Imagine a banking application. Create an Asp.Net Core Web Application that serves bank account details based on the request path.

Consider the following hard-coded bank account details:

`accountNumber = 1001, accountHolderName = "Example Name", currentBalance = 5000`

You can store these details as an anonymous object. Eg: `new { property1 = value, property2 = value }`

Example #1:

If you receive a HTTP GET request at path "/" (default route), it has to return welcome message with status code HTTP 200.

Request Url: /

Request Method: GET

Response Status Code: 200

Response body (output):

```
Welcome to the Best Bank
```

Example #2:

If you receive a HTTP GET request at path "/account-details", it has to return all the details of bank account as Json format as response with status code HTTP 200.

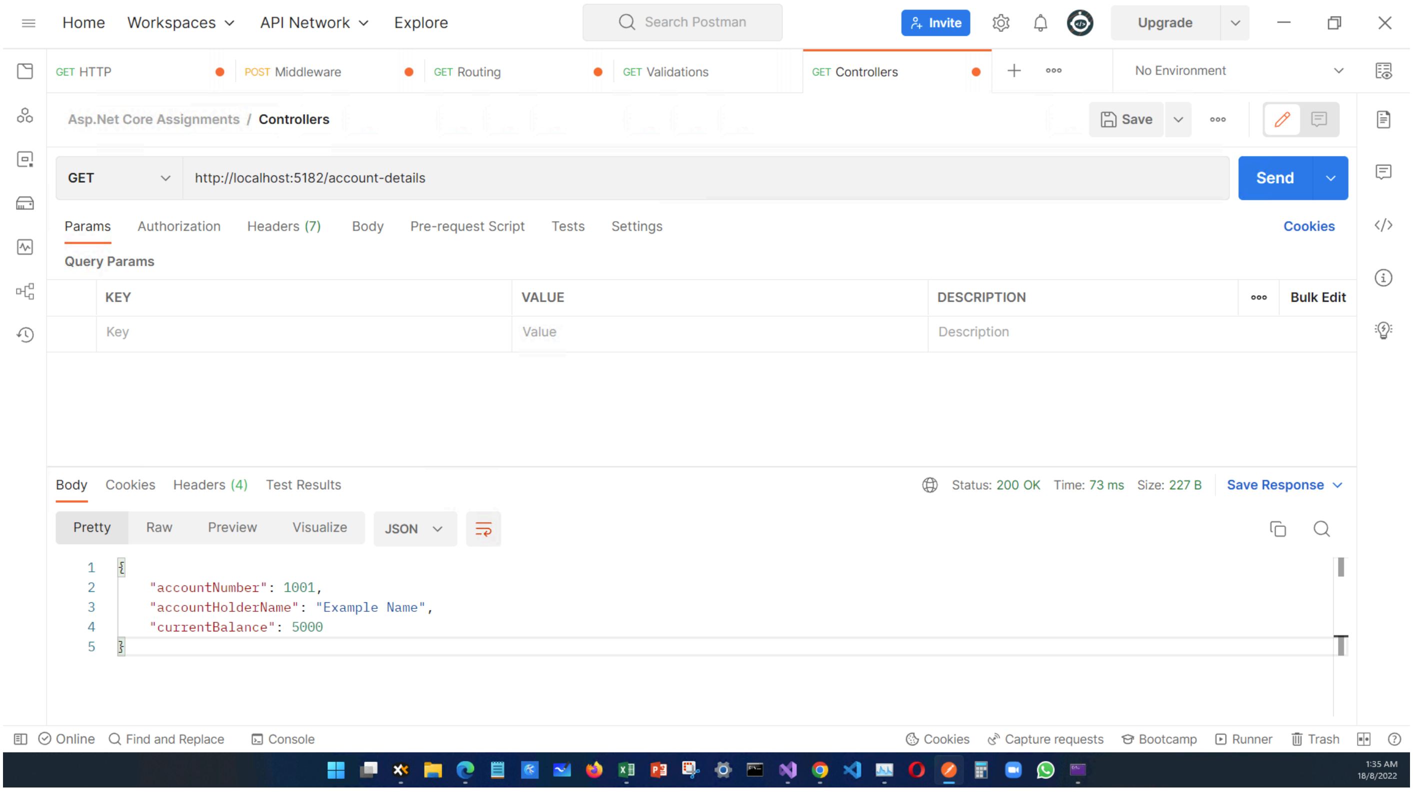
Request Url: /account-details

Request Method: GET

Response Status Code: 200

Response body (output):

```
{
    "accountNumber": 1001,
    "accountHolderName": "Example Name",
    "currentBalance": 5000
}
```



http://localhost:5182/account-details

Status: 200 OK Time: 73 ms Size: 227 B

```
{
    "accountNumber": 1001,
    "accountHolderName": "Example Name",
    "currentBalance": 5000
}
```

Example #3:

If you receive a HTTP GET request at path "/account-statement", it has to return a dummy PDF file (assumed as bank statement) as response with status code HTTP 200.

Request Url: /account-statement

Request Method: GET

Response Status Code: 200

Response body (output):

```
[some dummy PDF file]
```

Example #4:

If you receive a HTTP GET request at path "/get-current-balance/{accountNumber}", it has to return the corresponding current balance value as response with status code HTTP 200.

The "accountNumber" should be an int value and should be equal to "1001".

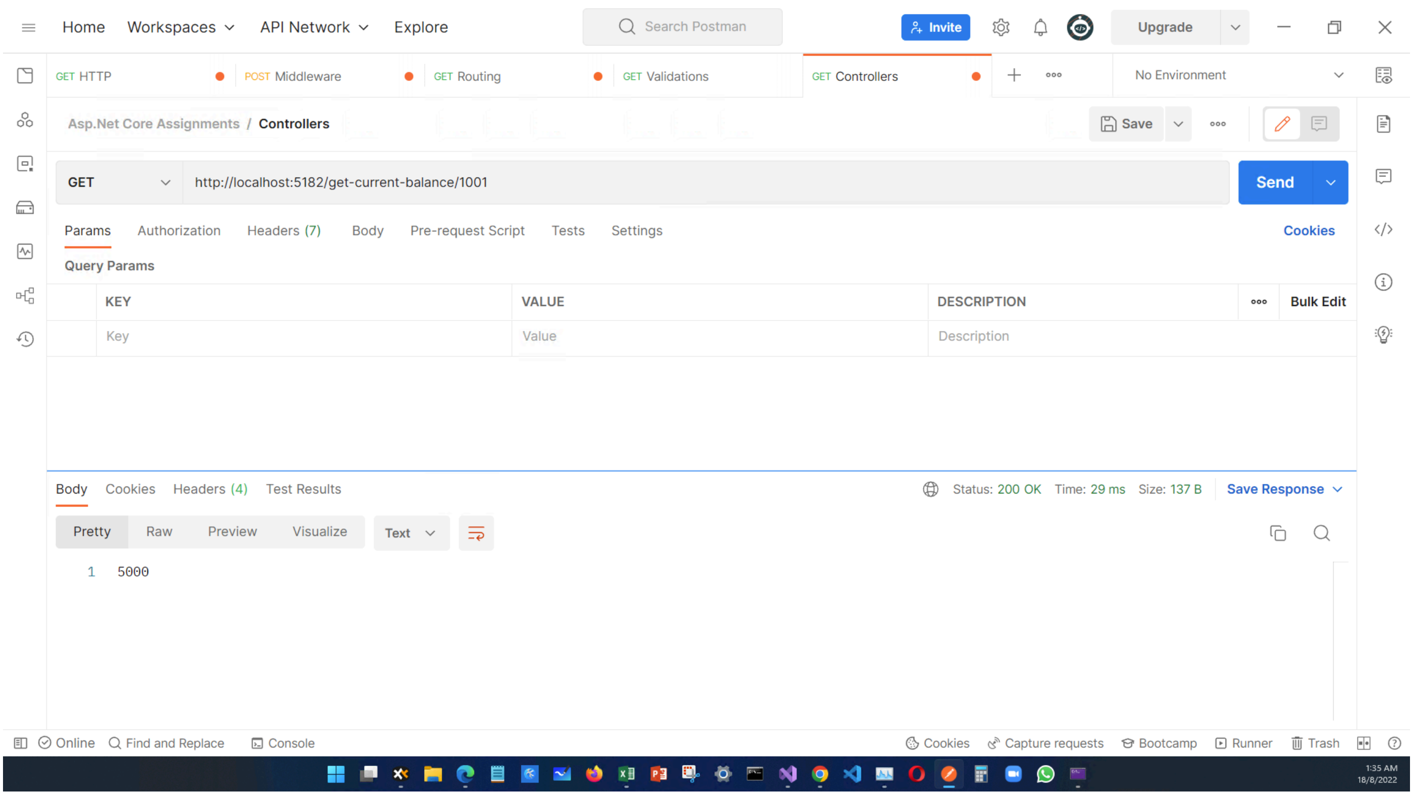
Request Url: /get-current-balance/1001

Request Method: GET

Response Status Code: 200

Response body (output):

5000



GET http://localhost:5182/get-current-balance/1001

Status: 200 OK Time: 29 ms Size: 137 B

```
1  5000
```

Example #5:

If you receive a HTTP GET request at path "/get-current-balance/", if the "accountNumber" is not supplied, it should return HTTP 404 response.

Request Url: /get-current-balance

Request Method: GET

Response Status Code: 404

Response body (output):

```
Account Number should be supplied
```

Example #6:

If you receive a HTTP GET request at path "/get-current-balance/{accountNumber}", if the "accountNumber" is not equal to '1001', it should return HTTP 400 response.

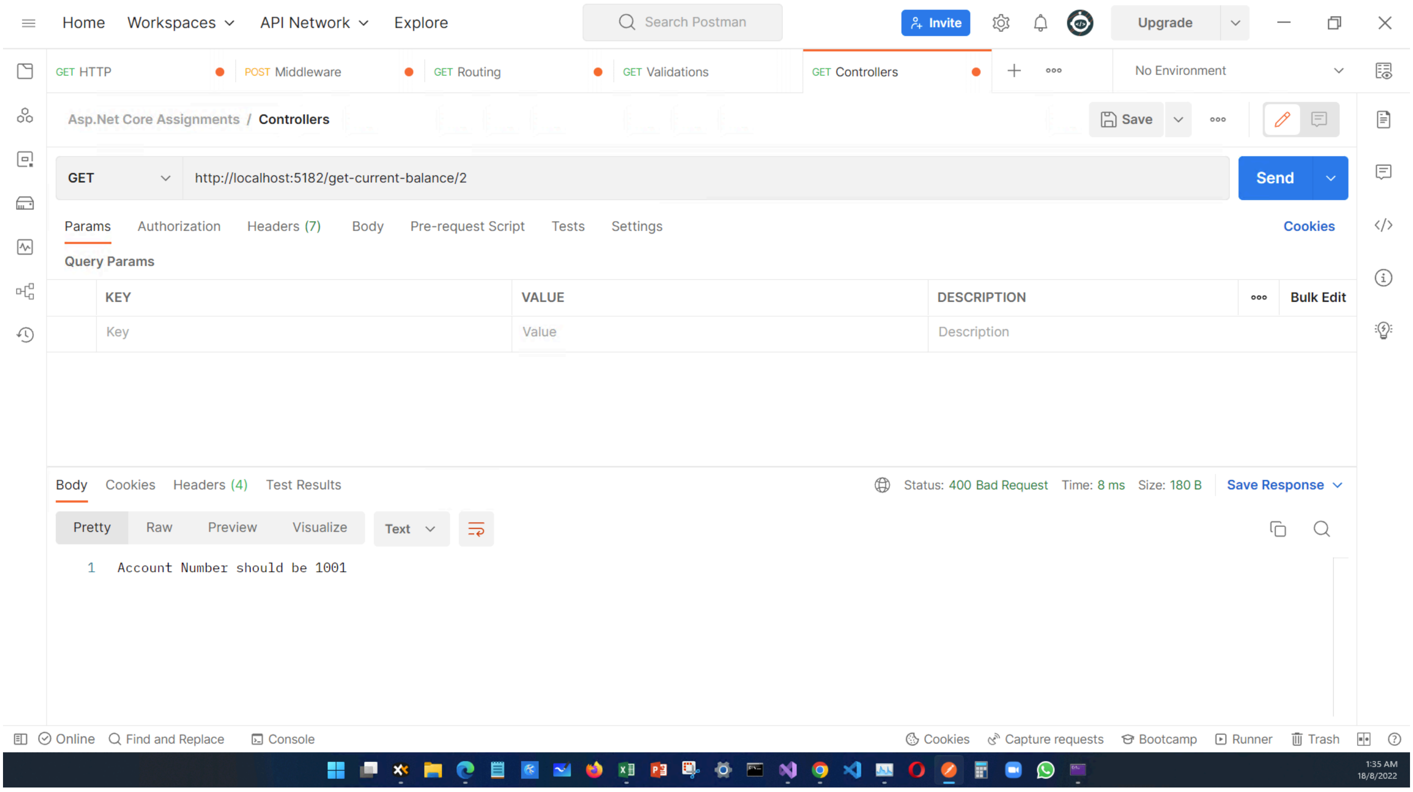
Request Url: /get-current-balance/10

Request Method: GET

Response Status Code: 400

Response body (output):

```
Account Number should be 1001
```



GET http://localhost:5182/get-current-balance/2

Status: 400 Bad Request Time: 8 ms Size: 180 B

```
Account Number should be 1001
```

Route Constraints:

The "accountNumber" parameter should be an int value

Instructions:

- Create controller(s) with attribute routing.
- Use essential route parameters and route constraints.
- Use parameter validation when necessary.
- Return ContentResult, JsonResult, FileResult and other status code results when necessary.
- Return the appropriate HTTP response status code, based on above examples.

Questions for this assignment

Check your source code with Instructor's source code.