Servlet Technology

Web Application Development

Zsolt Tóth

University of Miskolc

2017

1 Back-end
- Programming Languages
- Application Servers

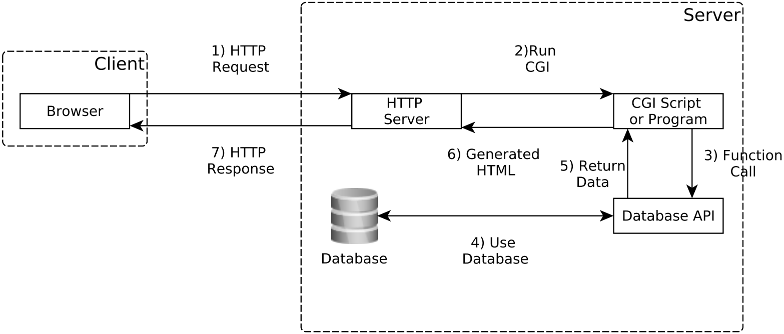
2 Web Applications
- Structure
- Deployment Descriptor

3 J2EE
- HttpServlet
- Filter

Common Gateway Interface

- CGI Program / Script
- PHP, C
- Generate HTML

- Started by the Web Server
- Create process for each HTTP Request
- Costly



Script Languages

PHP

- Most popular
  - Local companies use it.
- Frameworks
  - Yii
- Content Management Systems
  - Joomla
  - WordPress
- Easy–to–start
- Supported by web server hosting companies
- Wikipedia

Python

- High–level, general purpose language
- Popular but not in Miskolc
- Easy–to–Learn
- Google, Youtube

Ruby

- Ruby on Rails Framework
- Not popular in Miskolc
- Twitter

Object Oriented Languages

- Web Applications
  - Complex,
  - Reliable,
  - Stable
- Expensive Infrastructure
- Components can be separated easily
- Good, well-tested code (hopefully ☺)

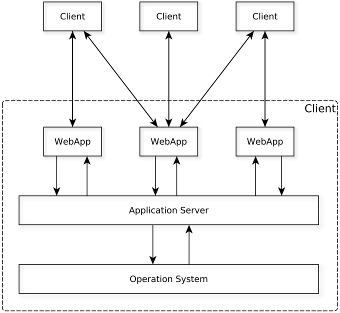
Development

- Slow
- Standardized
  - Fixed roles
  - Methodologies
- Difficult
  - If you do it well ☺
  - Lots of technology
- Both Java and C# are used by local companies

Overview

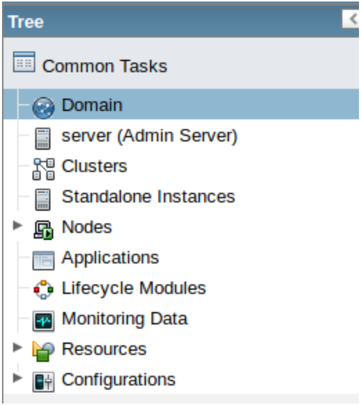
- Application
- Middleware
- Software Framework
- Environment for Dynamic Web Applications
- Contains
  - Servlet Container
  - JSP
  - Enterprise JavaBeans
- Servers
  - GlassFish, WildFly, Geronimo

  $ WebSphere, WebLogic, JBoss



Common Tasks

- Admin Console
- Deployment
- Scaling
  - Manage Nodes
  - Clustering
- Monitoring
  - Server
  - Applications
  - Resources
- Resource Management
  - JDBC
  - JNDI
  - JMS



Installation

- Java Application
- Platforms
  - Unix
  - Linux
  - Windows
- No `install.exe`
- Extract and run
- Script files
  - `bin/startup.sh`
  - `bin/shutdown.sh`

Hints for Developers

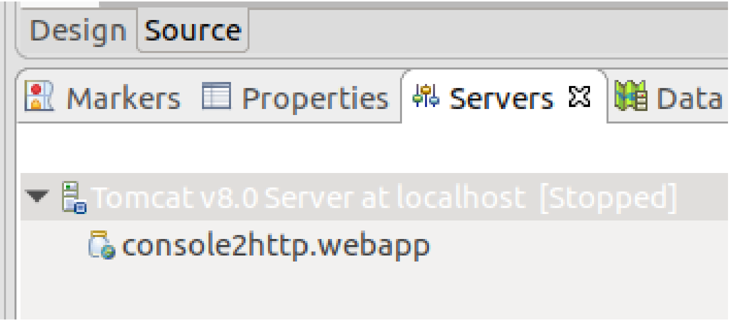
- Your are not a System Administrator.
- You should not operate an Application Server.
- Logging is essential.
- Log files grow quickly.
- Never run as `root`!

Usage as a Developer

- Local Deployment
- Manual Testing
- Demonstration
- ≈ Test Server

IDE Integration

- Eclipse
- IntelliJ IDEA
- NetBeans



Tomcat v8.0 Server at localhost [Stopped]
console2http.webapp

Tomcat

- Apache Software Foundation
- Lightweight
- Servlet Container
- Versions
  - 8.0
    - Servlet 3.1
    - JSP 2.3
  - 8.5
    - HTTP/2
    - OpenSSL
- Popular among developers!

Tomcat

Directories

- `bin`
  - Script files
  - `startup.sh`
  - `shutdown.sh`
- `conf`
  - XML
  - `server.xml`
  - `tomcat-users.xml`
- `webapps`
  -

```
.
+-- bin
+-- conf
+-- lib
+-- LICENSE
+-- logs
+-- NOTICE
+-- RELEASE-NOTES
+-- RUNNING.txt
+-- temp
+-- webapps
\-- work
```

Tomcat for Developers

- IDE
  - IntelliJ IDEA
  - Eclipse
- Multiple Servers
- Local Configuration
  - Workspace
- Start/Stop Server
- Deployment
- Refresh / Restart

Tasks

- Run Locally
- Test Manually

See also

- Database Management
- Software Testing
- Operation Systems

1. Back-end
   - Programming Languages
   - Application Servers

2. Web Applications
   - Structure
   - Deployment Descriptor

3. J2EE
   - HttpServlet
   - Filter

Structure

Web App

- Expected Directory Structure
- Standard Locations
- Deployed from Web ARchive

Root Directory

- Static Content
- Direct Access

```
+-- images
+-- index.jsp
+-- META-INF
|   \-- context.xml
+-- WEB-INF
|   +-- jsp
|   |   +-- 401.jsp
|   |   +-- 403.jsp
|   |   +-- 404.jsp
|   |   \-- sessions.jsp
|   \-- web.xml
```

Web ARchive

- ZIP File
- `*.war` Extension
- Contains Web Application
- Creation
  - `jar cvf ...`
  - Maven Packaging
- Can be Deployed

WEB-INF

Cannot be Accessed Directly!

`classes`

- `*.class` files
- Compiled Servlet Classes
- Package Structure

`lib`

- `*.jar` Files
- 3rd Party Dependencies

Deployment Descriptor

`WEB-INF/web.xml`

- Details of Web Application
- Name
- Description
- Servlet Definition
- URL Mapping
- Listeners
- Filters
- ...

```
<web-app xmlns="...">
<servlet>
    <servlet-name>
    greetings
    </servlet-name>
    <servlet-class>
    org.dummy.Greetings
    </servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>
    greetings
    </servlet-name>
    <url-pattern>/</url-patt
</servlet-mapping>
</web-app>
```

Servlet Mapping

- `servlet` tag
- Identified by `name`
- `servlet-class`
- Instance
- `init-param`
  - Initialization
- `load`

- `servlet-mapping`
- Key Value pair
- Extends application's URL
- * wild card

```
http://host:port/WebAppName/path
```

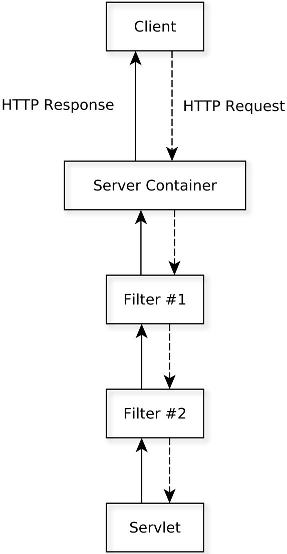
Filters

`filter`

- `filter-name` Identifier
- `filter-class`
- Similar to Servlet
- Preprocessing
- Forwarding / Rejection

`filter-mapping`

- `filter-name`
- Selection
  - `url-pattern`
  - `servlet-name`



Context-Param

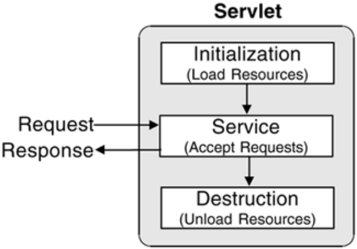
- Context Initialization
- Key-Value Paris
- Available in Java
- Predefined Keys
  - contextConfigLocation

```
<context-param>
 <param-name>
  contextConfigLocation
 </param-name>
 <param-value>
  /dispatcher-servlet.xml
  /applicationContext.xml
 </param-value>
</context-param>
```

1. Back-end
   - Programming Languages
   - Application Servers
2. Web Applications
   - Structure
   - Deployment Descriptor
3. J2EE
   - HttpServlet
   - Filter

Servlet

- `javax.servlet`
- Interface
- Generic Methods
  - `init`
  - `service`
  - `destroy`



HttpServlet

- `javax.servlet.http`
- Implements Servlet
- Run in Servlet Container
- Methods
  - `doGet`
  - `doPost`
  - `doPut`
  - `doDelete`

```
doGet(
 HttpServletRequest req,
 HttpServletResponse resp){
  //do something
 }
```

HttpServletRequest

- Request Information
- `getCookies`
- `getSession`
- `getHeaders`
- `getParameter`
- `getQueryString`

HttpServletResponse

- Servlet Response
- HTTP Header
- Cookies
- `add*`
- `sendError`
- `getWriter`
- `getOutputStream`

Filter

- Preprocessing
- Methods
    - `init`
    - `doFilter`
    - `destroy`

Types

- Authentication Filters
- Logging and Auditing Filters
- Image conversion Filters
- Data compression Filters
- Encryption Filters
- Tokenizing Filters
- Filters that trigger resource access events
- XSL/T filters
- Mime-type chain Filter