exist query to be efficient

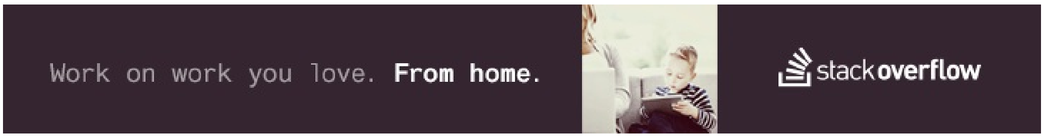
```
(ret, ), = Session.query(exists().where(SomeObject.field==value))
```

Mike Bayer explain it in his blog post:
http://techspot.zzzeek.org/2008/09/09/selecting-booleans/

You can use scalar if you don't want to have a tuple as result:

```
ret = Session.query(exists().where(SomeObject.field==value)).scalar()
```

answered Apr 14 '13 at 19:30
Rach
864 ●9 ●8



This has been asked a long time ago but for future visitors a more concise way to check is

```
if session.query(model).filter(some_filter).count():
    # do stuff
```

answered Oct 24 '12 at 16:46

elbear
492 ● 3 ● 14

2 This looks like exactly what he asked for. – num1 Nov 26 '12 at 21:08

5 `count()` may be concise, but it's also inefficient -- more expensive to check *how many* things match filter X than to check if *any* things match filter X; for the "exists" check, one can declare success when the first match is found, rather than continuing to look for more. – Charles Duffy Sep 18 '15 at 23:51

what is model and what is some_filter? – Steinfeld May 11 at 7:21

wrap it on a function (shamelessly stolen from django get_or_create, this doesnt return a tuple though)

```
get_or_create(model, **kwargs):
    try:
        # basically check the obj from the db, this syntax might be wrong
        object = session.query(model).filter(**kwargs).first()
        return object
    except DoesNotExistException: # or whatever error/exception it is on SQLA
        object = model()
        # do it here if you want to save the obj to the db
        return object
```

that's it. to use it:

```
obj = get_or_create(SomeObject, filters)
```

change the `**kwargs` to a simple argument (like some_filters) if you want

try to wrap something you often use (wrap them to functions or classes)

thats only pseudo code, there might be syntax error.

EDIT: emphasize

answered Jul 5 '11 at 20:04

kusut
1,018 ● 6 ● 19

1 bonus point: try make your function tells you, whether the object it gives you is from the db or newly created (the hint is already there: see django's get_or_create) – kusut Jul 5 '11 at 20:19

I know it's not all one step, but is this acceptable?

```
my_object = session.query(SomeObject).filter(some_filter).first()
if my_object is None:
    my_object = SomeObject()
#process
```

answered Jul 5 '11 at 20:02

multipleinterfaces
3,963 ● 3 ● 17 ● 29

```
from sqlalchemy.orm.util import has_identity

my_object = session.query(SomeObject).get(id) or SomeObject()
# Processing...

# Check if the object exists in the database
if not has_identity(my_object):
    session.add(my_object)

session.commit()
```

.get() can be replaced with a filter() + first() if needed answered Oct 30 '14 at 19:02

Metalstorm
629 ● 8 ● 15

```
if DBSession.query(ObjectType).filter(ObjectType.some_parametter == "This").first()
is None:
```

This is an efficient one line way of checking whether a record exists. It is efficient because it only grabs the first object, and it can be on one line because first() returns None when there are no matching records. Hope that helps!

edited Apr 13 at 10:13

Sean Murphy
3 ● 2

answered Dec 6 '15 at 0:54

artillery129
101 ● 1 ● 4