Introduction to the server side

Created by 이 윤 준 (yoonjoon.lee@gmail.com)

May, 2019

Look at server-side programming from a high level

- "What is it?"
- "How does it differ from client-side programming?"
- "Why it is so useful?"

What is server-side website programming?

Web browsers communicate with web servers using the HyperText Transfer Protocol (HTTP).

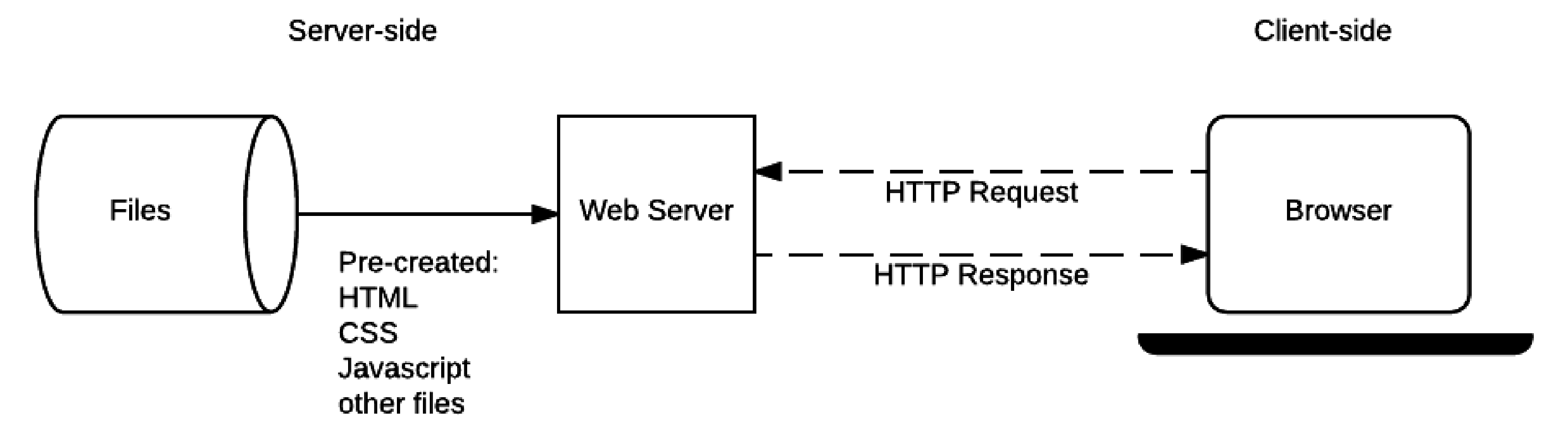
When you click a link on a web page, submit a form, or run a search, an HTTP request is sent from your browser to the target server.

The request includes

- a URL identifying the affected resource,
- a method that defines the required action
- may include additional information encoded in URL parameters as POST data, or in associated cookies.

Static sites

A basic web server architecture for *static site*



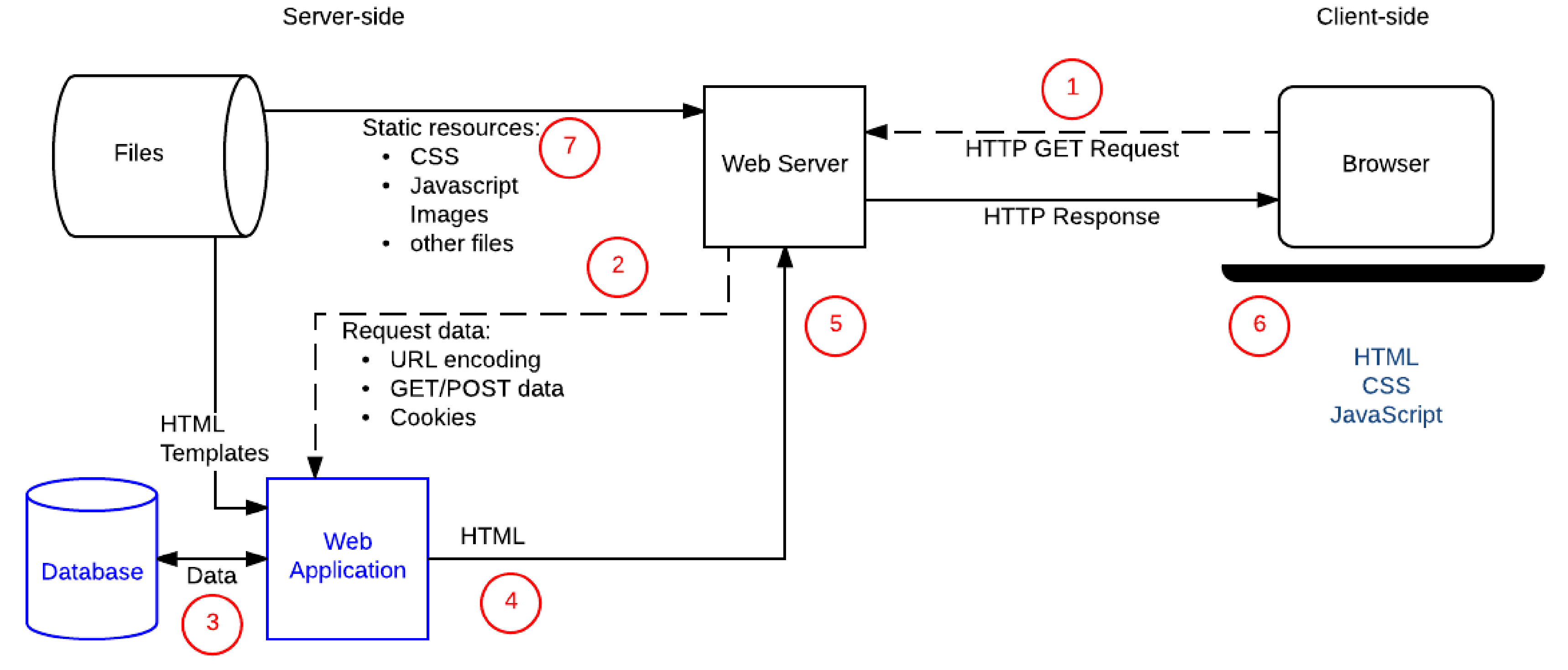
Dynamic sites

Some of the response content is generated dynamically only when needed.

Most of the code to support a dynamic website must run on the server.

Creating this code is known as "**server-side programming**".

A simple architecture for a *dynamic website*



Are server-side and client-side programming the same?

In server-side and client-side programming, the each code is significantly different:

- They have different purposes and concerns.
- They generally don't use the same programming languages.
- They run inside different operating system environments.

Code running in the browser is known as **client-side code** writen using HTML, CSS, and JavaScript.

This includes

- selecting and styling UI components,
- creating layouts,
- navigation,
- form validation, etc.

Server-side website programming mostly involves choosing *which* content is returned to the browser in response to requests.

The server-side code written in any number of programming languages — examples of popular server-side web languages include PHP, Python, Ruby, C#, and NodeJS(JavaScript) handles

- tasks like validating submitted data and requests,
- using databases to store and retrieve data
- sending the correct data to the client as required.

Developers typically write their code using **web frameworks**.

Web frameworks are collections of

- functions,
- objects,
- rules and
- other code constructs

designed to solve common problems, speed up development, and simplify the different types of tasks faced in a particular domain.

While both client and server-side code use frameworks, the domains are very different, and hence so are the frameworks.

Client-side web frameworks simplify layout and presentation tasks.

Server-side web frameworks provide a lot of "common" web server functionality that the developpers might otherwise have to implement themselves.

What can you do on the server-side?

Server-side programming allows us to efficiently deliver information tailored for individual users and thereby create a much better user experience.

Efficient storage and delivery of information

Server-side programming allows us to instead store the information in a database and dynamically construct and return HTML and other types of files (e.g. PDFs, images, etc.).

It is also possible to simply return data (JSON, XML, etc.) for rendering by appropriate client-side web frameworks.

The server is not limited to sending information from databases, and might alternatively return the result of software tools, or data from communications services.

Customised user experience

Servers can store and use information about clients to provide a convenient and tailored user experience.

A deeper analysis of user habits can be used to anticipate their interests and further customize responses and notifications.

Controlled access to content

Server-side programming allows sites to restrict access to authorized users and serve only the information that a user is permitted to see.

Store session/state information

Server-side programming allows developers to make use of sessions — basically, a mechanism that allows a server to store information on the current user of a site and send different responses based on that information.

Notifications and communication

Servers can send general or user-specific notifications through the website itself or via email, SMS, instant messaging, video conversations, or other communications services.

Data analysis

A website may collect a lot of data about users:

- what they search for,
- what they buy,
- what they recommend,
- how long they stay on each page.

Server-side programming can be used to refine responses based on analysis of this data.