#users\_test/test\_users.py

import unittest

from users import get\_users, get\_user

from unittest.mock import patch, Mock

class BasicTests(unittest.TestCase):

@patch('users.get\_users')

def test\_get\_one\_user(self, mock\_get\_users):

"""

Test for getting one user using their userID

Demonstrates mocking third party functions

"""

users = [

{'phone': '514-794-6957', 'first\_name': 'Brant', 'last\_name': 'Mekhi', 'id': 0},

{'phone': '772-370-0117', 'first\_name': 'Thalia', 'last\_name': 'Kenyatta', 'id': 1},

{'phone': '176-290-7637', 'first\_name': 'Destin', 'last\_name': 'Soledad', 'id': 2}

]

mock\_get\_users.return\_value = Mock()

mock\_get\_users.return\_value.json.return\_value = users

user = get\_user(2)

self.assertEqual(user, users[2])

if \_name\_ == "\_main\_":

unittest.main()

[3:29 PM, 8/1/2021] Gopal: const apiBaseUrl = "https://ilive4coding.free.beeceptor.com";

function getUsers(){

const endpoint = "/users";

axios.get(`${apiBaseUrl}${endpoint}`).then(response=>{

const users = response.data.data;

console.log(users);

});

}

function addUser(){

const endpoint = "/user";

axios.post(`${apiBaseUrl}${endpoint}`, {

firstName: 'Fred',

lastName: 'Flintstone'

})

.then(function (response) {

console.log(response);

})

.catch(function (error) {

console.log(error);

});

}

getUsers();

addUser(); #users\_test/test\_users.py

import unittest

from users import get\_users, get\_user

from unittest.mock import patch, Mock

class BasicTests(unittest.TestCase):

@patch('users.get\_users')

def test\_get\_one\_user(self, mock\_get\_users):

"""

Test for getting one user using their userID

Demonstrates mocking third party functions

"""

users = [

{'phone': '514-794-6957', 'first\_name': 'Brant', 'last\_name': 'Mekhi', 'id': 0},

{'phone': '772-370-0117', 'first\_name': 'Thalia', 'last\_name': 'Kenyatta', 'id': 1},

{'phone': '176-290-7637', 'first\_name': 'Destin', 'last\_name': 'Soledad', 'id': 2}

]

mock\_get\_users.return\_value = Mock()

mock\_get\_users.return\_value.json.return\_value = users

user = get\_user(2)

self.assertEqual(user, users[2])

if \_name\_ == "\_main\_":

unittest.main()

const apiBaseUrl = "https://ilive4coding.free.beeceptor.com";

function getUsers(){

const endpoint = "/users";

axios.get(`${apiBaseUrl}${endpoint}`).then(response=>{

const users = response.data.data;

console.log(users);

});

}

function addUser(){

const endpoint = "/user";

axios.post(`${apiBaseUrl}${endpoint}`, {

firstName: 'Fred',

lastName: 'Flintstone'

})

.then(function (response) {

console.log(response);

})

.catch(function (error) {

console.log(error);

});

}

getUsers();

addUser();

#users\_test/test\_users.py

import unittest

from users import get\_users, get\_user

from unittest.mock import patch, Mock

class BasicTests(unittest.TestCase):

@patch('users.get\_users')

def test\_get\_one\_user(self, mock\_get\_users):

"""

Test for getting one user using their userID

Demonstrates mocking third party functions

"""

users = [

{'phone': '514-794-6957', 'first\_name': 'Brant', 'last\_name': 'Mekhi', 'id': 0},

{'phone': '772-370-0117', 'first\_name': 'Thalia', 'last\_name': 'Kenyatta', 'id': 1},

{'phone': '176-290-7637', 'first\_name': 'Destin', 'last\_name': 'Soledad', 'id': 2}

]

mock\_get\_users.return\_value = Mock()

mock\_get\_users.return\_value.json.return\_value = users

user = get\_user(2)

self.assertEqual(user, users[2])

if \_name\_ == "\_main\_":

unittest.main()

[3:29 PM, 8/1/2021] Gopal: const apiBaseUrl = "https://ilive4coding.free.beeceptor.com";

function getUsers(){

const endpoint = "/users";

axios.get(`${apiBaseUrl}${endpoint}`).then(response=>{

const users = response.data.data;

console.log(users);

});

}

function addUser(){

const endpoint = "/user";

axios.post(`${apiBaseUrl}${endpoint}`, {

firstName: 'Fred',

lastName: 'Flintstone'

})

.then(function (response) {

console.log(response);

})

.catch(function (error) {

console.log(error);

});

}

getUsers();

addUser();