Server

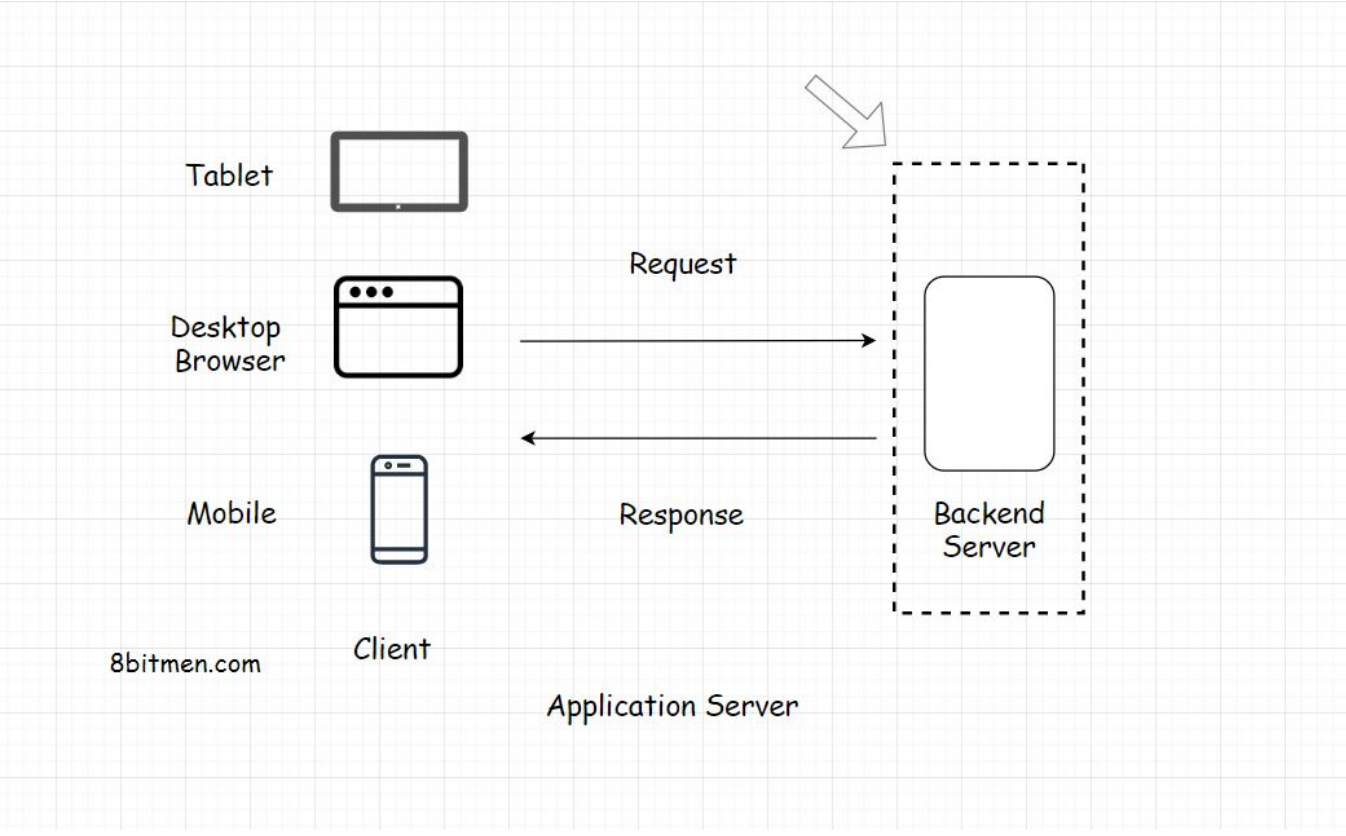
In this lesson, we will explore the Server component of the Client-Server Architecture.

WE'LL COVER THE FOLLOWING

What is A Web Server?

> The primary task of a web server is to receive the requests from the client & provide the response after executing the business logic based on the request parameters received from the client.

Every service, running online, needs a server to run. Servers running web applications are commonly known as the *application servers*.



Besides the *application servers*, there are other kinds of servers too with specific tasks assigned to them such as the:

- *Proxy server*
- *Mail server*
- *File server*
- *Virtual server*

The server configuration & the type can differ depending on the use case.

- For instance, if we run a backend application code written in *Java*, we would pick *Apache Tomcat* or *Jetty*.
- For simple use cases such as hosting websites, we would pick the *Apache HTTP Server*.

In this lesson, we will stick to the *application server*.

All the components of a web application need a server to run. Be it a database, a message queue, a cache or any other component. In modern application development, even the user interface is hosted separately on a dedicated server.

Server-Side Rendering

Often the developers use a server to render the user interface on the backend & then send the rendered data to the client. The technique is known as *server-side rendering*. I will discuss the pros & cons of *client-side* vs *server-side* rendering further down the course.

Now we have a fundamental understanding of both the client & the server. Let's delve into some of the concepts involved in the communication between them.