Lecture #15: Server-Side Programming

CS106E, Young

In this lecture, we learn about programming on the server. We consider several different models of how a program can generate a webpage. We see that there are many different languages and frameworks that can be used on the server-side. The tools used on both client-side and server-side are sometimes referred to as the Development Stack. Finally we consider two different data formats XML and JSON which are used in web programming.

Front End and Back End Engineering

- We will often divide the components of a webserver into two distinct parts – front-end and back-end.
    - You may hear people talk about being a front-end developer or a back-end developer or talk about their web applications front-end or back-end.

- The Front End consists of the HTML, CSS, and any Client-Side Programs (i.e., JavaScript)
- The Back End consists of Server-Side Programs and the Database
    - Back-end engineers would also work on server-side configuration, load balancing, content delivery network (CDN) and server infrastructure issues.

Basic Models of Server-Side Programming

- There are two traditional methods for creating a webpage server-side.

HTML with Server-Side Code Added

- With this approach a webpage on the server-side looks almost like a tradition HTML file, except in a few places, we ask the server to insert new information based on code executed on the server.
- Here is a sample using the server-side language PHP.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8" />
<title>Time of Day</title>
</head>
<body>
<h1>Time of Day</h1>

<?php
```

```
$now = new DateTime();

echo $now->format("h:i:sA");

?>

</body>
</html>
```

- Notice how this looks almost exactly like HTML, except for the section enclosed within the <?php and ?>. This is our PHP code.
- When a user visits our webpage, the server will send the HTML in our file, but before sending it, it will execute the PHP code within the <?php … ?> section.
- This approach to server-side programming works well if your webpage is mostly static, but you have a few items you want to add to it using server-side programming.
- Using this method provides a gentle introduction to server-side programming, as you can use all the HTML skills you've learned and then just learn a bit of server-side programming to enhance your webpages.

Server-Side Code Generating HTML

- In this approach we generate a new HTML file from scratch.
- Here's an example using JavaScript for Node.js

```
var http = require('http');

var server = http.createServer(function(request, response) {
      response.writeHead(200, {"Content-Type": "text/html"});
      response.write("<h1>Simple HTML Example</h1>");
      response.end("<p>Hello World!</p>");
   });

server.listen(8080);
```

- You can see the code used to print or write out a new file in the middle of this code. See for example:

```
response.write("<h1>Simple HTML Example</h1>");
```

which writes an <h1> header in the newly created HTML file.

- This looks like a traditional program which generates a new file, in contrast to the last approach which looked like an HTML file with a few enhancements.
- This approach can be used to generate new HTML files, new Image files, or any other type of file.

- There is a third approach which is increasingly becoming popular.

    Same HTML using Client-Side JavaScript to Modify Webpage
        - In this approach we always serve up a traditional HTML file. However, that file includes client-side JavaScript code, which pulls data from the server and modifies the webpage on the fly.
        - We'll take a closer look at this next lecture when we talk about client-side programming.
        - This approach is particularly useful when trying to create a **Single Page Application (SPA)**
            - An SPA is a website consisting of only a single webpage, where that webpage is dynamically modified in response to user interactions.
            - Google Mail is an example of an SPA. We go to the gmail.com webpage, and while we interact with it and we see different things as we click on our mail lists or on the "Compose" button, we never actually leave that initial webpage.

Server-Side Programming Languages and Frameworks

- There are many, many different programming languages used on the server-side.
- While some of these, such as PHP, were designed from the start to work as web languages, others are general purpose languages, that can also be used on the web.
- A general purpose language used on the web will usually have some sort of library or framework allowing it to be used in conjunction with a web server.
    - For example, Python is a general purpose programming language. Django is a framework (a set of predefined functions, classes, and objects) which allows us to write server-side code with Python.
    - Some other common examples include: JavaScript using Node.js, Ruby using Ruby on Rails, Java using Spring.
    - Some languages have several different frameworks that can be used for web application programming.
        - So if I say I'm using Python as our server-side language, I might be using Python with Django, or I might be using Python with Flask.
        - I might be using Java with the Spring framework, with the JavaServer Faces framework, or with Struts.

Development Stack

You may hear developers talking about the Development Stack or Software Stack they are using. What they are referring to is the range of technologies which underlies their web application.

Some popular stacks are:

LAMP

Traditionally this has consisted of

- a server running the Linux operating system,
- an Apache web server program actually serving up the webpages,
- a MySQL database
- PHP as the language generating webpages.

Some people use the term with Python replacing PHP.

MEAN

This stack consists of
- MongoDB as our database,
- Express (a framework which makes life easier when using Node.js)
- Angular as a client-side framework (which uses TypeScript – an enhanced superset of JavaScript)
- Node.js as our web server

Application Programming Interfaces (API)

- In general an API is the set of classes and functions that an application provides allowing other programs to take use its services.
  - For example the Windows API is the list of functions that a program written for the Windows Operating System can take advantage of.
  - Microsoft Office applications also provide APIs, which can be used by programmers who wish to integrate their software with Microsoft Office.
- On the web, many web servers provide APIs allowing other web servers to take advantage of their services.
  - For example, Twitter has an API which allows a program to programmatically search Tweets or get Twitter Account information.
  - Google Docs has an API that allows us to programmatically create documents and read and modify documents.

Data Formats Used by Web Services

- XML (Extensible Markup Language) and JSON (JavaScript Object Notation) are widely used as Data Formats for Server-Side APIs

- XML is not really a language. Instead it's a specification to allow the creation of new languages.
  - XML based languages use tags, attributes, and values to store information, just like HTML.
  - A variety of XML-based languages have been specified. For example:
    - CML (Chemical Markup Language) has tags and attributes to represent atoms, molecules, and bonds.
    - SVG (Scalar Vector Graphics) is used to represent diagrams. It has tags for ovals and rectangles and attributes to represent colors and line widths.
  - Languages following the XML rules can be processed by special tools.

  - Because XML is relatively easy to understand and has good tool support, it is sometimes used as a data format.

  - Here is a sample of XML generated by OpenWeather a server with an API providing information on the weather.

```
<?xml version="1.0" encoding="UTF-8"?>
<current>
    <city id="0" name="Stanford">
        <coord lon="-122.16" lat="37.42"/>
        <country>US</country> …
    </city>
    <temperature value="65.41" min="57.2"
                 max="71.6" unit="fahrenheit"/>
```

```
        <humidity value="82" unit="%"/>
        <pressure value="1017" unit="hPa"/>
        …
    </current>
```

- JSON is a variant of the format used to represent objects in the JavaScript language.
    - JSON represents objects as a set of key-value pairs.
    - It's relatively easy to understand and can be easily processed by JavaScript on the client-side (as we'll see next lecture, all web browsers can process JavaScript), some servers also use JavaScript on the server-side.

    - Here the same information in our OpenWeather example, instead represented as JSON (OpenWeather will provide the data in either format):

```
{
    "coord":{"lon":-122.16,"lat":37.42},
    …
    "main":{"temp":65.41,"pressure":1017,
            "humidity":82,
            "temp_min":57.2,"temp_max":71.6},
    "visibility":11265,
    "wind":{"speed":6.93,"deg":270},
        …
    "name":"Stanford",
…
}
```