

**Assignment:** [Tutorial] Making Manual HTTP Requests

**Due Date:** June 10th by 11:59pm

**Points:** 10

**Description:** In this lesson we will explore HTTP 1.1 requests and responses by creating requests by hand using the netcat linux tool.

**Overall Summary:**

Web Development / Week 1

**Making Manual HTTP Requests**  
from Nina Boonyasiti

Grade: 100, Points (%): 100, Graded: 1, Answered: 1, Assessments: 1

Assessments | Comments | Code Comments

Section	Assessment Name	Points	Type	Correct
Make a POST Request	Body Check	1	A	✓

Date: Jun 13th 2022 10:47pm (UTC -05:00) America/Chicago

Question: Paste the *body* section (and *only* the body section) of your response here.

Student answer: 

```
{ "args": {}, "data": "", "files": {}, "form": { "foo": "2", "bar": "stuff" }, "headers": { "x-forwarded-proto": "http", "x-forwarded-port": "80", "host": "postman-echo.com", "x-amzn-trace-id": "Root=1-62a804a5-348eef1c4c5ebb0f7a41c703", "content-length": "15", "content-type": "application/x-www-form-urlencoded" }, "json": { "foo": "2", "bar": "stuff" }, "url": "http://postman-echo.com/post" }
```

```
$ nc google.com 80
```

We begin the tutorial by asking Linux to run netcat. We use google.com and 80 as the web server and port we want to connect to.

```
GET / HTTP/1.1
```

We request a resource from the server and specify the version of HTTP we are using. We should now get an HTTP response from Google's server. Below is an image of what Google's server response looks like.



```
$ nc postman-echo.com 80
```

Now we will open a connection with netcat through port 80.

```
POST /post HTTP/1.1
Host: postman-echo.com
Content-Type: application/x-www-form-urlencoded
Content-Length: 15
foo=2&bar=stuff
```

We enter these lines in sequential order to run the netcat process and send our request. We will then receive a response like this:

#### 4. Make a POST Request

You should receive a response that looks something like:

```
HTTP/1.1 200 OK
Date: Wed, 03 Jun 2020 18:25:34 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 358
Connection: keep-alive
ETag: W/"166-WyFbG8UhWDh3DVeZERhLomrZo24"
Vary: Accept-Encoding
set-cookie: sails.sid=s%3AhxJUoJy2DIM_9eY8U5aYHLcFS_Bu4-nD.h5X
HNDQAB05d7DaFRaCXgcFhd5wklzuH0N8; Path=/; HttpOnly

[body...]
```

To complete our assignment, we post the body of the POST request response.

The screenshot shows a Codio IDE interface. On the left, a terminal window displays a message: "Sorry but as this assignment is marked as completed, the terminal is no longer accessible. Please contact your teacher if you require access to this assignment again." On the right, a guide window titled "4. Make a POST Request" is open. It features a "Body Check" section with a text area containing a JSON response body: {"args": {}, "data": {}, "files": {}, "form": {"foo": "2", "bar": "stuff"}, "headers": {"x-forwarded-proto": "http", "x-forwarded-port": "80", "host": "postman-echo.com", "x-amzn-trace-id": "Root=1-62a804a5-348eef1c4c5ebb0f7a41c703", "content-length": "15", "content-type": "application/x-www-form-urlencoded"}, "json": {"foo": "2", "bar": "stuff"}, "url": "http://postman-echo.com/post"}. Below the text area is a "Check It!" button. A score of "1 out of 1" is displayed, and the text "Great work!" is shown at the bottom.