Previous Next

Putting the pieces together

Estimated Time: 2-3 hours

To access PostgreSQL from Python you'll need the psycopg2 module. To install it run the following command:

```
sudo python3 -m pip install psycopg2
```

At the top of your snippets.py, add `import psycopg2` to make it available to the script. Then, after initializing logging, add the following lines to connect to the database from Python:

```
logging.debug("Connecting to PostgreSQL")
connection = psycopg2.connect(database="snippets")
logging.debug("Database connection established.")
```

Fleshing out the stubs

Now that you are connected to the database it's finally time to replace the FIXME stubs with something functional! You currently have this:

```
def put(name, snippet):
    """Store a snippet with an associated name."""
    logging.error("FIXME: Unimplemented - put({!r}, {!r})".format(name
```

> Update this code to run an `insert into` statement on the database and store a new snippet:

```
def put(name, snippet):
    """Store a snippet with an associated name."""
    logging.info("Storing snippet {!r}: {!r}".format(name, snippet))
    cursor = connection.cursor()
    command = "insert into snippets values (%s, %s)"
    cursor.execute(command, (name, snippet))
    connection.commit()
    logging.debug("Snippet stored successfully.")
    return name, snippet
```

First, you're creating a cursor object by calling the `connection.cursor` method. Cursors allow us to run SQL commands in the Postgres session. Next, you're constructing a string which contains the `insert into` statement, with two placeholders for the keyword and message indicated by `%s`. Then you're running the command on the database using the `cursor.execute` method, substituting in the snippet keyword and message by passing them as a tuple. Finally, you save changes to the database using the `connection.commit` function.

Try adding a new snippet using your put command by running `python3 snippets.py put Test "This is a test snippet"`. Take a look at the log. You should see that the snippet is being stored successfully. Then run a query on your database to make sure that it has been stored.

Get Challenge

Try to implement the `get()` function. It should:

- Construct a `select` statement to retrieve the snippet with a given keyword
- Run the statement on the database

- Fetch the corresponding row from the database
- Return the message from the row

Hint - To fetch the row you can call the *cursor.fetchone* method after executing your select statement. This returns a tuple of values for each field. For the moment you can assume that a snippet will always be correctly selected. You will be dealing with what happens when you try to select a non-existent snippet in the next assignment.

Hint - When you execute an SQL statement, the parameters must always be packaged up into a tuple. This requires a comma: `(word,)` not just `(word)`.

When you have completed the challenge you should be able to store new snippets, and retrieve the stored information. At the moment, your code won't cope well with problems, but you'll get to that in the next assignment.

Before moving on, remember to commit your work to source control.

☆ ☆ ☆ ☆ ☆ · Report a typo or other issue

Mark as completed

Previous Next