Algorithms and Programming IV
Web App Development 2: URI & URL (24-1)
Summer Term 2021 | 05.07.2021
Barry Linnert (Michael Tebbe)
Our Topic’s Today

• Recap Last Lecture

• Communication with the server
  - HTTP
  - REST Webservices
  - Sockets

• Webserver Examples
Recap: Web Application Architecture

Example taken from https://engineering.videoblocks.com/web-architecture-101-a3224e126947
Recap: Basic Architecture

User

Frontend
(Browser, Smartphone, Wearable...)

Internet

Web Server

Application Logic
PHP/Python/JavaScript

Database
MySQL/MongoDB/PostgreSQL

Web Application

Backend
# Recap: Layers ISO Model vs. TCP/IP Model

<table>
<thead>
<tr>
<th>ISO Model</th>
<th>TCP/IP Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>Application</td>
</tr>
<tr>
<td>Presentation</td>
<td></td>
</tr>
<tr>
<td>Session</td>
<td></td>
</tr>
<tr>
<td>Transport</td>
<td>Transport</td>
</tr>
<tr>
<td>Network</td>
<td>Internet</td>
</tr>
<tr>
<td>Data link</td>
<td>Network Access</td>
</tr>
<tr>
<td>Physical</td>
<td></td>
</tr>
</tbody>
</table>
# Recap: Layers ISO Model vs. TCP/IP Model

<table>
<thead>
<tr>
<th>ISO Model</th>
<th>TCP/IP Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>Application</td>
</tr>
<tr>
<td>Presentation</td>
<td></td>
</tr>
<tr>
<td>Session</td>
<td></td>
</tr>
<tr>
<td>Transport</td>
<td>Transport</td>
</tr>
<tr>
<td>Network</td>
<td>Internet</td>
</tr>
<tr>
<td>Data link</td>
<td>Network Access</td>
</tr>
<tr>
<td>Physical</td>
<td></td>
</tr>
</tbody>
</table>
Addition: A typical Ajax Request

1. User clicks, invoking event handler
2. Handler's JS code creates an XMLHttpRequest object
3. XMLHttpRequest object requests a document from a web server
4. Server retrieves appropriate data, sends it back
5. XMLHttpRequest fires event to say that the data has arrived (this is often called a callback; you can attach a handler to be notified when the data has arrived)
6. Your callback event handler processes the data and displays it
Fundamental Technologies of the Web

**HTML**: HyperText Markup Language. The publishing format for the Web, including the ability to format documents and link to other documents and resources.

**URI**: Uniform Resource Identifier. A kind of “address” that is unique to each resource on the Web.

**HTTP**: Hypertext Transfer Protocol. Allows for the retrieval of linked resources from across the Web.
Uniform Resource Identifier (URI)
Resource identification

A Uniform Resource Identifier (URI) provides a simple and extensible means for identifying a resource.

What is a resource?

- The term "resource" is used in a general sense for whatever might be identified by a URI. Familiar examples are websites, books, places, people, relations between these resources but also abstract concepts, such as the operators and operands of a mathematical equation.

The concept of an URI is already established in various domains such as the Web (URL), books (ISBN), digital object identifier (DOI).
URI, URL and URN

A URI can be further classified as a locator, a name, or both.

**Uniform Resource Locator (URL, RFC1738)**
- Refers to the subset of URIs that, in addition to identifying a resource, provide a means of locating the resource by describing its primary access mechanism (e.g., its network "location")
- Problem: can change over a lifetime of a web resource

**Uniform Resource Name (URN, RFC2141)**
- Refers to URIs, which are required to remain globally unique and persistent even when the resource ceases to exist or becomes unavailable, and to any other URI with the properties of a name
URL Schema


- protocol
- host
- port
- path
- anchor
- query
- authority
How does a Domain Name Server work?
DNS & HTTP

(Wilde, 2008)
How does your Browser communicate with a Server?

1. Web User requests for TCP connection to Web Server at port 80 at www.sjsu.edu

2. Web Server at www.sjsu.edu accepts the TCP connection at port 80 notifying web user.

3. Web User sends HTTP request message (containing URL) to TCP connection requesting the objects.

4. Web Server receives HTTP request message (containing URL) and responses with the requested objects.
How does your Browser communicate with a Server?

4. Web Server closes the TCP Connection.

5. Web User receives the requested HTML file with the objects and displays the file www.sjsu.edu

6. The above steps continue for the referenced file till the number of objects
Resources vs. Representations

URIs identify resources
- Abstractions which may not have physical representation
- Requesting a URI yields a resource representation
- Should be an appropriate and useful manifestation of the abstraction

Resources can have different representations
- In a well-designed environment, you should get what works best for you
- HTML for big screens vs. HTML for mobile devices
- An event calendar based on my location and preferences

(Wilde, 2008)