Web application development
(part 1)

Netzprogrammierung
(Algorithmen und Programmierung V)
Our topics today

Static webpages using Hyper Text Markup Language (HTML) and Cascading Stylesheets (CSS)

- mostly for self-study
- Focus on HTML5

Server Sided Approaches
- CGI
- Servlets
- PHP and JSPs

Client Sided Approaches
- Applets
- Java Script
- Ajax
Web application development (part 1)

Hyper Text Markup Language (HTML) and Cascading Stylesheets (CSS)
HTML 5 – What is new?

HTML5: „A vocabulary and associated APIs for HTML and XHTML“

One vocabulary, two serializations

- HTML Syntax
- XML Syntax

DOM focused
- parsing rules for a DOM (incl. Execution handling)

Definition of DOM-APIs
HTML 5 Serializations and DOM

HTML syntax

```html
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Example doc</title>
  </head>
  <body>
    <p>Example paragraph</p>
  </body>
</html>
```

XML syntax

```xml
<?xml version="1.0" encoding="UTF-8"?>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Example document</title>
  </head>
  <body>
    <p>Example paragraph</p>
  </body>
</html>
```

One „Document Object Model“
HTML 5 Syntax Variants and Exception Handling (examples)

Empty elements `<img .../>` `<img ...>`
Minimal attributes `<input>`
Upper and lower-case `<html>` `<HTML>`
Attributes without quotations `<img scr=... >`
Element Nesting `<b>a<i>b</i>c</b>`
HTML 5 New Features (examples)

Audio
- `<audio src="Mc202example.ogg" controls autoplay></audio>`

Video
- `<video src="Mirko_The_Artist-med.ogg", controls> </video>`

Graphics
- `<svg xmlns:svg="http://www.w3.org/2000/svg" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 200 100" width="200px" height="100px">
  <circle cx="50" cy="50" r="30", style="stroke:none; fill:#ff0000;"/>
</svg>`
HTML 5 New Elements (examples)

section
article
aside
hgroup
header
footer
nav
dialog
...

HTML 5 – New Attributes

Examples
- charset within <meta> element
- pattern within <input> element

ARIA (Accessible Rich Internet Applications):
- ARIA role
- ARIA aria-* Attribute
HTML 5 – New APIs

2D drawing API for `<canvas>`
APIs for `<video>` and `<audio>`
Drag & drop API and draggable attribute
API for access on browser history
Geolocation API (not defined within HTML5)
...

Summary: Differences HTML 5 and HTML 4

HTML 5

- DOM as basis
- User agents vs. authors
- Syntax+Parsing
- MathML+SVG support + APIs
- New Elem. & Attr.

HTML 4

- Only Markup vocabulary
- No separation
- Only Syntax

see http://www.w3.org/TR/html5-diff/
Dynamic Web Applications

... more than then returning a static web content (HTML, GIF, AVI, …)

Two approaches for dynamic WBIS:
- **Client sided** integration = Code at Client and DB access by Browser with means from the WWW, e.g. ActiveX, Applets
- **Server sided** integration = dynamic HTML generation

Combination possible and useful;
Focus in this lecture: Server sided approaches
Server Sided Approaches - Examples

(more than the generation of static HTML web pages)

CGI - Common Gateway Interface

Web Server APIs (e.g. Servlets)

Server sided Scripts (e.g. JSPs)

…
Web application development (part 1)

**Common Gateway Interface (CGI)**
CGI basics

The Common Gateway Interface (CGI) is a standard (protocol) for interfacing external applications with an HTTP server. A CGI program is executed in real-time, so that it can output dynamic information. ([http://www.ietf.org/rfc/rfc3875](http://www.ietf.org/rfc/rfc3875))

CGI defines the way by which web server software communicates with "backend programs" running on the server host. CGI is completely independent of programming language, operating system and web server.

CGI can be embedded into pages or produce complete pages
- embedded CGI results are expected to produce HTML fragments
- complete CGI results are expected to produce complete HTML content
- Check out [http://radblast-aa.wunderground.com/cgi-bin/radar/WUNIDS_map?station=MUX&brand=wui&num=6&delay=15&type=N0R&frame=0&scale=0.498&noclutter=0&t=1165711666&lat=37.86912155&lon=-122.26962280&label=Berkeley,+CA&showstorms=0&map.x=400&map.y=240&centr...](http://radblast-aa.wunderground.com/cgi-bin/radar/WUNIDS_map?station=MUX&brand=wui&num=6&delay=15&type=N0R&frame=0&scale=0.498&noclutter=0&t=1165711666&lat=37.86912155&lon=-122.26962280&label=Berkeley,+CA&showstorms=0&map.x=400&map.y=240&centr...
Application areas

The Web server can call up a program, while passing user-specific data to the program.

The program then processes that data and the server passes the program's response back to the Web browser.

Forms, e.g. shopping, booking

Gateways, e.g. search engine, database

Virtual documents, e.g. guestbook, chat, bulletin board, dictionary
Programming in CGI

Now that I know how CGI works, what programming language can I use?

You can use whatever language you want, but it should ideally include
• Ease of text manipulation
• Ability to interface with other software libraries and utilities
• Ability to access environment variables

Perl (UNIX, Windows, Macintosh) is the most widely used for CGI programming!
• Highly portable and readily available
• Powerful string manipulation operators
• Very simple and concise constructs
• Calling shell commands, and useful equivalents of certain UNIX system functions
• Extensions on top of Perl for specialized functions
Common Gateway Interface (CGI)

1. Server detects: request applies to program/script (no document)
2. Server starts the program/script with the arguments (e.g. from HTML form)
3. Server returns output of the program/script (HTTP + HTML) via Web Server to the client
4. In case of a failure the server returns a failure message to the client

Client (WWW-Browser) → get /cgi-bin/uptime.pl HTTP/1.0

WWW Server

Content-type: text/plain
9:36 am up 54 days,
HTML Forms

```html
<FORM method=GET action=http://www.google.com/custom>
  <INPUT TYPE=text name=q size=31 maxlength=255 value="">
  <INPUT type=submit name=sa VALUE="Search">
</form>
```
Environment Variables I

Server provides the program with environment variables!
Contains variables transferred by GET and system variables

- SERVER_NAME: Hostname/DNS/IP
- SERVER_PROTOCOL: Name + Revision
- REQUEST_METHOD: POST, GET
- PATH_INFO: Path of the program
- SCRIPT_NAME: Path + Name
- QUERY_STRING: data transferred by GET
- REMOTE_ADDR: IP of the requesting host
Environment Variables II

CONTENT_TYPE  provided data
CONTENT_LENGTH length of the data
HTTP_ACCEPT  MIME types of the browser
HTTP_USER_AGENT user name of the browser
HTTP_COOKIE  cookie data, if sent
HTTP_REFERER  address, from which the Browser came

Access to environment variable from programs:
  - C or C++:
    ```
    #include <stdlib.h>
    char *query = getenv("QUERY_STRING");
    ```
  - Perl:
    ```
    $query = $ENV{"QUERY_STRING"};
    ```
  - C shell:
    ```
    set QUERY = $QUERY_STRING
    ```
Hand-over via Standard Input

Server checks, if information is attached to header and puts
information on standard input of the CGI program
  - Server sets `CONTENT_LENGTH` and `CONTENT_TYPE`

Example: `x=4&y=2` transferred with POST

`x=4&y=2` on standard input of the program
  - `CONTENT_LENGTH=7`
  - `CONTENT_TYPE= application/x-www-form-urlencoded`

Example: Perl statement to read the data
  - `read(STDIN, my $Daten, $ENV{CONTENT_LENGTH});`
CGI Database Request

Confirm with Submit-Button
<br>&lt;FORM METHOD="POST"
ACTION="http://www.xyz.de/
cgi-bin/my-form">

Call script
with parameters
initialized for each
request

Start connection with
DB for each request

Script queries DB

Script generates
HTML

Script sends
HTML and terminates

Initiate connection

SQL request

Result

HTTP Server replies

Request

Response

CGI-Script

Barry Linnert, linnert@inf.fu-berlin.de, Netzprogrammierung WS 2015/16
Problem with State in HTTP, CGI

- Server cannot recognize the client due to the repeated new start of the server program
- Typical example: Online-Shop: Virtual shopping cart, customer ID

Identification
(Insert username and password)

Identified User:
Reference must be transferred between the Interactions, e.g. as an ID.

Anonymous user (general public information)
State Management

- URL-Encoding: predefined ID as argument of a form:
  `<FORM ACTION="http://csw/cgi-bin/states?ID=451" METHOD=POST>`

- Alternative: ID as hidden field:
  `<INPUT TYPE="HIDDEN" NAME="ID" VALUE="451">`

- Most browser support cookies, stored in the browser & resend to server with next request:
  Content-type: text/html
  Set-Cookie: ID=451
State management

Essential for supporting stateless interactions

Cookies are a frequently used mechanism for managing state
• In many cases used for maintaining session state (login/logout)
• More convenient than having to embed the state in every representation

Cookies have two interesting client-side side-effects
• They are stored persistently independent from any representation
• They are “shared state” within the context of one browser
Summary: Assessment of CGI

Advantages
- Arbitrary programs can be integrated (language independent)
- Security by own process (in user space)
- Implemented by all standard Web servers

Disadvantages
- One process per request
- For each DB request new connection and disconnection (!)
- No state management by HTTP Server
- No separation of presentation and application logic
- No load balancing on servers; except the application runs on separated server (invocation problem!)
Modern Web-Server APIs

Developed to overcome disadvantages of CGI scripts

Loaded in sever context
- Often need to be loaded only once
- Are executed in threads instead of processes
- Are able to store states

Prominent approaches
- Java Servlets (Java Servlet API, Sun)
- Apache API (MODPERL, MODPHP, …)
- NSAPI (Netscape)
- ISAPI
- …
Web application development (part 1)

Java Servlets
API Based Approaches

Confirm with Submit-Button
<FORM METHOD="POST" ACTION="http://www.xyz.de/cgi-bin/my-form">

Call the program, Handover parameter

Start connection to DB

Start connection to DB

Arbitrary many SQL queries

Rest State

Program

Initiate connection

SQL query

result

HTML generation

HTML response

Request

Response
Server sided Java => Servlets

Simple Web Server API for Java Applications

Platform independent and simple portability

Servlet keeps loaded, after request has been processes; CGI program needs to be reloaded every time

Servlet specification by Sun
Implementation by different Servlet engines
- Tomcat from Apache
- JRun from Macromedia (Allaire)
- IBM WebSphere
- Oracle Application Server
- …
Multithreading

Main Process

JVM

Servlet1

Servlet2

Request for Servlet1

Request for Servlet2

Request for Servlet1

thread

thread

thread
Implementation: Class HttpServlet
(Package: javax.servlet.http)

Provides basic functionalities to react to HTTP requests
Therefore the following methods can be implemented:
- **doGet**, to accept HTTP GET requests.
- **doPost**, to accept HTTP POST request
- **doPut**, for HTTP PUT
- **doDelete**, for HTTP DELETE requests
- **init** and **destroy**, for resources, which should be used during the life cycle of the servlet
Overview

GET request

response

POST request

response

Web Server

HttpGet Subclass

service()

doGet()

doPost()
Java Servlet - Methods

void init()

Initializes the Servlet (e.g. sets parameters, instantiates classes or opens DB connections)

void doGet(HttpServletRequest, HttpServletResponse)

called by requests of type GET, contains functionality of the service

void doPost(HttpServletRequest, HttpServletResponse)

called by requests of type POST, contains functionality of service

void destroy()

removes Servlet instance from memory (e.g. closes DB connection)
Example 1/3

```html
<html>
<body>
<form action="RequestParamExample" method=POST>
Vorname:
<input type=text size=20 name=firstname>
<br>
Nachname:
<input type=text size=20 name=lastname>
<br>
<input type=submit>
</form>
</body>
</html>
```
Example 2/3

```java
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.IOException;
import java.io.PrintWriter;

public class CSWServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // Auslesen der Request-Parameter
        String firstName = request.getParameter("firstname");
        String lastName = request.getParameter("lastname");
    }
}
```
Example 3/3

// Set ContentType and request output stream
response.setContentType("text/html");
PrintWriter out = response.getWriter();

// Output
out.write("<html>
<head><title>Servlet Example -
Result</title></head>");
out.write("<body>
<font face="Arial">
Vorname: " + firstName + "<br>
Nachname: " + lastName + "<br>
" + out.write("</font></body></html>"};

} // doPost
} // CSWServlet
State Management: Session Interfaces

java.servlet.http.Cookie
  - Set explicit Cookies

java.servlet.http.HttpSession
  - Simpler and more powerful
  - Cookies/URL rewriting transparent for developers
  - Objects are stored „in Sessions“
RequestDispatcher

1. **Input.html** (post) → **Servlet for Checking the Input** → **Database-Servlet** (forward)

2. **Input.html** (post) → **Component Servlet**
   - If (x < 0) → **Content1.html** (include)
   - If (x > 0) → **Content2.html** (include)

3. **Header.html** (include)

Barry Linnert, linnert@inf.fu-berlin.de, Netzprogrammierung WS 2015/16
Servlet Filter

Filter can change Requests or Responses before they reach a resource, or after they have been processed by the resource

- Sequence of filters can be define in the Deployment Descriptor

Application

- Check HTTP Request
- Authentication and authorization
- Auditing and logging
- …
Life Cycle of Servlets
Servlet Engines

Servlet Engines are basically a JVM as runtime environment for Servlets with specific methods for the life cycle management

Web-Server with integrated Servlet engine:
- Example: Java WebServer from Sun
- WebServer written in Java, Class files of the Servlet engine added to the WebServer files.

Separate Servlet engine
- Requires a separated runtime environment for the Servlet engine and a Web Server Plug-In for the integration of the Java environment
- The Plug-In among other is responsible for the communication between the Web Server and the Servlet engine.
- Tomcat
  - full Web-Server, often used as module of the Apache Web Server
  - Jakarta project, developed under Apache software licence
  - http://jakarta.apache.org
Assessment of Servlets

Advantages
- Overcomes most of the problems of CGI

Disadvantages
- No separation of presentation and application logic
Web application development (part 1)

Summary
What have we discussed today?

• We had a quick look of how to create static web pages with Hyper Text Markup Language (HTML) and Cascading Stylesheets (CSS).
• We looked at the new features of HTML 5
• We looked at CGI and Servlets for programming server-sided dynamic web applications
Questions

What is new in the W3C HTML5 recommendation?
Explain how CGI works and name the advantages and disadvantages?
Explain how Servlets work and how they overcome the disadvantages of CGI?
What is meant by server-sided and client-sided web applications?
References

Servlet and JSP Online Tutorial:
- http://www.apl.jhu.edu/~hall/java/Servlet-Tutorial/
- http://www.galileocomputing.de/openbook/javainsel5/, Chapter 17

Servlet and JSP book:

Overview article: