Side Effects

You can also create side effects of mock objects via the **side_effect** argument. A side effect is something that happens when you run your function. For example, some videogames have integration into social media. When you score a certain number of points, win a trophy, complete a level or some other predetermined goal, it will record it AND also post about it to Twitter, Facebook or whatever it is integrated with. Another side effect to running a function is that it might be tied to closely with your user interface and cause it to redraw unnecessarily.

Since we know about these kinds of side effect up front, we can mock them in our code. Let's look at a simple example:

```
from unittest.mock import Mock


def my_side_effect():
    print('Updating database!')

def main():
    mock = Mock(side_effect=my_side_effect)
    mock()

if __name__ == '__main__':
    main()
```

Here we create a function that pretends to update a database. Then in our **main** function, we create a mock object and give it a side effect. Finally we call our mock object. If you do this, you should see a message printed to stdout about the database being updated.

The Python documentation also points out that you can make side effect raise an exception if you want to. One fairly common reason to want to raise an exception if you called it incorrectly. An example might be that you didn’t pass in enough arguments. You could also create a mock that raises a Deprecation warning.