Why HTTP

Communicating on the web

Instagram would be pretty terrible if you had to manually copy your photos to your friend's phone when you wanted to share them. Modern applications need to be able to communicate information between devices over the internet.

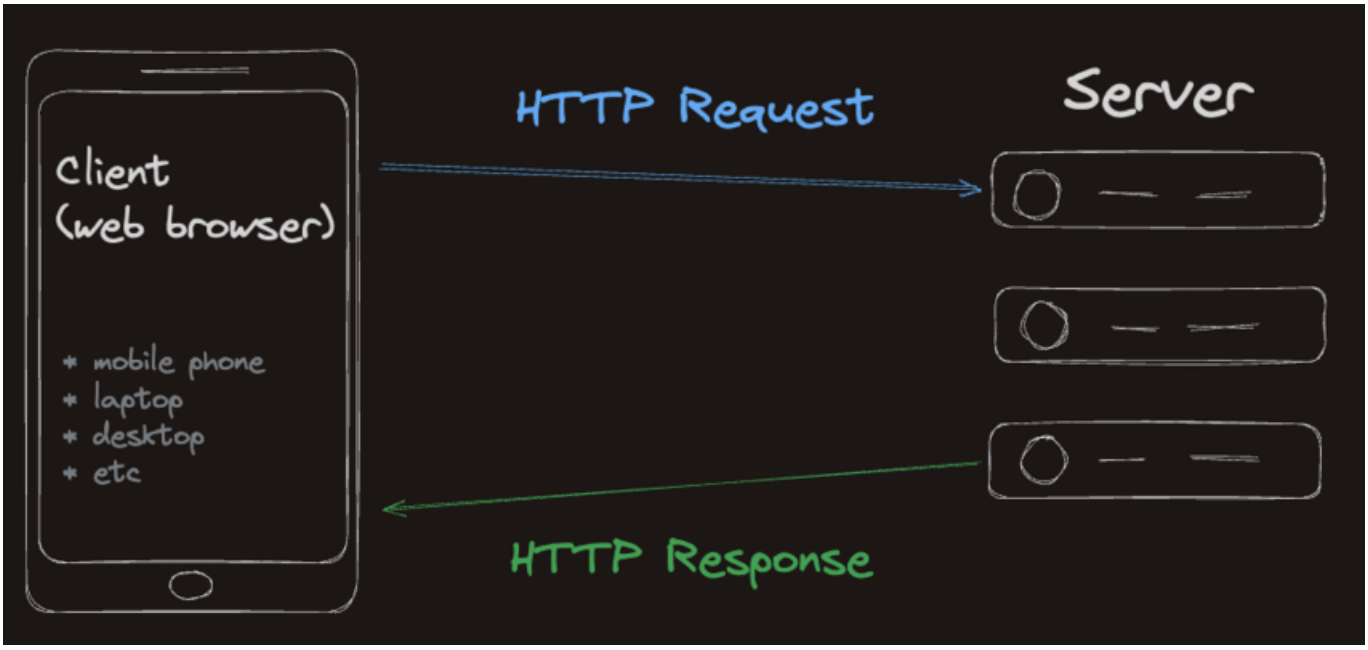
- Gmail doesn't just store your emails in variables on your computer, it stores them on computers in their data centers
- You don't lose your Slack messages if you drop your computer in a lake, those messages exist on Slack's servers How does web communication work?

When two computers communicate with each other, they need to use the same rules. An English speaker can't communicate verbally with a Japanese speaker, similarly, two computers need to speak the same language to communicate.

This **"language"** that computers use is called a **protocol**. The most **popular** protocol for web communication is **HTTP**, which stands for Hypertext Transfer Protocol.

HTTP Requests and Responses

At the heart of HTTP is a simple request-response system. The "requesting" computer, also known as the " the client", asks another computer for some information. That computer, "the server" sends back a response with the information that was requested.



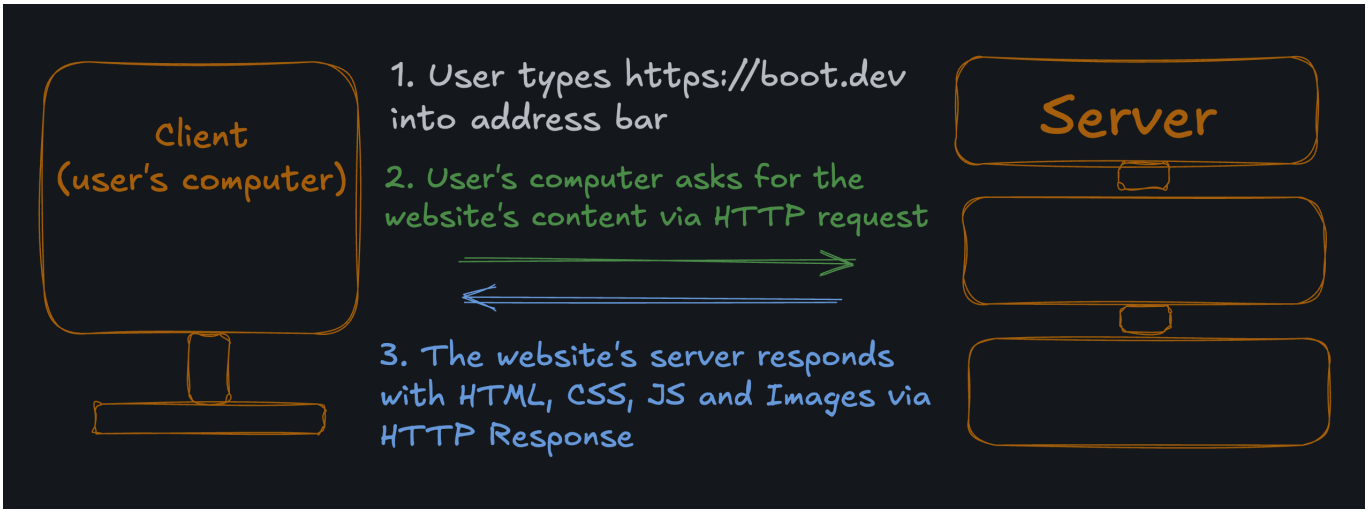
We'll talk about the specifics of how the "requests" and "responses" are formatted later. For now, just think of it as a simple question-and-answer system.

- Request: "What issues are on Jello?"
- Response: `["Fix bug", "Improve auth flow"]`

HTTP Powers websites

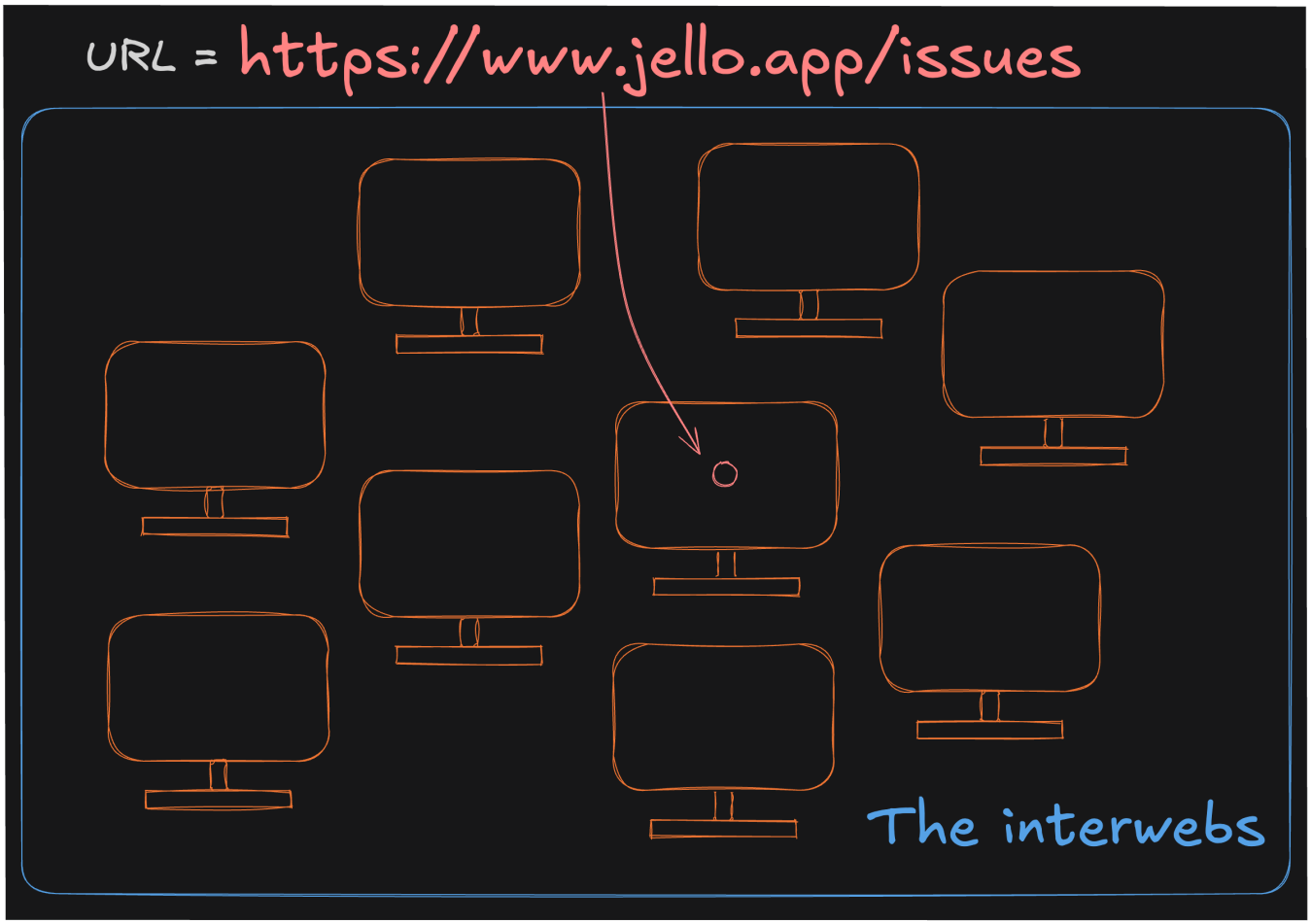
HTTP, or Hypertext Transfer Protocol, is a protocol designed to transfer information between computers.

There are other protocols for communicating over the internet, but HTTP is the most popular and is particularly great for websites and web applications. Each time you visit a website, your browser is making an HTTP request to that website's server. The server responds with all the text, images, and styling information that your browser needs to render its pretty website!



HTTP URLs

A URL, or Uniform Resource Locator, is the address of another computer, or "server" on the internet. Part of the URL specifies where to reach the server, and part of it tells the server what information we want.



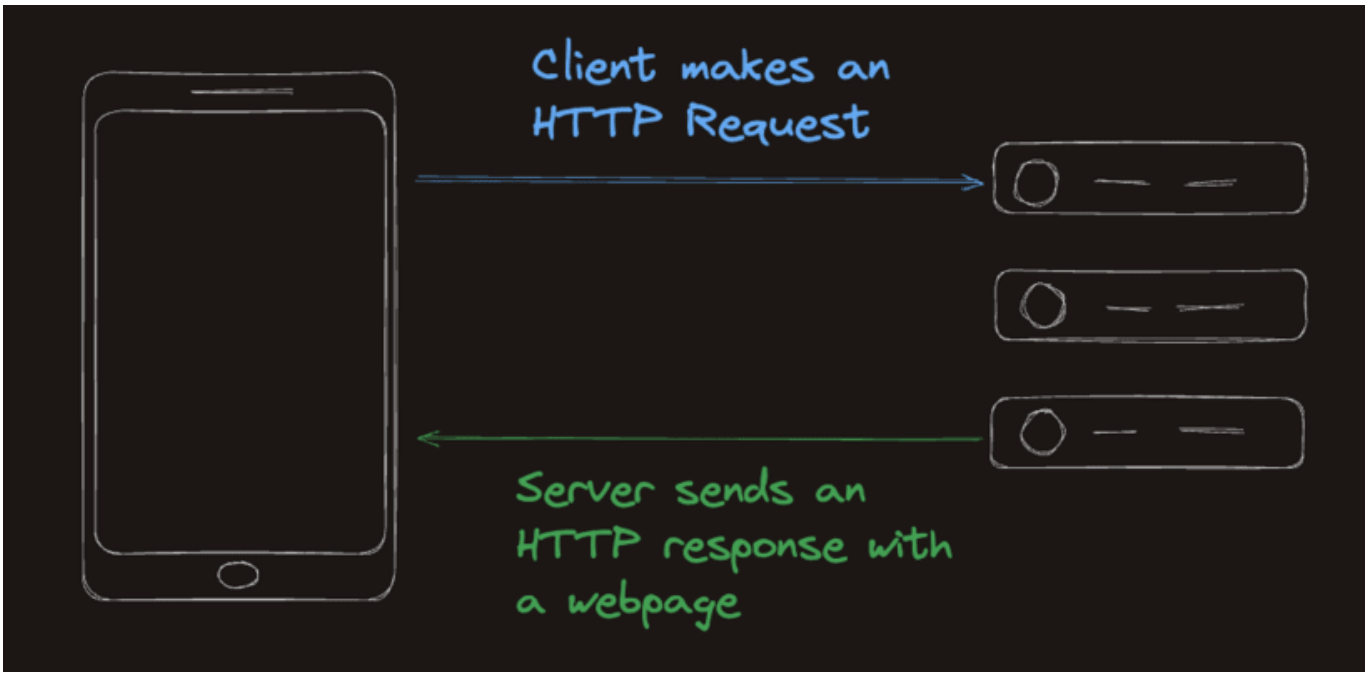
Put simply, a URL represents a piece of information on some computer somewhere. We can get access to it by making a request, and reading the response that the server replies with.

Using URLs in HTTP

The `http://` at the beginning of a website URL specifies that the `http` protocol will be used for communication.

Other communication protocols use URLs as well, (hence "Uniform Resource Locator"). That's why we need to be specific when we're making HTTP requests by prefixing the URL with ´http://`

Summary Requests and Responses




```
const issueURL = 'https://api.boot.dev/v1/courses_rest_api/learn-http/issues'

const issues = await getData(issueURL)

console.log(issues)

async function getData(url) {
  const response = await fetch(url, {
    method: 'GET',
    mode: 'cors',
    headers: {
      'X-API-Key': 'Testing',
      'Content-Type': 'application/json'
    }
  })
  return response.json()
}
```