



Build your own HTTP server

Learn about TCP servers, the HTTP protocol and more



[Start Building](#)

HTTP is the protocol that powers the web. In this challenge, you'll build a HTTP server that's capable of handling simple GET/POST requests, serving files and handling multiple concurrent connections.

Along the way, we'll learn about TCP connections, HTTP headers, HTTP verbs, handling multiple connections and more.

Stages



Bind to a port

VERY EASY



Respond with 200

VERY EASY



Extract URL path

EASY

- Respond with body EASY
- Read header EASY
- Concurrent connections EASY
- Return a file MEDIUM
- Read request body MEDIUM

HTTP Compression

- Compression headers EASY
- Multiple compression schemes MEDIUM
- Gzip compression MEDIUM

There are few sites I like as much that
“ have a step by step guide. The real-time
feedback is so good, it's creepy!



Ananthalakshmi Sankar

Automation Engineer at Apple

I think the instant feedback right there in
“ the git push is really cool. Didn't even
know that was possible!



Patrick Burris

Senior Software Developer, CenturyLink



Programming challenges for seasoned developers.

**CHALLENGES**

- Git
- Redis
- Shell
- SQLite
- Grep
- BitTorrent
- HTTP Server
- DNS Server

SUPPORT

- Docs
- Status

COMPANY

- About
- Changelog

LEGAL

- Terms
- Privacy

© 2024 CodeCrafters, Inc. All rights reserved.



< back next >



Upgrade



Introduction

✓ Completed

Instructions

Introduction

Welcome to the Build your own HTTP server challenge!

HTTP is the protocol that powers the web. In this challenge, you'll build a HTTP server that's capable of handling simple GET/POST requests, serving files and handling multiple concurrent connections.

Along the way, we'll learn about TCP connections, HTTP headers, HTTP verbs, handling multiple connections and more.

Pre-Challenge Assessment

✓ Completed

Preferred Language: Rust ✓

Language Proficiency: Beginner ✓

Tests failed. Show logs

RECENT ATTEMPTS 11



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



kill-face



adamc...



maxim...



matthe...



tasnuval

1 / 11



< back next >



Upgrade



Repository Setup Completed

Instructions

Repository Setup Completed

We've prepared a starter repository with some Rust code for you.

Clone the repository

command line

```
git clone https://git.codecrafters.io/da327dc0945696e2 codecrafters-http-server-rust
cd codecrafters-http-server-rust
```

copy

Push an empty commit

command line

```
git commit --allow-empty -m 'test'
git push origin master
```

copy

RECENT ATTEMPTS 11



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



kill-face



adamc...



maxim...



matthe...



tasnuval

1 / 11

Tests failed. Show logs



< back next >



Upgrade



Repository Setup

✓ Completed

Instructions

Repository Setup

✓ Completed

We've prepared a starter repository with some Rust code for you.

Clone the repository

command line

copy

```
git clone https://git.codecrafters.io/da327dc0945696e2 codecrafters-http-server-rust  
cd codecrafters-http-server-rust
```

Push an empty commit

command line

copy

```
git commit --allow-empty -m 'test'  
git push origin master
```

Tests failed. Show logs

RECENT ATTEMPTS 11



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



kill-face



adamc...



maxim...



matthe...



tasnuval

1 / 11



< back next >



Upgrade



Bind to a port #AT4 Completed

Instructions



You completed this stage 7 months ago.

[View next stage](#)

You just earned the **Turing** Badge! Read the story.

How to pass this stage

Since this is the first stage, we've included some commented code to help you get started. To pass this stage, simply uncomment the code and submit your changes.

Step 1: Navigate to `src/main.rs` ✓

Step 2: Uncomment code ✓

Step 3: Submit changes ✓

RECENT ATTEMPTS 11



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



kill-face



adamc...



maxim...



matthe...



tasnuval

1 / 11

Tests failed. [Show logs](#)

 Instructions

Stage #AT4

 CompletedStep 3: Submit changes 

Your Task

 Completed

Share Feedback

VERY EASY

In this stage, you'll create a TCP server that listens on port 4221.

[TCP](#) is the underlying protocol used by HTTP servers.

Tests

The tester will execute your program like this:

```
$ ./your_program.sh
```

Then, the tester will try to connect to your server on port 4221. The connection must succeed for you to pass this stage.

[View Next Stage →](#)[Collapse ↑](#)Tests failed. [Show logs](#)

RECENT ATTEMPTS

11



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



kill-face



adamc...



maxim...



matthe...



tasnuval

1 / 11



Invite a friend



Private

leaderboards are

 Instructions

Stage #AT4

 Completed Collapse ↑

Hints

Filter by Rust 

Write

Preview

Found an interesting resource? Share it with the community.

 Markdown supported. Comment**stefanscheidt** 1 year ago

To try this locally on macOS, you could run `./your_server.sh` in one terminal session, and `nc -vz 127.0.0.1 4221` in another. (`-v` gives more verbose output, `-z` just scan for listening daemons, without sending any data to them.)

 428 replyTests failed. RECENT ATTEMPTS 

alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



kill-face



adamc...



maxim...



matthe...



tasnuval

1 / 11



Invite a friend



Private

leaderboards are
here



< back next >



Upgrade



Respond with 200 #IA4

In-progress

Instructions

Code Examples

Screencasts

Concepts

Forum

How to pass this stage

In this stage, you'll implement your own solution. Unlike stage 1, your repository doesn't contain commented code to pass this stage.

98% of users who attempt this stage complete it.

Step 1: Read instructions

Step 2: Implement solution

Step 3: Run tests

Tests failed. [Click here](#) to view logs.

To run tests again, make changes to your code and run the test command:

```
codecrafters cli git
```

copy

```
codecrafters test # Visit https://codecrafters.io/cli to install
```

RECENT ATTEMPTS



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



kill-face



adamc...



maxim...



matthe...



tasnuval

1 / 11

Tests failed. [Show logs](#)

[Instructions](#)[Code Examples](#)[Screencasts](#)[Concepts](#)[Forum](#)

Stage #IA4

In-progress

Your Task

In-progress

[Share Feedback](#)

VERY EASY

In this stage, your server will respond to an HTTP request with a `200` response.

HTTP response

An HTTP response is made up of three parts, each separated by a [CRLF](#) (`\r\n`):

1. Status line.
2. Zero or more headers, each ending with a CRLF.
3. Optional response body.

In this stage, your server's response will only contain a status line. Here's the response your server must send:

`HTTP/1.1 200 OK\r\n\r\n`

Here's a breakdown of the response:

```
// Status line
HTTP/1.1 // HTTP version
200 // Status code
```

Tests failed.

[Show logs](#)**RECENT ATTEMPTS**

11



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



kill-face



adamc...



maxim...



matthe...



tasnuval

1/11



Invite a friend



Private

X

leaderboards are

[Instructions](#)[Code Examples](#)[Screencasts](#)[Concepts](#)[Forum](#)

Stage #IA4

In-progress

```
// Status line
HTTP/1.1 // HTTP version
200 // Status code
OK // Optional reason phrase
\r\n // CRLF that marks the end of the status line

// Headers (empty)
\r\n // CRLF that marks the end of the headers

// Response body (empty)
```

Tests

The tester will execute your program like this:

```
$ ./your_program.sh
```

The tester will then send an HTTP GET request to your server:

```
$ curl -v http://localhost:4221
```

Your server must respond to the request with the following response:

```
HTTP/1.1 200 OK\r\n\r\n
```

RECENT ATTEMPTS 11



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



kill-face



adamc...



maxim...



matthe...



tasnuval

1/11



Invite a friend



Private

X

leaderboards are

Tests failed.

Show logs

[Instructions](#)[Code Examples](#)[Screencasts](#)[Concepts](#)[Forum](#)

Stage #IA4

In-progress

Your server must respond to the request with the following response:

HTTP/1.1 200 OK\r\n\r\n

Notes

- You can ignore the contents of the request. We'll cover parsing requests in later stages.
- For more information about HTTP responses, see the [MDN Web Docs on HTTP responses](#) or the [HTTP/1.1 specification](#).
- This challenge uses HTTP/1.1.

[View Code Examples](#)[View Screencasts](#)[Collapse ↑](#)

Hints

Filter by Rust

Tests failed. [Show logs](#)

RECENT ATTEMPTS



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



kill-face



adamc...



maxim...



matthe...



tasnuval

1 / 11



Invite a friend



Private



leaderboards are

 Instructions

Stage #AT4

 Completed

Expand ↓

Hints

Filter by Rust 

Write

Preview

Found an interesting resource? Share it with the community.

 Markdown supported. Comment**stefanscheidt** 1 year ago

To try this locally on macOS, you could run `./your_server.sh` in one terminal session, and `nc -vz 127.0.0.1 4221` in another. (`-v` gives more verbose output, `-z` just scan for listening daemons, without sending any data to them.)

 428 reply

Tests failed.

 Show logsRECENT ATTEMPTS 

alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



adamc...



kill-face



maxim...



matthe...



tasnuval

1 / 11



Invite a friend



Private

leaderboards are
here



< back next >



Upgrade



Extract URL path #IH0 Pending

Complete previous stages to gain access to this stage.

[Instructions](#)[Code Examples](#)[Screencasts](#)[Concepts](#)[Forum](#)

Unlock all stages with a CodeCrafters membership. No limits.

Plans start at \$30/month \$15/month (discounted price for Bangladesh) when billed annually.

Want CodeCrafters for free? You might be eligible to [claim the membership fees](#) back from your employer. Or you can [refer friends](#) to earn free weeks.

[View Membership Plans →](#)[No limits on content](#)[Turbo test runs](#)[Code examples](#)[Dark mode](#)[Anonymous mode](#)

RECENT ATTEMPTS 11



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



kill-face



adamc...



maxim...



matthe...

Tests failed. [Show logs](#)

[Instructions](#)[Code Examples](#)[Screencasts](#)[Concepts](#)[Forum](#)

Stage #IHO

Pending

Your Task

Pending

[Share Feedback](#)

EASY

In this stage, your server will extract the URL path from an HTTP request, and respond with either a `200` or `404`, depending on the path.

HTTP request

An HTTP request is made up of three parts, each separated by a [CRLF](#) (`\r\n`):

1. Request line.
2. Zero or more headers, each ending with a CRLF.
3. Optional request body.

Here's an example of an HTTP request:

```
GET /index.html HTTP/1.1\r\nHost: localhost:4221\r\nUser-Agent: curl/7.1
```

Here's a breakdown of the request:

```
// Request line
GET                                // HTTP method
/index.html                         // Request target
```

Tests failed.

[Show logs](#)

RECENT ATTEMPTS

11



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



adamc...



kill-face



maxim...



matthe...



tasnuval

1/11



Invite a friend



Private

leaderboards are

[Instructions](#)[Code Examples](#)[Screencasts](#)[Concepts](#)[Forum](#)

Stage #IHO

Pending

Here's a breakdown of the request:

```
// Request line
GET                                // HTTP method
/index.html                         // Request target
HTTP/1.1                            // HTTP version
\r\n                                // CRLF that marks the end of the request

// Headers
Host: localhost:4221\r\n           // Header that specifies the server's host
User-Agent: curl/7.64.1\r\n         // Header that describes the client's user
Accept: */*\r\n                   // Header that specifies which media types
\r\n                                // CRLF that marks the end of the headers

// Request body (empty)
```

The "request target" specifies the URL path for this request. In this example, the URL path is `/index.html`.

Note that each header ends in a CRLF, and the entire header section also ends in a CRLF.

Tests

The tester will execute your program like this:

```
$ ./your_program.sh
```

Tests failed. [Show logs](#)

RECENT ATTEMPTS



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



adamc...



kill-face



maxim...



matthe...



tasnuval

1/11



Invite a friend



Private

X

leaderboards are

[Instructions](#)[Code Examples](#)[Screencasts](#)[Concepts](#)[Forum](#)

Stage #IHO

Pending

Here's an example of an HTTP request:

```
nHost: localhost:4221\r\nUser-Agent: curl/7.64.1\r\nAccept: */*\r\n\r\n
```

Here's a breakdown of the request:

```
// HTTP method
// Request target
// HTTP version
// CRLF that marks the end of the request line

\r\n      // Header that specifies the server's host and port
1\r\n      // Header that describes the client's user agent
      // Header that specifies which media types the client can accept
      // CRLF that marks the end of the headers
```

)

The "request target" specifies the URL path for this request. In this example, the URL path is `/index.html`.

Note that each header ends in a CRLF, and the entire header section also ends in a CRLF.

Tests failed.

Show logs

RECENT ATTEMPTS



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



adamc...



matthe...



gurugu...



maness...



tasnuval

1 / 11



Invite a friend



Private

leaderboards are

[Instructions](#)[Code Examples](#)[Screencasts](#)[Concepts](#)[Forum](#)

Stage #IHO

Pending

Tests

The tester will execute your program like this:

```
$ ./your_program.sh
```

The tester will then send two HTTP requests to your server.

First, the tester will send a `GET` request, with a random string as the path:

```
$ curl -v http://localhost:4221/abcdefg
```

Your server must respond to this request with a `404` response:

```
HTTP/1.1 404 Not Found\r\n\r\n
```

Then, the tester will send a `GET` request, with the path `/`:

```
$ curl -v http://localhost:4221
```

Your server must respond to this request with a `200` response:

```
HTTP/1.1 200 OK\r\n\r\n
```

RECENT ATTEMPTS



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



adamc...



kill-face



maxim...



matthe...



tasnuval

1/11



Invite a friend



Private

X

leaderboards are

Tests failed.

Show logs

[Instructions](#)[Code Examples](#)[Screencasts](#)[Concepts](#)[Forum](#)

Stage #IH0

Pending

Your server must respond to this request with a `200` response:

```
HTTP/1.1 200 OK\r\n\r\n\r\n
```

Notes

- You can ignore the headers for now. You'll learn about parsing headers in a later stage.
- In this stage, the request target is written as a URL path. But the request target actually has [four possible formats](#). The URL path format is called the "origin form," and it's the most commonly used format. The other formats are used for more niche scenarios, like sending a request through a proxy.
- For more information about HTTP requests, see the [MDN Web Docs on HTTP requests](#) or the [HTTP/1.1 specification](#).

[View Code Examples](#)[View Screencasts](#)[Collapse ↑](#)

Tests failed.

[Show logs](#)

RECENT ATTEMPTS



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



adamc...



kill-face



maxim...



matthe...



tasnuval

1 / 11



Invite a friend



Private

leaderboards are

[Instructions](#)[Code Examples](#)[Screencasts](#)[Concepts](#)[Forum](#)

Stage #IHO

Pending

My suggestion is to use `curl` to send custom http requests and debug locally. Eg:
`curl -i -X GET http://localhost:4221/index.html`

[^ 137](#) [▼](#) [↳ reply](#)**LakshyaMittal3301** 9 months ago

It's really helpful to use curl. I was stuck for some time, then realized that I needed to close the socket after the server received the data for curl to work. Also, you can do `curl -i localhost:4221` to print the response you receive.

[^ 14](#) [▼](#)**kotharePrajwal** 1 year ago

For stage 1 and 2 you can refer to this explanation by arpit bhiyani

https://youtu.be/f9gUFy-9uCM?si=yDVsUJfdQ8o_OE6

[^ 57](#) [▼](#) [↳ reply](#)**sherubthakur** 1 year ago

The HTTP request lacks an explicit End-of-File (EOF) marker. Exercise caution when reading from the stream and avoid relying on functions that expect an EOF signal.

[^ 52](#) [▼](#) [↳ reply](#)**abbygriffiths** 5 months ago

Thanks for the heads up. This was what was tripping me up the first time I attempted to solve the challenge.

[Tests failed.](#) [Show logs](#)**RECENT ATTEMPTS**

11



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



adamc...



kill-face



maxim...



matthe...



tasnuval

1 / 11



Invite a friend



Private

leaderboards are

[Instructions](#)[Code Examples](#)[Screencasts](#)[Concepts](#)[Forum](#)

Stage #IH0

Pending

It's really helpful to use curl. I was stuck for some time, then realized that

needed to close the socket after the server received the data for curl to work.

Also, you can do `curl -i localhost:4221` to print the response you receive.

[^ 14](#)[▼](#)**kotharePrajwal** 1 year ago

For stage 1 and 2 you can refer to this explanation by arpit bhiyani

https://youtu.be/f9gUFy-9uCM?si=yDVrSUJfdQ8o_OE6

[^ 57](#)[▼](#)[reply](#)**sherubthakur** 1 year ago

The HTTP request lacks an explicit End-of-File (EOF) marker. Exercise caution when reading from the stream and avoid relying on functions that expect an EOF signal.

[^ 52](#)[▼](#)[reply](#)**abbygriffiths** 5 months ago

Thanks for the heads up. This was what was tripping me up the first time I attempted to solve the challenge.

[^ 8](#)[▼](#)**RECENT ATTEMPTS**

alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



adamc...



kill-face



maxim...



matthe...



tasnuval

1 / 11



Private

leaderboards are here.

[Create one for your](#)

Tests failed.

[Show logs](#)



< back next >



Upgrade



Respond with body #CN2

Pending

Complete previous stages to gain access to this stage.

[Instructions](#)[Code Examples](#)[Screencasts](#)[Concepts](#)[Forum](#)

Unlock all stages with a CodeCrafters membership. No limits.

Plans start at \$30/month \$15/month (discounted price for Bangladesh) when billed annually.

Want CodeCrafters for free? You might be eligible to [claim the membership fees](#) back from your employer. Or you can [refer friends](#) to earn free weeks.

[View Membership Plans →](#)[No limits on content](#)[Turbo test runs](#)[Code examples](#)[Dark mode](#)[Anonymous mode](#)

Tests failed.

[Show logs](#)

RECENT ATTEMPTS 11



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



adamc...



jsec



gurugu...



maness...

[Instructions](#)[Code Examples](#)[Screencasts](#)[Concepts](#)[Forum](#)

Stage #CN2

Pending

Your Task

Pending

[Share Feedback](#)

EASY

In this stage, you'll implement the `/echo/{str}` endpoint, which accepts a string and returns it in the response body.

Response body

A response body is used to return content to the client. This content may be an entire web page, a file, a string, or anything else that can be represented with bytes.

Your `/echo/{str}` endpoint must return a `200` response, with the response body set to given string, and with a `Content-Type` and `Content-Length` header.

Here's an example of an `/echo/{str}` request:

```
GET /echo/abc HTTP/1.1\r\nHost: localhost:4221\r\nUser-Agent: curl/7.64
```

And here's the expected response:

```
HTTP/1.1 200 OK\r\nContent-Type: text/plain\r\nContent-Length: 3\r\n\r\nabc
```

RECENT ATTEMPTS

11



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



adamc...



maxim...



matthe...



maness...



tasnuval

1/11



Invite a friend



Private

leaderboards are

Tests failed.

[Show logs](#)

[Instructions](#)[Code Examples](#)[Screencasts](#)[Concepts](#)[Forum](#)

Stage #CN2

Pending

And here's the expected response:

```
HTTP/1.1 200 OK\r\nContent-Type: text/plain\r\nContent-Length: 3\r\n\r\n
```

Here's a breakdown of the response:

```
// Status line
HTTP/1.1 200 OK
\r\n                                // CRLF that marks the end of the status

// Headers
Content-Type: text/plain\r\n    // Header that specifies the format of the
Content-Length: 3\r\n            // Header that specifies the size of the
\r\n                                // CRLF that marks the end of the headers

// Response body
abc                                // The string from the request
```

The two headers are required for the client to be able to parse the response body. Note that each header ends in a CRLF, and the entire header section also ends in a CRLF.

Tests

The tester will execute your program like this:

Tests failed. [Show logs](#)

RECENT ATTEMPTS



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



adamc...



maxim...



matthe...



maness...



tasnuval

1/11



Invite a friend



Private

X

leaderboards are

[Instructions](#)[Code Examples](#)[Screencasts](#)[Concepts](#)[Forum](#)

Stage #CN2

Pending

Here's an example of an `/echo/{str}` request:

```
.nHost: localhost:4221\r\nUser-Agent: curl/7.64.1\r\nAccept: */*\r\n\r\n
```

And here's the expected response:

```
/1.1 200 OK\r\nContent-Type: text/plain\r\nContent-Length: 3\r\n\r\nabc
```

Here's a breakdown of the response:

```
// CRLF that marks the end of the status line\r\n\r\n// Header that specifies the format of the response body\r\n// Header that specifies the size of the response body, in bytes\r\n// CRLF that marks the end of the headers\r\n\r\n
```

```
// The string from the request
```

The two headers are required for the client to be able to parse the response

Tests failed.

Show logs

RECENT ATTEMPTS



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



adamc...



matthe...



gurugu...



maness...



tasnuval

1/11



Invite a friend



Private

X

leaderboards are

[Instructions](#)[Code Examples](#)[Screencasts](#)[Concepts](#)[Forum](#)

Stage #CN2

Pending

Tests

The tester will execute your program like this:

```
$ ./your_program.sh
```

The tester will then send a `GET` request to the `/echo/{str}` endpoint on your server, with some random string.

```
$ curl -v http://localhost:4221/echo/abc
```

Your server must respond with a `200` response that contains the following parts:

- `Content-Type` header set to `text/plain`.
- `Content-Length` header set to the length of the given string.
- Response body set to the given string.

`HTTP/1.1 200 OK\r\nContent-Type: text/plain\r\nContent-Length: 3\r\n\r\nabc`

Notes

- For more information about HTTP responses, see the [MDN Web Docs](#)

Tests failed.

Show logs

RECENT ATTEMPTS



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



adamc...



maxim...



matthe...



maness...



tasnuval

1 / 11



Invite a friend



Private

X

leaderboards are

[Instructions](#)[Code Examples](#)[Screencasts](#)[Concepts](#)[Forum](#)

Stage #CN2

Pending

- Content-Length header set to the length of the given string.
- Response body set to the given string.

HTTP/1.1 200 OK\r\nContent-Type: text/plain\r\nContent-Length: 3\r\n\r\nr\r\n

Notes

- For more information about HTTP responses, see the [MDN Web Docs on HTTP responses](#) or the [HTTP/1.1 specification](#).

[View Code Examples](#)[View Screencasts](#)[Collapse ↑](#)

Hints

Filter by Rust 

Write

Preview

Found an interesting resource? Share it with the community.

Tests failed.

Show logs

RECENT ATTEMPTS



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



adamc...



maxim...



matthe...



maness...



tasnuval

1 / 11



Invite a friend



Private

leaderboards are

[Instructions](#)[Code Examples](#)[Screencasts](#)[Concepts](#)[Forum](#)

Stage #CN2

Pending

The tester will then send a `GET` request to the `/echo/{str}` endpoint on your server, with some random string.

```
$ curl -v http://localhost:4221/echo/abc
```

Your server must respond with a `200` response that contains the following parts:

- `Content-Type` header set to `text/plain`.
- `Content-Length` header set to the length of the given string.
- Response body set to the given string.

```
'/1.1 200 OK\r\nContent-Type: text/plain\r\nContent-Length: 3\r\n\r\nabc
```



Notes

- For more information about HTTP responses, see the [MDN Web Docs on HTTP responses](#) or the [HTTP/1.1 specification](#).

[View Code Examples](#)[View Screencasts](#)[Collapse](#)

Tests failed.

[Show logs](#)

RECENT ATTEMPTS



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



adamc...



matthe...



gurugu...



maness...



tasnuval

1/11



Invite a friend



Private



leaderboards are

[Instructions](#)[Code Examples](#)[Screencasts](#)[Concepts](#)[Forum](#)

Stage #CN2

Pending

Here's an example of an `/echo/{str}` request:

```
.nHost: localhost:4221\r\nUser-Agent: curl/7.64.1\r\nAccept: */*\r\n\r\n
```

And here's the expected response:

```
/1.1 200 OK\r\nContent-Type: text/plain\r\nContent-Length: 3\r\n\r\nabc
```

Here's a breakdown of the response:

```
// CRLF that marks the end of the status line
\r\n // Header that specifies the format of the response body
 // Header that specifies the size of the response body, in bytes
 // CRLF that marks the end of the headers
```

```
// The string from the request
```

The two headers are required for the client to be able to parse the response

Tests failed.

Show logs

RECENT ATTEMPTS 11



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



adamc...



matthe...



gurugu...



maness...



tasnuval

1/11



Invite a friend



Private

X

leaderboards are



< back next >



Upgrade



Read header #FS3 Pending

Complete previous stages to gain access to this stage.

[Instructions](#)[Code Examples](#)[Screencasts](#)[Concepts](#)[Forum](#)

Unlock all stages with a CodeCrafters membership. No limits.

Plans start at \$30/month \$15/month (discounted price for Bangladesh) when billed annually.

Want CodeCrafters for free? You might be eligible to [claim the membership fees](#) back from your employer. Or you can [refer friends](#) to earn free weeks.

[View Membership Plans →](#)[No limits on content](#)[Turbo test runs](#)[Code examples](#)[Dark mode](#)[Anonymous mode](#)

RECENT ATTEMPTS 11



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



adamc...



maxim...



matthe...



maness...

Tests failed. [Show logs](#)

[Instructions](#)[Code Examples](#)[Screencasts](#)[Concepts](#)[Forum](#)

Stage #FS3

Pending

RECENT ATTEMPTS 11

alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



adamc...



matthe...



gurugu...



maness...



tasnuval

1 / 11



Invite a friend



Private

X

leaderboards are

Your Task

Pending

[Share Feedback](#)

EASY

In this stage, you'll implement the `/user-agent` endpoint, which reads the `User-Agent` request header and returns it in the response body.

The User-Agent header

The [User-Agent](#) header describes the client's user agent.

Your `/user-agent` endpoint must read the `User-Agent` header, and return it in your response body. Here's an example of a `/user-agent` request:

```
// Request line
GET
/user-agent
HTTP/1.1
\r\n

// Headers
Host: localhost:4221\r\n
User-Agent: foobar/1.2.3\r\n // Read this value
Accept: */*\r\n
\r\n
```

Tests failed.

[Show logs](#)

[Instructions](#)[Code Examples](#)[Screencasts](#)[Concepts](#)[Forum](#)

Stage #FS3

Pending

Accept: */*\r\n

\r\n

// Request body (empty)

Here is the expected response:

```
// Status line  
HTTP/1.1 200 OK // Status code must be 200
```

\r\n

// Headers

Content-Type: text/plain\r\n

Content-Length: 12\r\n

\r\n

// Response body

foobar/1.2.3 // The value of `User-Agent`

Tests

The tester will execute your program like this:

\$./your_program.sh

The tester will then send a GET request to the /user-agent endpoint on your server. The request will have a User-Agent header

Tests failed.

Show logs

RECENT ATTEMPTS



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



adamc...



matthe...



gurugu...



maness...



tasnuval

1/11



Invite a friend



Private

leaderboards are

[Instructions](#)[Code Examples](#)[Screencasts](#)[Concepts](#)[Forum](#)

Stage #FS3

Pending

The tester will then send a `GET` request to the `/user-agent` endpoint on your server. The request will have a `User-Agent` header.

```
$ curl -v --header "User-Agent: foobar/1.2.3" http://localhost:4221/use
```

Your server must respond with a `200` response that contains the following parts:

- `Content-Type` header set to `text/plain`.
- `Content-Length` header set to the length of the `User-Agent` value.
- Message body set to the `User-Agent` value.

```
HTTP/1.1 200 OK\r\nContent-Type: text/plain\r\nContent-Length: 12\r\n\r\n
```

Notes

- Header names are [case-insensitive](#).

[View Code Examples](#)[View Screencasts](#)[Collapse ↑](#)

Tests failed.

[Show logs](#)

RECENT ATTEMPTS



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



adamc...



matthe...



gurugu...



maness...



tasnuval

1 / 11



Invite a friend



Private

leaderboards are

[Instructions](#)[Code Examples](#)[Screencasts](#)[Concepts](#)[Forum](#)

Stage #FS3

Pending

The tester will then send a `GET` request to the `/user-agent` endpoint on your server. The request will have a `User-Agent` header.

```
-v --header "User-Agent: foobar/1.2.3" http://localhost:4221/user-agent
```

Your server must respond with a `200` response that contains the following parts:

- `Content-Type` header set to `text/plain`.
- `Content-Length` header set to the length of the `User-Agent` value.
- Message body set to the `User-Agent` value.

```
!K\r\nContent-Type: text/plain\r\nContent-Length: 12\r\n\r\nfoobar/1.2.3
```

Notes

- Header names are [case-insensitive](#).

[View Code Examples](#)[View Screencasts](#)[Collapse ↑](#)Tests failed. [Show logs](#)

RECENT ATTEMPTS



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



adamc...



matthe...



gurugu...



maness...



tasnuval

1 / 11



Invite a friend



Private

leaderboards are

[Instructions](#)[Code Examples](#)[Screencasts](#)[Concepts](#)[Forum](#)

Stage #FS3

Pending

[Collapse ↑](#)

Hints

[Filter by Rust](#)[Write](#)[Preview](#)

Found an interesting resource? Share it with the community.

Markdown supported.

[Comment](#)**jzwood** 1 year ago

test with `curl --verbose 127.0.0.1:4221/user-agent`

[^ 63](#) [reply](#)

RECENT ATTEMPTS



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



adamc...



matthe...



gurugu...



maness...



tasnuval

1 / 11

[Invite a friend](#)

Private

leaderboards are here.



Tests failed.

[Show logs](#)



< back next >



Upgrade



Concurrent connections #EJ5 Pending

Complete previous stages to gain access to this stage.

[Instructions](#)[Code Examples](#)[Screencasts](#)[Concepts](#)[Forum](#)

Unlock all stages with a CodeCrafters membership. No limits.

Plans start at \$30/month \$15/month (discounted price for Bangladesh) when billed annually.

Want CodeCrafters for free? You might be eligible to [claim the membership fees](#) back from your employer. Or you can [refer friends](#) to earn free weeks.

[View Membership Plans →](#)[No limits on content](#)[Turbo test runs](#)[Code examples](#)[Dark mode](#)[Anonymous mode](#)

Tests failed. [Show logs](#)

RECENT ATTEMPTS 11



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



adamc...



matthe...



gurugu...



maness...

[Instructions](#)[Code Examples](#)[Screencasts](#)[Concepts](#)[Forum](#)

Stage #EJ5

Pending

Your Task

Pending

[Share Feedback](#)

EASY

In this stage, you'll add support for concurrent connections.

Tests

The tester will execute your program like this:

```
$ ./your_program.sh
```

Then, the tester will create multiple concurrent TCP connections to your server. (The exact number of connections is determined at random.) After that, the tester will send a single `GET` request through each of the connections.

```
$ (sleep 3 && printf "GET / HTTP/1.1\r\n\r\n") | nc localhost 4221 &
$ (sleep 3 && printf "GET / HTTP/1.1\r\n\r\n") | nc localhost 4221 &
$ (sleep 3 && printf "GET / HTTP/1.1\r\n\r\n") | nc localhost 4221 &
```

Your server must respond to each request with the following response:

```
HTTP/1.1 200 OK\r\n\r\n
```

RECENT ATTEMPTS

11



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



adamc...



matthe...



gurugu...



maness...



tasnuval

1/11



Invite a friend



Private

X

leaderboards are

Tests failed.

[Show logs](#)

[Instructions](#)[Code Examples](#)[Screencasts](#)[Concepts](#)[Forum](#)

Stage #EJ5

Pending

In this stage, you will add support for concurrent connections.

Tests

The tester will execute your program like this:

```
$ ./your_program.sh
```

Then, the tester will create multiple concurrent TCP connections to your server. (The exact number of connections is determined at random.) After that, the tester will send a single `GET` request through each of the connections.

```
$ (sleep 3 && printf "GET / HTTP/1.1\r\n\r\n") | nc localhost 4221 &
$ (sleep 3 && printf "GET / HTTP/1.1\r\n\r\n") | nc localhost 4221 &
$ (sleep 3 && printf "GET / HTTP/1.1\r\n\r\n") | nc localhost 4221 &
```

Your server must respond to each request with the following response:

```
HTTP/1.1 200 OK\r\n\r\n
```

[View Code Examples](#)[View Screencasts](#)[Collapse ▾](#)

Tests failed.

[Show logs](#)

RECENT ATTEMPTS



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



adamc...



matthe...



gurugu...



maness...



tasnuval

1/11



Invite a friend



Private

X

leaderboards are

[Instructions](#)[Code Examples](#)[Screencasts](#)[Concepts](#)[Forum](#)

Stage #EJ5

Pending

You can use a server testing cli called [oha](#), which sends multiple concurrent requests by default. You can use it like `oha http://localhost:4221`.

[^ 76](#)[▼](#)[reply](#)**themilar** 7 months ago

nice looking tool, thanks for recommending, could come in handy beyond this challenge

[^ 1](#)[▼](#)**sherubthakur** 1 year ago

An easy way to test this locally would be to use `nc` to create a connection to your server and then use `curl` to hit it with a request. i.e. `nc localhost 4221` in one window followed by `curl localhost:4221` in a separate terminal window.

[^ 66](#)[▼](#)[reply](#)**ryan-gang** staff 1 year ago

For windows, which doesn't have `nc` built-in. `ncat` is a suitable substitute.

Docs : <https://nmap.org/ncat/>

Windows portable version : <https://nmap.org/dist/ncat-portable-5.59BETA1.zip>

After unzipping its just `ncat localhost 4221`

[^ 19](#)[▼](#)**daniel-j-anderson-dev** 11 months ago

Tests failed.

[Show logs](#)**RECENT ATTEMPTS**

11



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



adamc...



matthe...



gurugu...



maness...



tasnuval

1 / 11



Invite a friend



Private

X

leaderboards are

[Instructions](#)[Code Examples](#)[Screencasts](#)[Concepts](#)[Forum](#)

Stage #EJ5

Pending

check out [The Rust Book's final chapter](#) for a good and simple ThreadPool

^ 10 ▾ reply



GilTeixeira 1 year ago

If curl is hanging your request, you need to either close the connection or supply a content-length header field (set to zero). See RFC 7230, Section 3.3.3. (source: <https://stackoverflow.com/a/52869383>).

^ 9 ▾ reply



mevimo 9 months ago

For those seeing strange results when using a server testing tool like `oha` - make sure you implement keep-alive. It is the default in HTTP/1.1, and if not implemented, you'll probably close the connection while the server testing tool still means to use it for the next request. This won't present itself as a problem when testing with `curl`, because that's a one-connection-one-request scenario.

^ 8 ▾ reply



Remokc 6 months ago

For anyone having problems with undefined reference to 'pthread_create' :
<https://github.com/codecrafters-io/build-your-own-redis/issues/112#issuecomment-1872592856>

^ 7 ▾ reply

RECENT ATTEMPTS



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



adamc...



matthe...



gurugu...



maness...



tasnuval

1 / 11



Invite a friend



Private

x

leaderboards are

Tests failed. [Show logs](#)



< back next >



Upgrade



Return a file #AP6 Pending

Complete previous stages to gain access to this stage.

[Instructions](#)[Code Examples](#)[Screencasts](#)[Concepts](#)[Forum](#)

Unlock all stages with a CodeCrafters membership. No limits.

Plans start at \$30/month \$15/month (discounted price for Bangladesh) when billed annually.

Want CodeCrafters for free? You might be eligible to [claim the membership fees](#) back from your employer. Or you can [refer friends](#) to earn free weeks.

[View Membership Plans →](#)[No limits on content](#)[Turbo test runs](#)[Code examples](#)[Dark mode](#)[Anonymous mode](#)Tests failed. [Show logs](#)

RECENT ATTEMPTS 11



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



adamc...



matthe...



gurugu...



maness...

[Instructions](#)[Code Examples](#)[Screencasts](#)[Concepts](#)[Forum](#)

Stage #AP6

Pending

Your Task

Pending

[Share Feedback](#)

MEDIUM

In this stage, you'll implement the `/files/{filename}` endpoint, which returns a requested file to the client.

Tests

The tester will execute your program with a `--directory` flag. The `--directory` flag specifies the directory where the files are stored, as an absolute path.

```
$ ./your_program.sh --directory /tmp/
```

The tester will then send two `GET` requests to the `/files/{filename}` endpoint on your server.

First request

The first request will ask for a file that exists in the files directory:

```
$ echo -n 'Hello, World!' > /tmp/foo
$ curl -i http://localhost:4221/files/foo
```

RECENT ATTEMPTS

11



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



adamc...



matthe...



gurugu...



maness...



tasnuval

1/11



Invite a friend



Private

X

leaderboards are

Tests failed.

[Show logs](#)

[Instructions](#)[Code Examples](#)[Screencasts](#)[Concepts](#)[Forum](#)

Stage #AP6

Pending

```
$ curl -i http://localhost:4221/files/foo
```

Your server must respond with a `200` response that contains the following parts:

- `Content-Type` header set to `application/octet-stream`.
- `Content-Length` header set to the size of the file, in bytes.
- Response body set to the file contents.

```
HTTP/1.1 200 OK\r\nContent-Type: application/octet-stream\r\nContent-Le
```

Second request

The second request will ask for a file that doesn't exist in the files directory:

```
$ curl -i http://localhost:4221/files/non_existant_file
```

Your server must respond with a `404` response:

```
HTTP/1.1 404 Not Found\r\n\r\n
```

[View Code Examples](#)[View Screencasts](#)

Tests failed.

Show logs

RECENT ATTEMPTS



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



adamc...



matthe...



gurugu...



maness...



tasnuval

1/11



Invite a friend



Private

leaderboards are

[Instructions](#)[Code Examples](#)[Screencasts](#)[Concepts](#)[Forum](#)

Stage #AP6

Pending

- Content-Type header set to application/octet-stream .
- Content-Length header set to the size of the file, in bytes.
- Response body set to the file contents.

pe: application/octet-stream\r\nContent-Length: 14\r\n\r\nHello, World!



Second request

The second request will ask for a file that doesn't exist in the files directory:

```
$ curl -i http://localhost:4221/files/non_existant_file
```

Your server must respond with a 404 response:

```
HTTP/1.1 404 Not Found\r\n\r\n
```

[View Code Examples](#)[View Screencasts](#)[Collapse ↑](#)

Tests failed.

[Show logs](#)

RECENT ATTEMPTS



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



adamc...



jsec



gurugu...



maness...



tasnuval

1 / 11



Invite a friend



Private



leaderboards are

[Instructions](#)[Code Examples](#)[Screencasts](#)[Concepts](#)[Forum](#)

Stage #AP6

Pending

[Collapse ↑](#)

Hints

[Filter by Rust](#)[Write](#)[Preview](#)

Found an interesting resource? Share it with the community.

Markdown supported.

[Comment](#)**ryan-gang** staff 1 year ago

Just FYI the tests expect the Content-Length header.

[▲ 115](#)[▼](#)[Reply](#)

RECENT ATTEMPTS



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



adamc...



matthe...



gurugu...



maness...



tasnuval

1 / 11



Invite a friend



Private

leaderboards are here.

Tests failed.

[Show logs](#)



< back next >



Upgrade



Read request body #QV8

Pending

Complete previous stages to gain access to this stage.

[Instructions](#)[Code Examples](#)[Screencasts](#)[Concepts](#)[Forum](#)

Unlock all stages with a CodeCrafters membership. No limits.

Plans start at \$30/month \$15/month (discounted price for Bangladesh) when billed annually.

Want CodeCrafters for free? You might be eligible to [claim the membership fees](#) back from your employer. Or you can [refer friends](#) to earn free weeks.

[View Membership Plans →](#)[No limits on content](#)[Turbo test runs](#)[Code examples](#)[Dark mode](#)[Anonymous mode](#)

RECENT ATTEMPTS 11



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



adamc...



matthe...



gurugu...



maness...

Tests failed.

[Show logs](#)

[Instructions](#)[Code Examples](#)[Screencasts](#)[Concepts](#)[Forum](#)

Stage #QV8

Pending

Your Task

Pending

[Share Feedback](#)

MEDIUM

In this stage, you'll add support for the `POST` method of the `/files/{filename}` endpoint, which accepts text from the client and creates a new file with that text.

Request body

A request body is used to send data from the client to the server.

Here's an example of a `POST /files/{filename}` request:

```
// Request line
POST /files/number HTTP/1.1
\r\n

// Headers
Host: localhost:4221\r\n
User-Agent: curl/7.64.1\r\n
Accept: */*\r\n
Content-Type: application/octet-stream // Header that specifies the fo
Content-Length: 5\r\n                                // Header that specifies the si
\r\n

// Reauest Body
```

Tests failed.

[Show logs](#)

RECENT ATTEMPTS

11



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



adamc...



matthe...



gurugu...



maness...



tasnuval

1/11



Invite a friend



Private

X

leaderboards are

[Instructions](#)[Code Examples](#)[Screencasts](#)[Concepts](#)[Forum](#)

Stage #QV8

Pending

\r\n

// Request Body

12345

Tests

The tester will execute your program with a `--directory` flag. The `--directory` flag specifies the directory to create the file in, as an absolute path.

```
$ ./your_program.sh --directory /tmp/
```

The tester will then send a `POST` request to the `/files/{filename}` endpoint on your server, with the following parts:

- `Content-Type` header set to `application/octet-stream`.
- `Content-Length` header set to the size of the request body, in bytes.
- Request body set to some random text.

```
$ curl -v --data "12345" -H "Content-Type: application/octet-stream" ht
```

RECENT ATTEMPTS



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



adamc...



matthe...



gurugu...



maness...



tasnuval

1/11



Invite a friend



Private

X

leaderboards are

Tests failed.

Show logs

[Instructions](#)[Code Examples](#)[Screencasts](#)[Concepts](#)[Forum](#)

Stage #QV8

Pending

Request body

A request body is used to send data from the client to the server.

Here's an example of a `POST /files/{filename}` request:

```
stream // Header that specifies the format of the request body  
      // Header that specifies the size of the request body, in bytes
```

Tests

The tester will execute your program with a `--directory` flag. The `--directory` flag specifies the directory to create the file in, as an absolute

Tests failed. [Show logs](#)

RECENT ATTEMPTS



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



adamc...



matthe...



gurugu...



maness...



tasnuval

1/11



Invite a friend



Private

X

leaderboards are

[Instructions](#)[Code Examples](#)[Screencasts](#)[Concepts](#)[Forum](#)

Stage #QV8

Pending

Tests

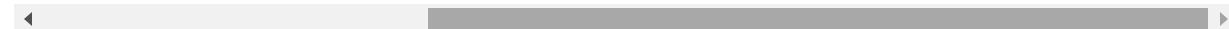
The tester will execute your program with a `--directory` flag. The `--directory` flag specifies the directory to create the file in, as an absolute path.

```
$ ./your_program.sh --directory /tmp/
```

The tester will then send a `POST` request to the `/files/{filename}` endpoint on your server, with the following parts:

- `Content-Type` header set to `application/octet-stream`.
- `Content-Length` header set to the size of the request body, in bytes.
- Request body set to some random text.

`Content-Type: application/octet-stream" http://localhost:4221/files/file_123`



Your server must return a `201` response:

```
HTTP/1.1 201 Created\r\n\r\n
```

Your server must also create a new file in the files directory, with the

Tests failed. [Show logs](#)

RECENT ATTEMPTS



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



adamc...



matthe...



gurugu...



maness...



tasnuval

1/11



Invite a friend



Private



leaderboards are

[Instructions](#)[Code Examples](#)[Screencasts](#)[Concepts](#)[Forum](#)

Stage #QV8

Pending

```
Content-Type: application/octet-stream" http://localhost:4221/files/file_123
```

Your server must return a `201` response:

```
HTTP/1.1 201 Created\r\n\r\n
```

Your server must also create a new file in the files directory, with the following requirements:

- The filename must equal the `filename` parameter in the endpoint.
- The file must contain the contents of the request body.

[View Code Examples](#)[View Screencasts](#)[Collapse ↑](#)

Hints

[Filter by Rust](#)

Tests failed. [Show logs](#)

RECENT ATTEMPTS

11



alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



adamc...



matthe...



gurugu...



maness...



tasnuval

1/11



Invite a friend



Private



leaderboards are

[Instructions](#)[Code Examples](#)[Screencasts](#)[Concepts](#)[Forum](#)

Stage #QV8

Pending

 Markdown supported.
[Comment](#)**Jorjon** 1 year ago

To test locally using cURL:

```
curl -vvv -d "hello world" localhost:4221/files/readme.txt
```

[^ 88](#)
[▼](#)
[Reply](#)
**yevelnad** 1 year ago

this "\0" will make your life a nightmare.

[^ 44](#)
[▼](#)
[Reply](#)
**friendlymatthew** 1 year ago

was pulling out my hair for the same thing haha.

note to myself: when converting buffer to a string, consider only the portion that was actually read as there may contain the null byte :)

[^ 23](#)
[▼](#)
**mwettste** 1 year ago

Haha same here. Lesson learned 😅

[^ 10](#)
[▼](#)
**that-ambuj** 1 year agoTests failed. [Show logs](#)RECENT ATTEMPTS 11

alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



adamc...



matthe...



gurugu...



maness...



tasnuval

1 / 11



Invite a friend



Private

X

leaderboards are

[Instructions](#)[Code Examples](#)[Screencasts](#)[Concepts](#)[Forum](#)

Stage #QV8

Pending

**friendlymatthew** 1 year ago

was pulling out my hair for the same thing haha.

note to myself: when converting buffer to a string, consider only the portion that was actually read as there may contain the null byte :)

[^ 23](#)[v](#)**mwettste** 1 year ago

Haha same here. Lesson learned 😅

[^ 10](#)[v](#)**that-ambuj** 1 year ago

For people using Rust, beware not to use `.read_to_string()` methods using something like a `BufReader`, this can potentially block your the thread and potentially the whole program(I don't who why but it just wouldn't work for me). instead use `.seek(&mut buf)` available on `TcpStream` because it seems to just work.

[^ 18](#)[v](#)[reply](#)**skunk-143** 1 year ago

Yes, because `read_to_string()` is like `read_to_end()` and it waits for the EOF, which means reset of TCP connection

[^ 25](#)[v](#)**RECENT ATTEMPTS**

alex-ev...



glenn...



adama...



vhfmag



lucasm...



DanDFav



adamc...



matthe...



gurugu...



maness...



tasnuval

1 / 11



Invite a friend

X



Private

leaderboards are
here.

Tests failed.

[Show logs](#)