

HTTP & Application Servers

A word on prerequisites

As per the course website, the recommended prerequisites

- Good programming skills in Java
- Knowledge in object oriented design and UML
- Basic knowledge of data structures and algorithms, communication, Internet protocols and database design
- Courses **DAT050, DAT055, LET375, LEU061, TDA357** or equivalent

With the exception of Java (which you really need to know well to be able to take this course) we try to structure the course in a way that keeps the need for the other prerequisites to a minimum.

HTTP, HyperText Transfer Protocol



The problem that HTTP solves

- Client and server can be any type of machine
- Software can be written in any language
- Designed for TCP/IP, but can be used over any other transport layer
- HTTP stateless protocol, meaning the server does not have to retain information or status about each client for the duration of multiple requests. However, some web applications implement states or server side sessions using for instance HTTP cookies or hidden variables within web forms.

The model behind HTTP

What is the web?

- There are **resources** (files) at various locations on the Internet
- A location is given by an URL such as;
<https://tools.ietf.org/html/rfc7230>
- The client (browser) sends a **request** to the server
- The server sends back a **response** which indicates the the status of the resource and (if successful) contains the body of the resource

HTTP message layout

Method

Target

```
GET /hello.htm HTTP/1.1
User-Agent: Mozilla/4.0
(compatible; MSIE5.01; Windows
NT)
Host: www.tutorialspoint.com
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
```

Headers

Status Code

Status Text

```
HTTP/1.1 200 OK
Date: Mon, 27 Jul 2009
12:28:53 GMT
Server: Apache/2.2.14
(Win32)
Last-Modified: Wed, 22 Jul
2009 19:15:56 GMT
Content-Length: 88
Content-Type: text/html
Connection: Closed
```

Headers

```
<html>
<body>
<h1>Hello, World!</h1>
</body>
</html>
```

Body

Lets test it!



Lets send the following requests an see what we get

HTTP

```
telnet example.com 80
```

```
GET / HTTP/1.1
```

```
Host: example.com
```

HTTPS

```
openssl s_client -connect  
example.com:443
```

```
GET / HTTP/1.1
```

```
Host: example.com
```

HTTP Requests

A request has this format

method request-target HTTP-version
request-headers
request-body

Methods:

- GET – ask server for resource at request - target
- PUT – create or replace resource at request - target
- POST – ask the resource at request - target to process the data in the appropriate way
- DELETE – delete the resource at request - target
- PATCH – update resource at request - target

HTTP Responses

A response has this format

HTTP-version status-code status-text
Response-headers
response-body

Status codes:

- 2xx Success
 - 200 OK – Most common
- 3xx Redirection
- 4xx Client Error
 - 403 Forbidden – Client does not have permission for that request
 - 404 Not Found – no resource at that location
- 5xx Server Error

HTTP, HyperText Transfer Protocol



The problem that HTTP solves

- Client and server can be any type of machine
- Software can be written in any language
- Designed for TCP/IP, but can be used over any other transport layer
- HTTP stateless protocol, meaning the server does not have to retain information or status about each client for the duration of multiple requests. However, some web applications implement states or server side sessions using for instance HTTP cookies or hidden variables within web forms.

The structure of an URL

`http://adam:password@en.wikipedia.org:80/wiki/URL?query=1#section1`

`protocol://username:password@hostname:port/path?query#fragment`

Most of the sections in the URL can be omitted with the exception of the location (hostname and path).

Application servers

So what are they all about ?

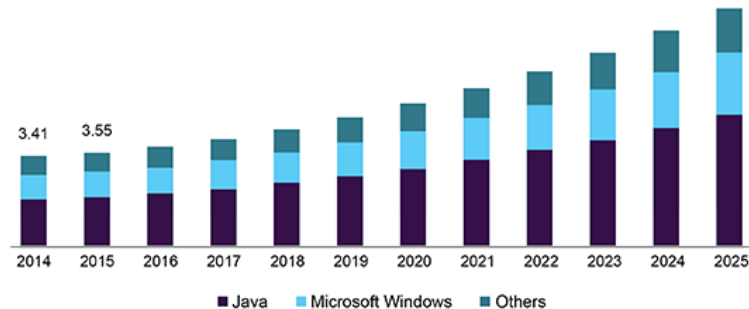
- Provide a server environment to run, manage and deploy web applications
- While they generally provide a fully featured http server, they are often exposed behind an instance of Apache or nginx
- Implement services like database handling, clustering, fail-over and load-balancing, allowing developers to focus on implementing the business logic
- There are many application servers that support Java EE / Jakarta EE. Some are open-source, others are commercial and proprietary
- Let's take a look; en.wikipedia.org/wiki/List_of_application_servers#Java

Popularity of Application servers

Used in many industries with an ever increasing market

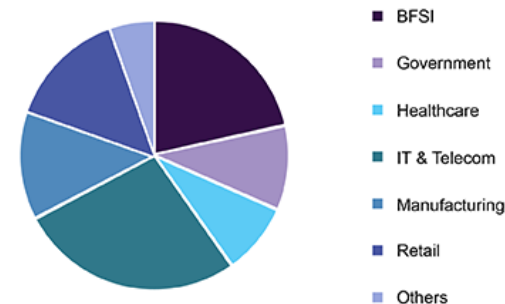
- Market size valued at USD 13.13 billion in 2018
- Java is the dominating platform

U.S. application server market size, by type, 2014 - 2025 (USD Billion)



Source: www.grandviewresearch.com

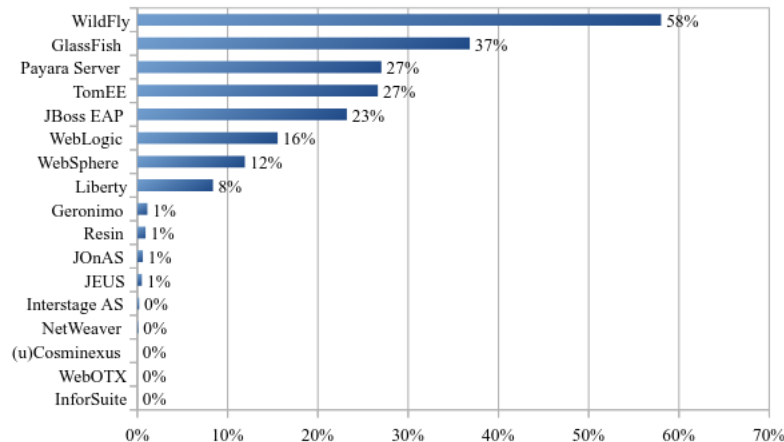
Europe application server market share, by end use, 2018 (%)



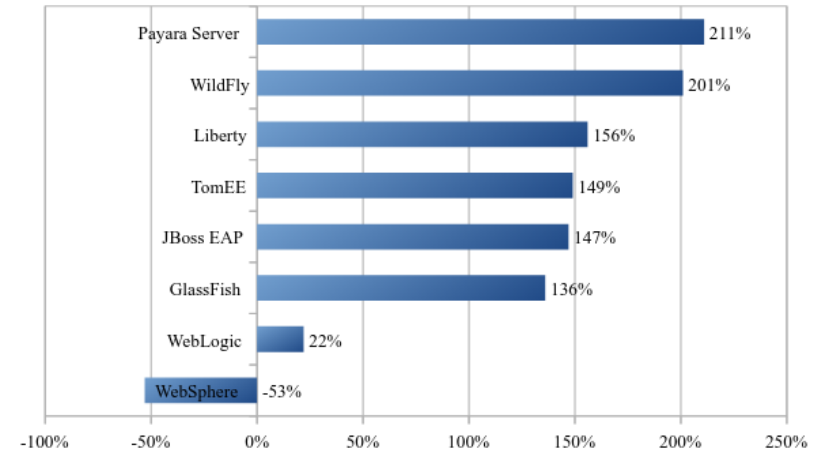
Source: www.grandviewresearch.com

Which one is the most common?

In a recent survey of the OmniFaces community we can see an indicator of which server is the most popular

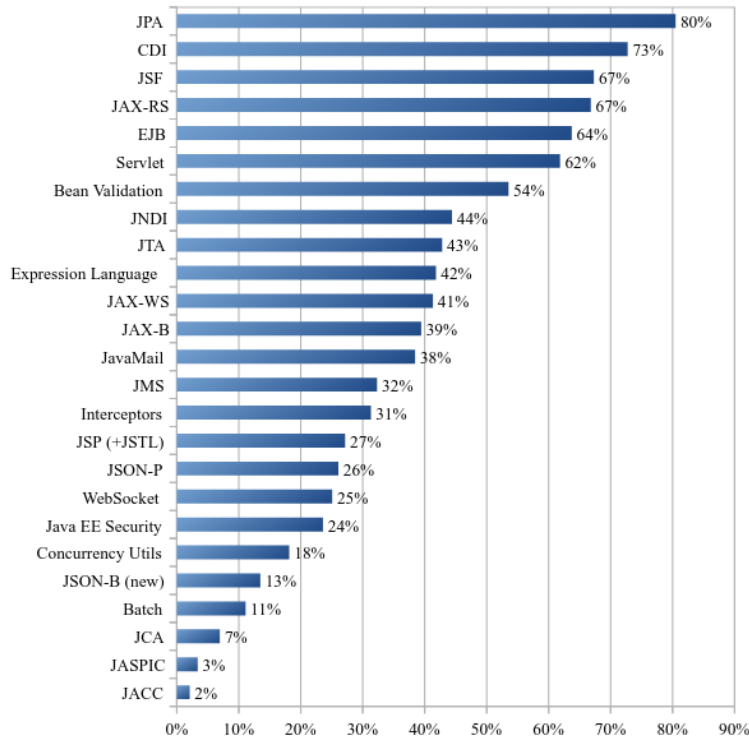


Which application servers have you recently used?



How would you rate the application servers that you've used?

What do people use?



The same survey also covers which parts of the EE stack are most commonly used

- Unfortunately, we can only scratch on the surface of a few of the API's offered by EE
- With the exception of servlets, we cover the most important parts of the more popular API's

Development environment



- We will be using NetBeans as our main development environment together with the Payara application server
- We use Maven to fetch any dependencies we need in both the project and lab assignments
- If you have problems under Windows (unfortunately, this is not uncommon), please install a Linux distribution and do your work there
- Download the latest version of NetBeans (11.2);
netbeans.apache.org/download

Who wants to be a student rep?



Send me an email

Adam Waldenberg <adam.waldenberg@ejwa.se>

About the upcoming lab sessions



- Because of scheduling issues, we do not have time to cover the HTML and CSS lecture before the first two lab sessions
- You should still be able to start with Lab assignment 2 without any major issues. Just ask us for help during the lab session if you get stuck and refer to Google and Stack-overflow (stackoverflow.com) as a reference.
- Experiment as much as you can and have fun!