Development of Web Applications

Principles and Practice

Vincent Simonet, 2013-2014
Université Pierre et Marie Curie, Master Informatique, Spécialité STL
1 Architecture of Web Applications

Vincent Simonet, 2013-2014
Université Pierre et Marie Curie, Master Informatique, Spécialité STL
Objectives of the course

- Have an overall knowledge of the principles and technologies for the development of web applications.
- Practice by developing one complete web application.
Objectives of the course

The challenge: There is a multitude of technologies for developing web applications.

The solution: Explain principles, give an overview of the market, and focus on one example: Java Servlets.

Why Java Servlets?
- Widely used,
- Java,
- Basic mechanisms remain visible,
- Cloud hosting is possible.
Contents

1. Architecture of web applications
2. Communication
3. Server Technologies
4. Client Technologies
5. Web Development Frameworks
6. Practical Aspects
7. Project Presentations
Prerequisites

- Java programming,
- Basics in HTML and CSS,
- Basics in JavaScript.

If you're not familiar with these technologies, follow the tutorials referenced in the lecture notes.
Evaluation

- Continuous evaluation: **50%** (surprise tests!)
- Project: **50%**

Missing a test without acceptable justification = 0
References and Further Reading

Books:
- A few general books (see the list in the lecture notes),
- A multitude a technology-specific books!

The best documentation is probably on the web! (and free :) See in the lecture notes (especially Wikipedia).
Contact information

vincent.simonet@vtst.net

http://www.normalesup.org/
~simonet/teaching/upmc-master
Web applications?
Client/server: a software definition

Servers *(a.k.a. services or daemons)* execute by waiting for requests from *client* programs to arrive, and then processing those requests.

Client programs might be applications used by human beings, or they could be servers that need to make their own requests.
Client/server: a hardware definition

Client/server

Peer-to-peer
What is a web application?

It is a client/server application that uses a web browser as its client program, and performs an interactive service by connecting with servers over the Internet (or Intranet).

A web site simply delivers content from static files. A web application presents dynamically tailored content based on request parameters, tracked user behaviors, and security considerations.
Examples of web applications
Benefits

- Easy to deploy and upgrade,
- Cross-platform compatibility,
- Limited resources on client side,
- Interoperability.
Drawbacks

- Limitations on user interface compared to native Graphical User Interface,
- Compatibility issues with some web browser,
- Require a network connection,
- The user does not own the software.
A Brief History
Key dates

1993: Mosaic browser, CGI
1995: PHP 1.0
1996: JavaScript 1.0
1999: Web Application, Java Servlet (server)
2005: AJAX
2008: HTML5 first public working draft
2014?: HTML5 specification
(User) client vs (remote) server

- **70s**: Light user terminals, everything is done by the server.
- **80s/90s**: Personal computers. Everything happens on client side.
- **90s/2000s**: Light client (web browser), all logic in server.
- **2000s**: Logic is back in the client ("Web 2.0").
- **2010s**: Mobile applications.
Overall Architecture
Typical architecture of a web application

- **Persistent data store**
- **Web Server (front-end)**
- **Web Browser**

Communication: HTTP(S)
Web browser

- Mainly user interface,
- Short term state (in general),
- May implement some logic, especially for fast response time (but untrusted),
- Communicate with the web server using HTTP(S),
- Executing HTML, CSS and JavaScript code.
Web server (front-end)

- Answers to HTTP(S) requests from the web clients,
- Stateless,
- Reads and writes data in a persistent data store,
- Performs most of the business logic,
- Consists in a general of a server/container (Apache, Tomcat) and a framework (PHP, Java Servlets, etc.) running business logic.
Data store

- The state of the web application,
- Historically a (My)SQL database, some more recent evolutions,
- The synchronisation point.
Back-end

- All what needs to be done in the server, but which is not triggered by a client request.
Typical architecture of a web application

Persistent data store

Web Server (front-end)

Web Browser

Back-end