Exercises: Testing API

1. Implement the following API endpoint and write a test function for it:

    ```
    POST /api/add

    {"num1":XXX, "num2":YYY}
    ```

    The endpoint should return the sum of num1 and num2: {"result":ZZZ}

    The test function should check that the API returns the correct sum of the arguments.

    Use the requests library to implement the test function.

Solution

```
@app.route('/api/add', methods=['POST'])

def add():

  return {"result":request.json['num1']+request.json['num2']}


def test_add():

  num1=3

  num2=5

  sum=num1+num2

  api_sum=requests.post("http://127.0.0.1:5000/api/sum",{"num1":num1, "num2":num2}).json()["result"]

  return api_sum==sum
```

2. Implement the following API endpoint and write a test function for it:

   POST /api/divide

   {"num1":XXX, "num2":YYY}

   The endpoint should return num1 divided by num2: {"result":ZZZ}

   if num2 is zero, the endpoint should return {"result":"error"}

   The test function should check that the API returns the correct division of the arguments, and also test the case where num2 equals zero.

   Use the requests library to implement the test function.

Solution

```
@app.route('/api/divide', methods=['POST])

def divide():

  try:

    return {"result":request.json['num1'] / request.json['num2']}

  except:
    return {"result":"error"}


def test_divide():
  num1=10
  num2=5
  div=num1/num2
  api_div=requests.post("http://127.0.0.1:5000/api/sum",{"num1":num1, "num2":num2}).json()["result"]
  api_div_zero=requests.post("http://127.0.0.1:5000/api/sum",{"num1":num1, "num2":0}).json()["result"]


  return api_div==div and api_div_zero=="error"
```

3. Implement the following API endpoint and write a test function for it:

   POST /api/login

   {"username":XXX, "password":YYY}

   If the credentials are correct the API should return {"status":"ok"}

   Else, the endpoint should return {"status":"error"}

The test function should check that the API behaves correctly by providing good and bad credentials and testing that the status is correct.

Use the requests library to implement the test function.

4. Implement the following API endpoint and write a test function for it:

   POST /api/login/password

   {"username":XXX, "password":YYY}

   The API should change the password for this user and return {"status":"ok"} or {"status":"error"}.

   The test function should check that the API behaves correctly by checking the relevant scenarios and testing that the status is correct.

   For example, it should test what happens if the user does not exist.

   Use the requests library to implement the test function.