

JSP & Servlets

inleiding en een eerste servlet

Voorstelling cursus

Cursus JSP & Servlets

Met behulp van JAVA webapplicaties bouwen

Duur: 2 weken

Technologieën:

JSP: Java in HTML

Servlets: HTML in JAVA code

Het beste uit 2 werelden: JSP & Servlets

Hoe gaan we te werk ?

Overlopen theorie vorige dag + vragen

PowerPoints - Cursus - demos: verwerken van theorie

Oefeningen: aanpassingen aan het project

Studie en praktijk

Simultaan en individueel opbouwen van eigen project

Overzicht voor vandaag

Leerstof voor vandaag

Een woordje uitleg over **requests** en **responses**

Het installeren en configureren van de **webserver**
(indien nog niet gebeurd)

Het aanmaken van een webapplicatie - **stappenplan**

Een **eerste servlet** aanmaken

Bespreking **architectuur** van een servlet en de diverse **methods**

Requests & responses

Browsers en webservers

Het internet verbindt computers.

Op sommige van deze computers is een **webservice** geïnstalleerd.
Daarop kunnen één of meerdere **websites** zijn geïnstalleerd.

Op sommige van deze computers is een **browser** geïnstalleerd.

Het internet verbindt de browsers en de webservers.

Gebruikers kunnen via hun browser informatie aan de webservers aanvragen.

Requests en responses

Request: browser → webserver

Response: antwoord vd webserver (pdf, html, afbeelding ...)

Computer met browser



Request



Response

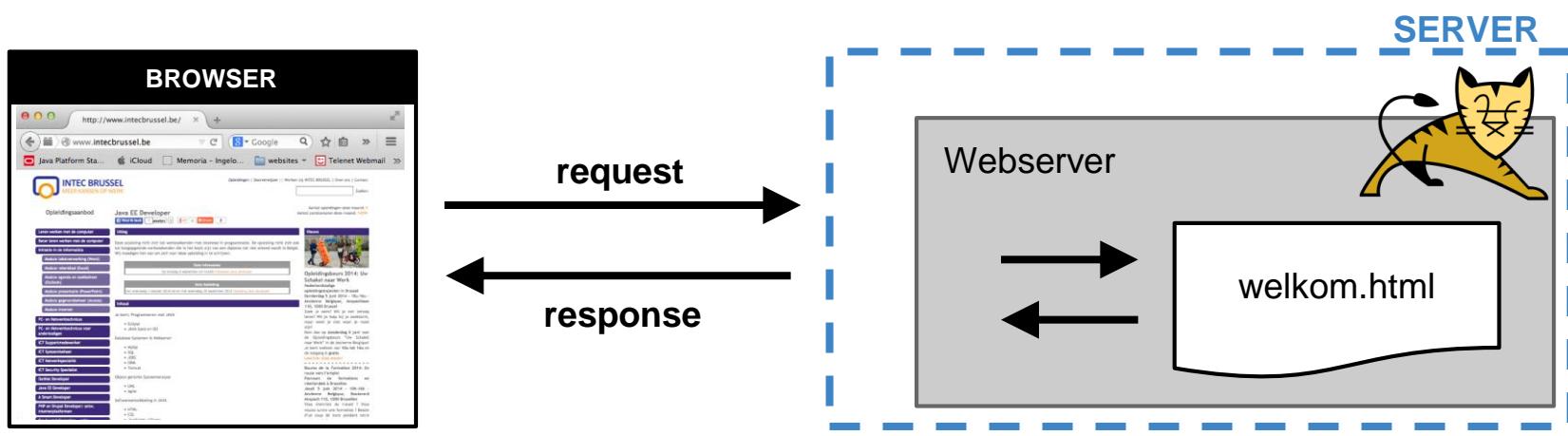


Server

Statische vs dynamische pagina's

Statische pagina:

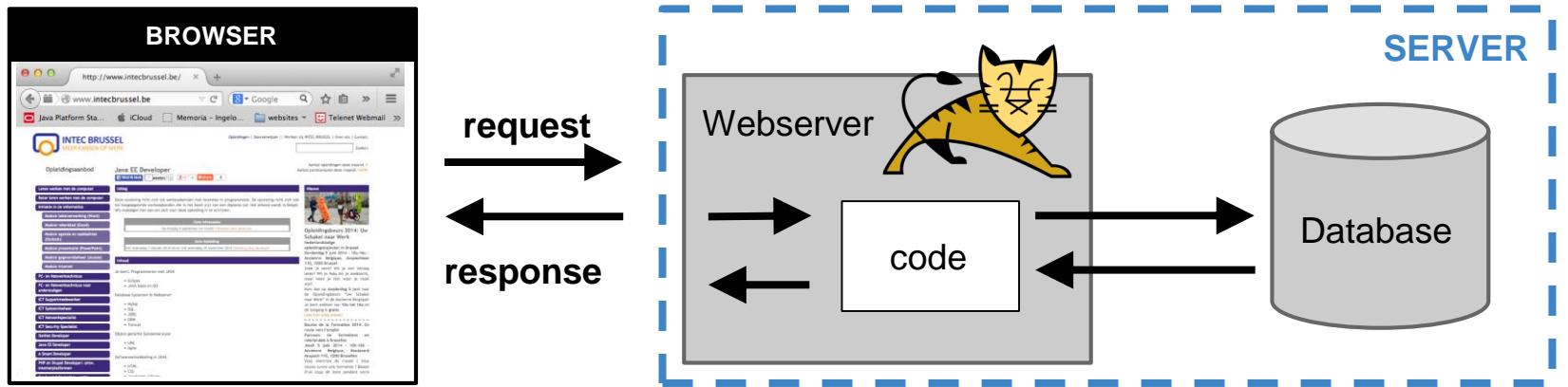
- = Een bestand op de webserver, bijvoorbeeld HTML .
- Browser → request → webserver (statische pagina)
- Webserver → response → browser (inhoud)



Statische vs dynamische pagina's

Dynamische pagina:

Een dynamische pagina → programmacode op server
request → url → webserver voert code uit → response naar de browser.



Java webservers:

Niet alle webservers ondersteunen dynamische pagina's geschreven in Java. De webserver van Microsoft (IIS) ondersteunt bijvoorbeeld geen Java. Maar er bestaan voldoende webservers die wél Java ondersteunen.

Voorbeelden

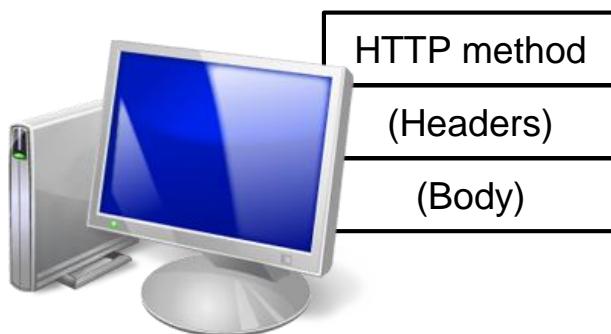
Webserver	Organisatie of firma die deze webserver maakt
Tomcat	Apache (deze gebruiken wij)
JBoss	JBoss
GlassFish	Oracle
WebSphere	IBM

Onderdelen van een request

Een **request** bevat drie onderdelen

1. **HTTP method** (verplicht onderdeel)
2. **Headers** (optioneel onderdeel)
3. **Body** (optioneel onderdeel)

Computer met browser



Request

Server



1. De HTTP Method van een request

Een HTTP method geeft het soort request aan:

GET request:

Te gebruiken wanneer men data opvraagt. (Parameters in URL)

POST request:

Men gebruikt POST voor alle andere soorten requests. (Parameters in Body)

Om data te verwijderen of toe te voegen.

Om data te wijzigen

Om in te loggen

Om een mail te versturen...

2. De headers van een request

Bevatten informatie over de browser.

Headers hebben een naam en een waarde.

Voorbeelden:

1. **Content-Type: text/html**: Meer informatie over het type bericht!

2. De header met de naam **accept-language** bevat de taal en het land van de gebruiker. Bij een Nederlandstalige Belgische gebruiker is dit: nl-be.

3. *De body van een request*

GET request:

Geen body

POST request:

Kan wel een body bevatten. Deze bevat data die de website nodig heeft om de request te verwerken.

De body heeft volgende structuur:

Ieder stukje data in de body heeft **een naam en een waarde**.

Tussen de naam en de waarde staat het teken =.

Tussen de stukjes data staat het teken &.

De onderdelen van een response

Een response bevat ook drie onderdelen

1. **Status code** (verplicht onderdeel)
2. **Headers** (optioneel onderdeel)
3. **Body** (optioneel onderdeel)

Computer met browser



Response

Status code
(Headers)
(Body)



Server

1. De status code van een response

200 (OK)

404 (Not Found)

De webserver vond de URL in de request niet.

401 (Unauthorized)

Om een request te doen naar de URL moet de gebruiker ingelogd zijn.

5XX (Fout bij de webserver)

2. De headers van een response

Bevatten informatie over de response.

Voorbeeld:

De header met de naam **content-type** bevat het type data in de body. Dit type data heet het MIME-type.

Populaire MIME-types:

text/html

De body bevat HTML

text/css

De body bevat CSS

text/javascript

De body bevat JavaScript

image/jpeg

De body bevat een afbeelding in JPEG

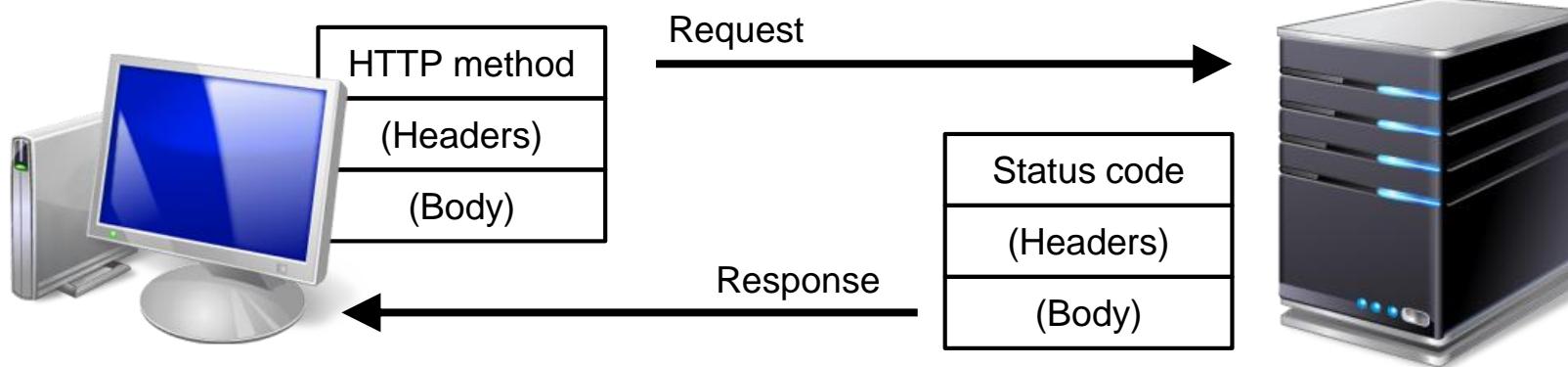
application/pdf

De body bevat PDF

3. De body van een response

Bevat data die de request vraagt. Dit kan HTML zijn, een afbeelding, ...

Samenvatting



De Tomcat webserver

Installatie - SAMEN

Downloaden

<http://tomcat.apache.org>

The screenshot shows a Microsoft Edge browser window with multiple tabs open at the top. The active tab displays the Apache Tomcat download page (<http://tomcat.apache.org/download-80.cgi>). The page features the Apache logo (a yellow cat) and the text "Apache Tomcat®". A banner for "TomcatCon May 16-18, 2017 MIAMI, FL." is visible, along with the Apache Software Foundation logo. The main content area is titled "Tomcat 8 Software Downloads" and includes sections for "Quick Navigation" (with links to KEYS, 8.5.11, 8.0.41, Browse, and Archives), "Release Integrity" (warning about file verification), "Mirrors" (listing the current mirror as <http://apache.cu.be/>), and "8.5.11" (with a note to read the README file). At the bottom, there's a section for "Binary Distributions". The taskbar at the bottom shows several other open files like "LVB08 JSP & Servle....zip", "Frituur Frida.pptx", and "FrituurFrida.docx".

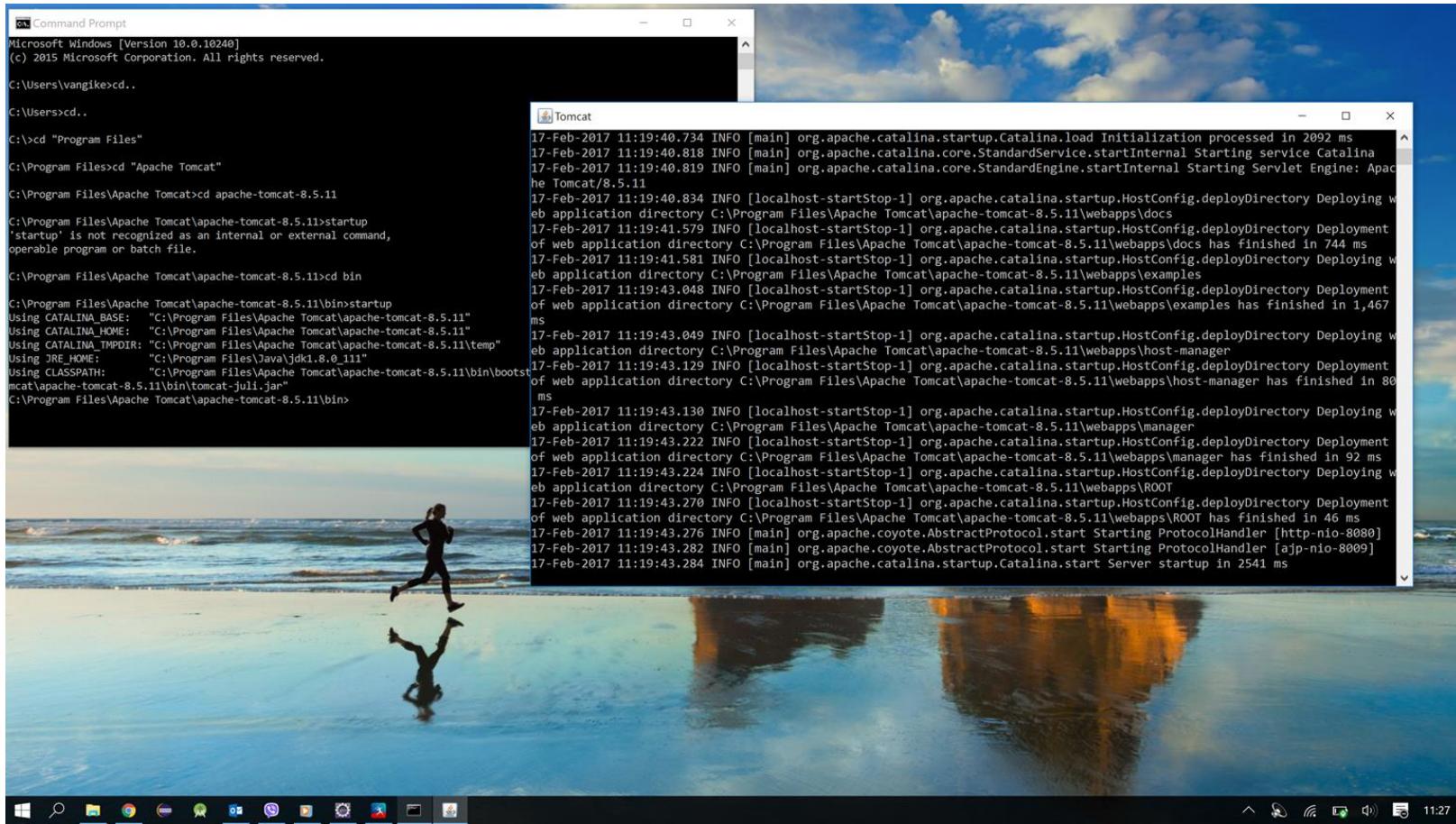
Uitpakken in een map naar keuze

Voorbeeld: C:\Program Files\Apache Tomcat

De submappen structuur van Tomcat

bin	Bevat de binaries met o.a. startup.bat en shutdown.bat
conf	Bevat de instellingen zoals: server.xml , web.xml , context.xml en tomcat-users.xml
lib	Bevat de JAR bestanden van het Tomcat systeem die door alle webapps bereikbaar zijn. Hier kan je eventueel ook externe JAR files plaatsen zoals de MySQL JDBC Driver.
logs	Bevat de log files . Je kan hier de foutberichten checken.
work	Werkmap dat gebruikt wordt door JSP voor de communicatie met de servlets .
temp	Tijdelijke bestanden.
webapps	In deze map (of andere) komen de websites terecht. (zie volgende dia's)

Tomcat opstarten vanuit de cmd-line



Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

```
C:\Users\vangike>cd..  
C:\Users>cd..  
C:\>cd "Program Files"  
C:\Program Files>cd "Apache Tomcat"  
C:\Program Files\Apache Tomcat>cd apache-tomcat-8.5.11  
C:\Program Files\Apache Tomcat\apache-tomcat-8.5.11>startup  
'startup' is not recognized as an internal or external command,  
operable program or batch file.  
C:\Program Files\Apache Tomcat\apache-tomcat-8.5.11>cd bin  
C:\Program Files\Apache Tomcat\apache-tomcat-8.5.11>startup  
Using CATALINA_BASE:  "C:\Program Files\Apache Tomcat\apache-tomcat-8.5.11"  
Using CATALINA_HOME:  "C:\Program Files\Apache Tomcat\apache-tomcat-8.5.11"  
Using CATALINA_TMPDIR:  "C:\Program Files\Apache Tomcat\apache-tomcat-8.5.11\temp"  
Using JRE_HOME:  "C:\Program Files\Java\jdk1.8.0_111"  
Using CLASSPATH:  "C:\Program Files\Apache Tomcat\apache-tomcat-8.5.11\bin\bootstrap.jar;  
C:\Program Files\Apache Tomcat\apache-tomcat-8.5.11\bin\tomcat-juli.jar"  
C:\Program Files\Apache Tomcat\apache-tomcat-8.5.11>
```

Tomcat

```
17-Feb-2017 11:19:40.734 INFO [main] org.apache.catalina.startup.Catalina.load Initialization processed in 2092 ms  
17-Feb-2017 11:19:40.818 INFO [main] org.apache.catalina.core.StandardService.startInternal Starting service Catalina  
17-Feb-2017 11:19:40.819 INFO [main] org.apache.catalina.core.StandardEngine.startInternal Starting Servlet Engine: Apache Tomcat/8.5.11  
17-Feb-2017 11:19:40.834 INFO [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory C:\Program Files\Apache Tomcat\apache-tomcat-8.5.11\webapps\docs  
17-Feb-2017 11:19:41.579 INFO [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory C:\Program Files\Apache Tomcat\apache-tomcat-8.5.11\webapps\docs has finished in 744 ms  
17-Feb-2017 11:19:41.581 INFO [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory C:\Program Files\Apache Tomcat\apache-tomcat-8.5.11\webapps\examples  
17-Feb-2017 11:19:43.048 INFO [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory C:\Program Files\Apache Tomcat\apache-tomcat-8.5.11\webapps\examples has finished in 1,467 ms  
17-Feb-2017 11:19:43.049 INFO [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory C:\Program Files\Apache Tomcat\apache-tomcat-8.5.11\webapps\host-manager  
17-Feb-2017 11:19:43.129 INFO [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory C:\Program Files\Apache Tomcat\apache-tomcat-8.5.11\webapps\host-manager has finished in 80 ms  
17-Feb-2017 11:19:43.130 INFO [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory C:\Program Files\Apache Tomcat\apache-tomcat-8.5.11\webapps\manager  
17-Feb-2017 11:19:43.222 INFO [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory C:\Program Files\Apache Tomcat\apache-tomcat-8.5.11\webapps\manager has finished in 92 ms  
17-Feb-2017 11:19:43.224 INFO [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory C:\Program Files\Apache Tomcat\apache-tomcat-8.5.11\webapps\ROOT  
17-Feb-2017 11:19:43.276 INFO [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory C:\Program Files\Apache Tomcat\apache-tomcat-8.5.11\webapps\ROOT has finished in 46 ms  
17-Feb-2017 11:19:43.276 INFO [main] org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler [http-nio-8080]  
17-Feb-2017 11:19:43.282 INFO [main] org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler [ajp-nio-8009]  
17-Feb-2017 11:19:43.284 INFO [main] org.apache.catalina.startup.Catalina.start Server startup in 2541 ms
```

Checken of Tomcat opgestart is

localhost:8080

The screenshot shows a Mozilla Firefox browser window displaying the Apache Tomcat 8.0.0-RC10 welcome page. The title bar reads "Apache Tomcat/8.0.0-RC10 - Mozilla Firefox". The address bar shows "localhost:8080". The page content includes the Apache logo, the Apache Software Foundation link, and a message: "If you're seeing this, you've successfully installed Tomcat. Congratulations!". It also features a cartoon cat icon, recommended reading links, and developer quick start sections. Below the main content are three boxes: "Managing Tomcat", "Documentation", and "Getting Help".

Apache Tomcat/8.0.0-RC10 - Mozilla Firefox

Bestand Beyerken Beeld Geschiedenis Bladwijzers Extra Help

Apache Tomcat/8.0.0-RC10

localhost:8080

Meest bezocht Aan de slag Java Platform Standard... iCloud Memoria - Ingelogd al... Handleiding HTML - D... IVO Brugge - Cursusw... QuestBase | Create on...

Home Documentation Configuration Examples Wiki Mailing Lists Find Help

Apache Tomcat/8.0.0-RC10

The Apache Software Foundation <http://www.apache.org/>

If you're seeing this, you've successfully installed Tomcat. Congratulations!

TM

Recommended Reading:

[Security Considerations HOW-TO](#)

[Manager Application HOW-TO](#)

[Clustering/Session Replication HOW-TO](#)

Developer Quick Start

[Tomcat Setup](#)

[First Web Application](#)

[Realms & AAA](#)

[JDBC DataSources](#)

[Examples](#)

[Servlet Specifications](#)

[Tomcat Versions](#)

Managing Tomcat

For security, access to the [manager webapp](#) is restricted. Users are defined in:
\${CATALINA_HOME}/conf/tomcat-users.xml

In Tomcat 8.0 access to the manager

Documentation

[Tomcat 8.0 Documentation](#)

[Tomcat 8.0 Configuration](#)

[Tomcat Wiki](#)

Getting Help

[FAQ and Mailing Lists](#)

The following mailing lists are available:

announce@tomcat.apache.org

Important announcements, releases, security

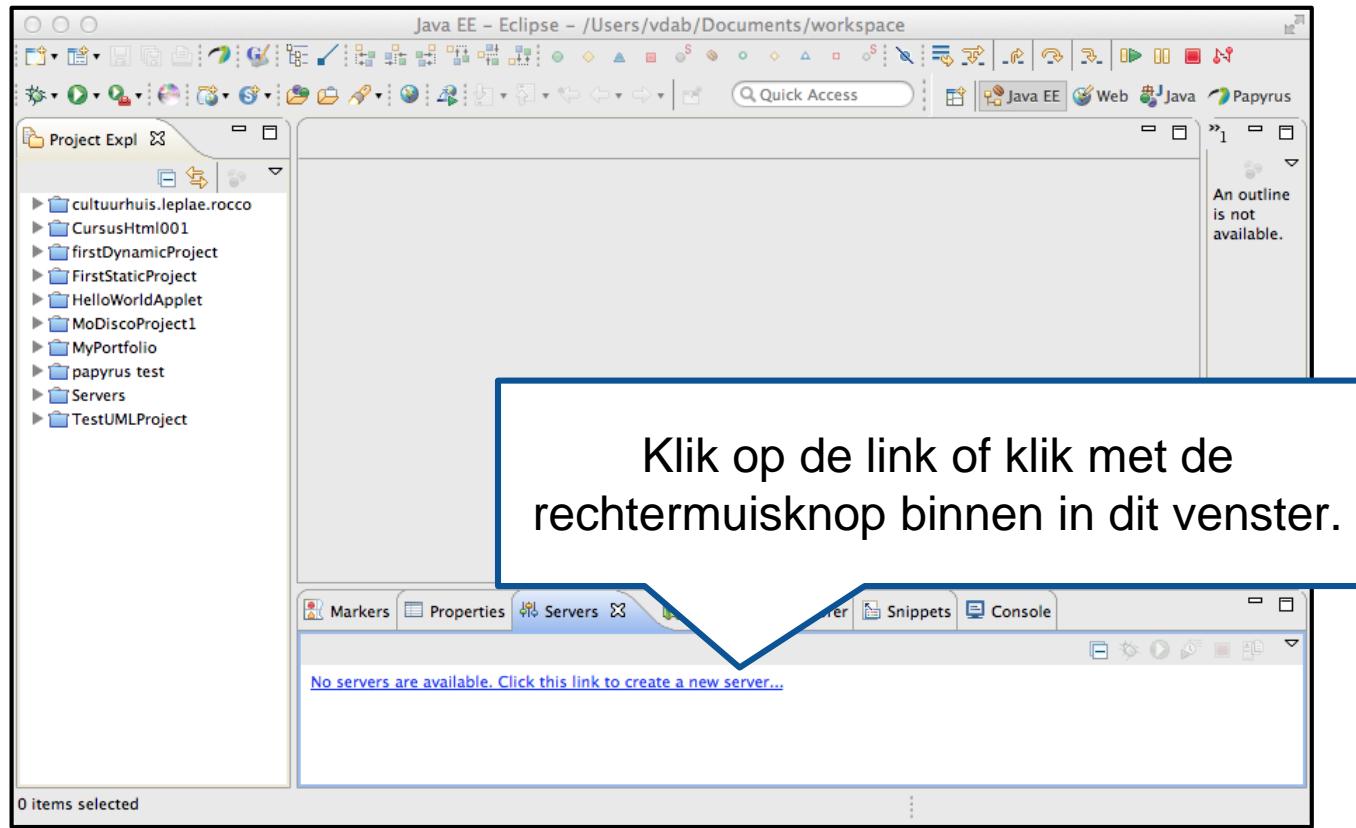
Server Status

Manager App

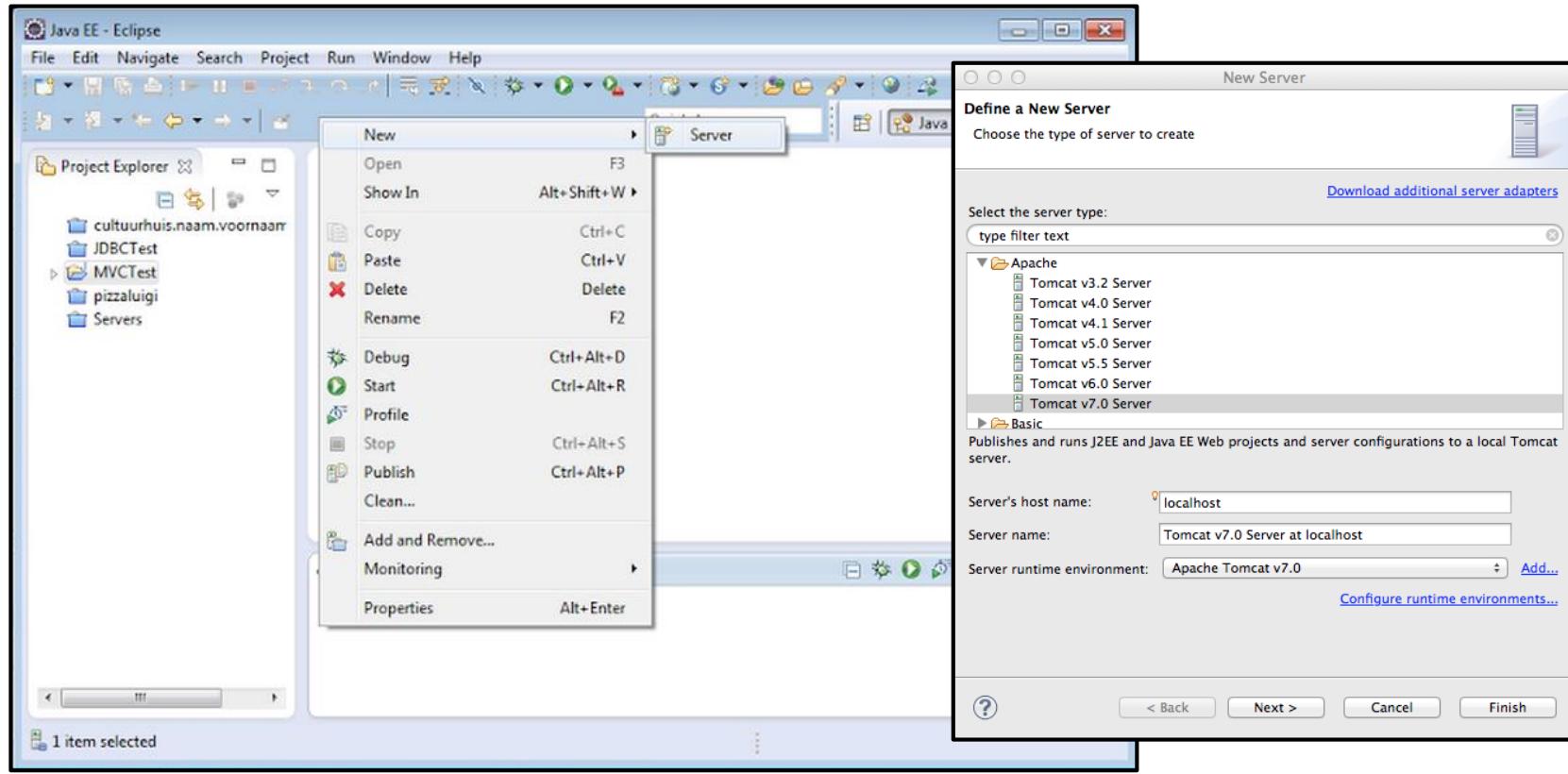
Host Manager

Integratie in Eclipse: SAMEN

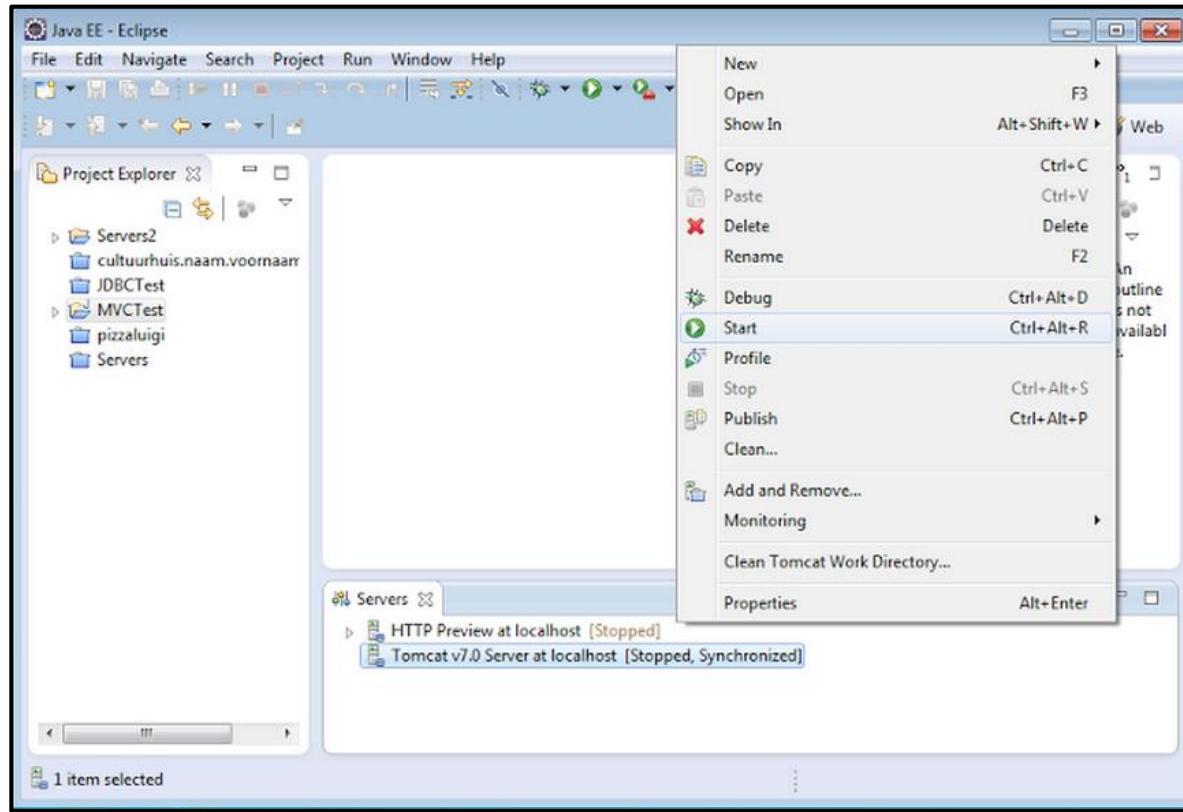
Voeg een nieuwe server toe



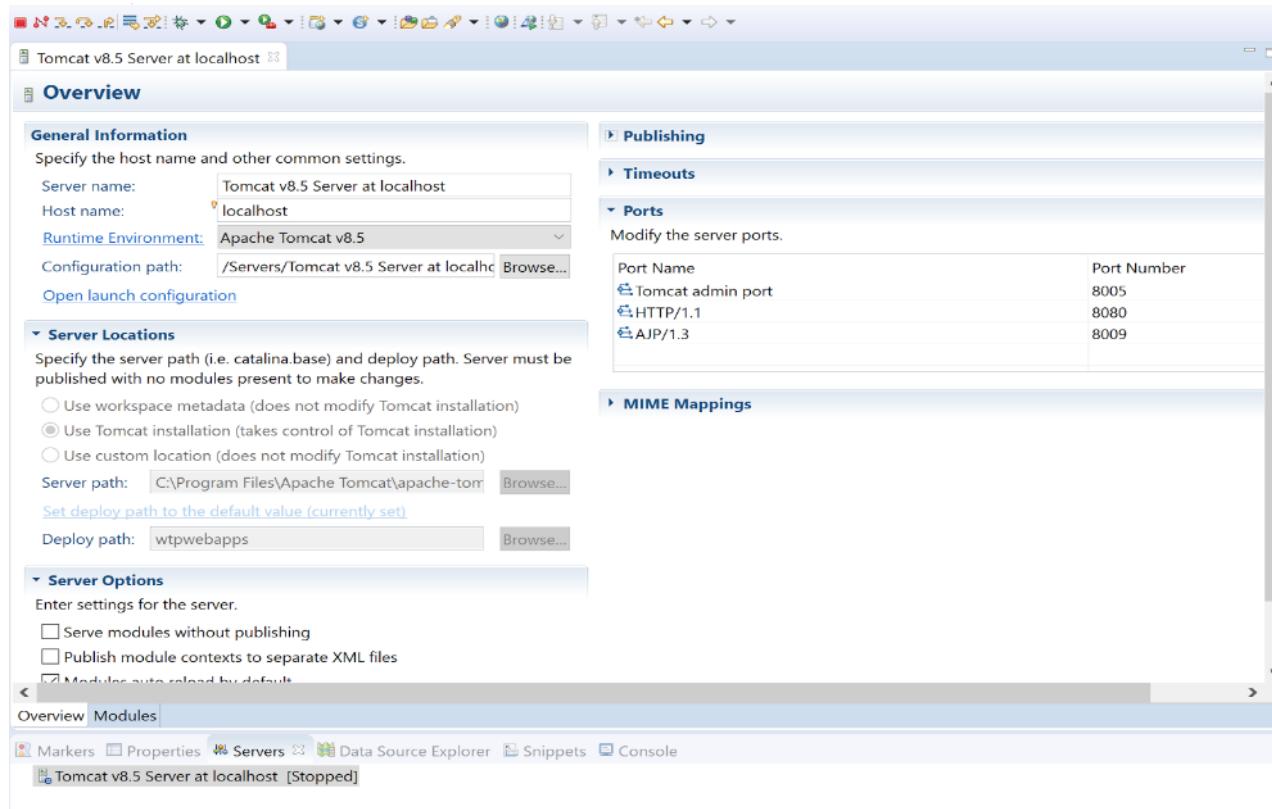
Voeg een nieuwe server toe



Starten van de Tomcat server

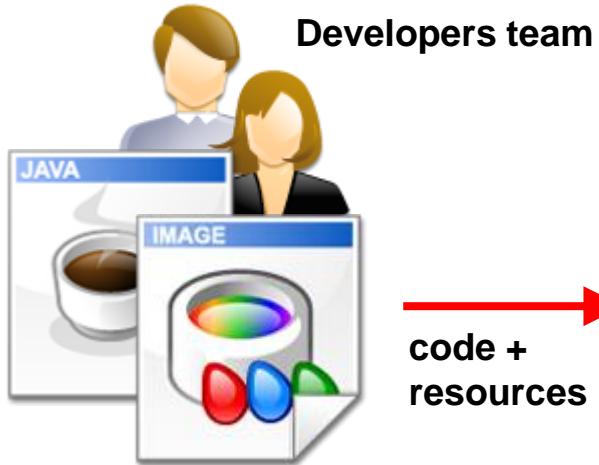


Use Tomcat Installation



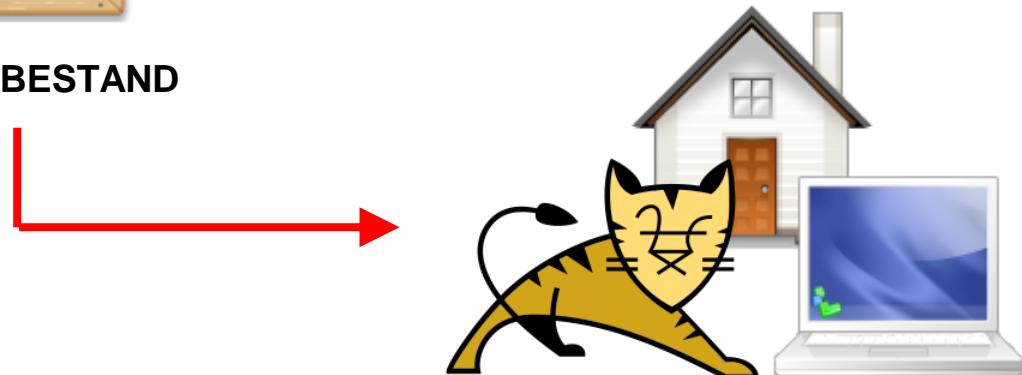
Deployment van een website

code wordt verpakt in een war file



Als de code is geschreven, wordt deze samen met alle resources verpakt in een .war bestand.

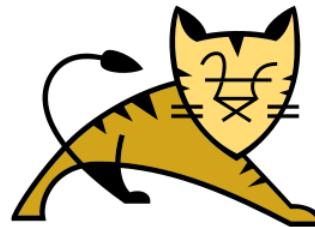
Dan wordt deze naar de webserver gestuurd.



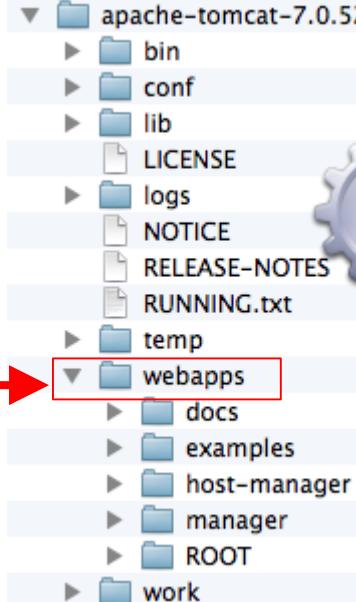
Uploaden....



.WAR BESTAND



Tomcat bouwt
automatisch de website
op. (= deploy)

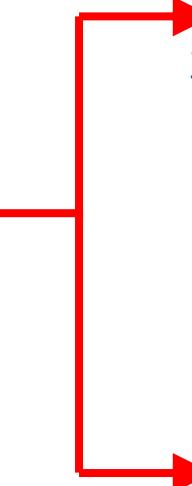
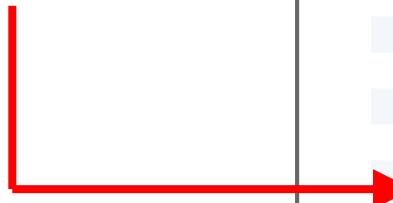


www.pizzamamamia.be



localhost:8080

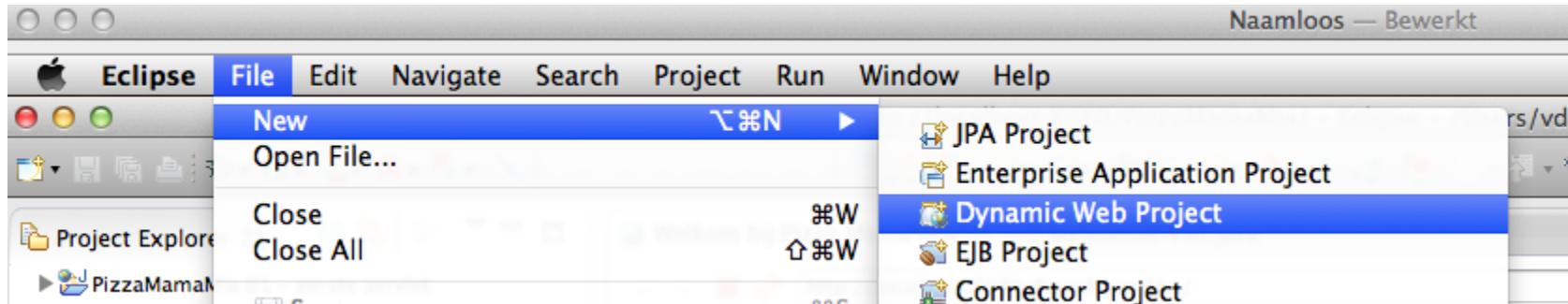
Uploaden naar de map
webapps van de Tomcat
directory.
(manueel of via uw IDE)



Project PizzaMamaMia

Het project aanmaken

Een nieuw dynamisch web project aanmaken

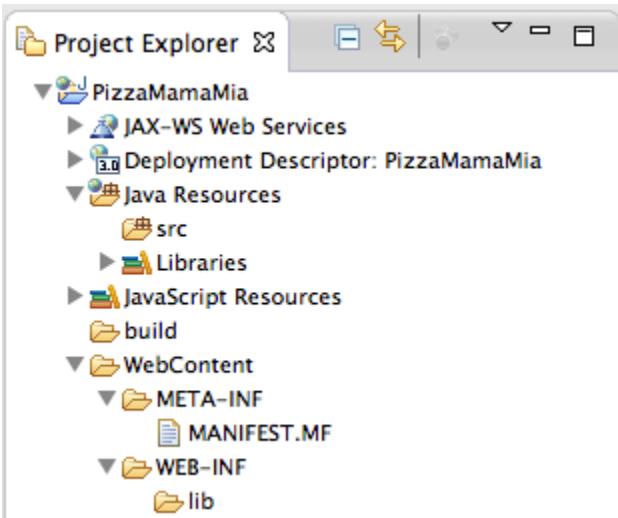


Maak een nieuw dynamisch project aan...

[File] [New] [Dynamic Web Project]

project naam: PizzaMamaMia
alle andere opties ongewijzigd laten.

De mappenstructuur

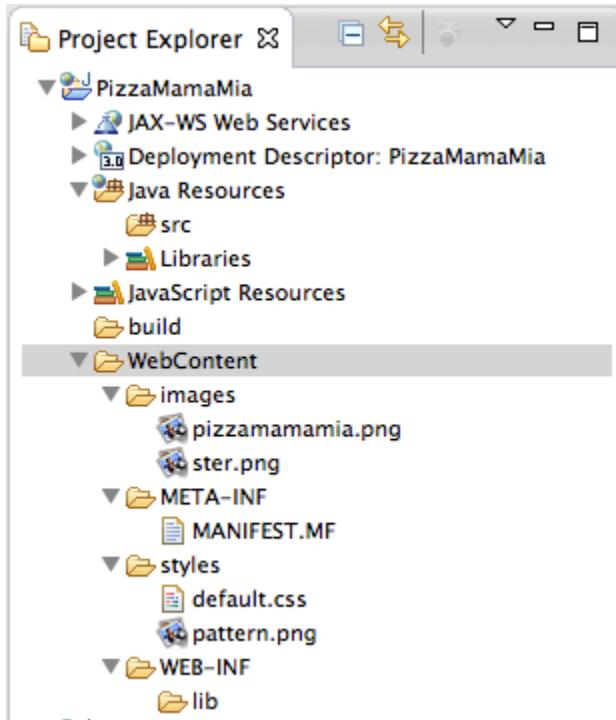


Zo ziet de structuur van het project eruit:

Java Resources zal de java bronbestanden van je website bevatten. In de submap **src** komen de packages te staan

WebContent zal onderdelen van je website bevatten die geen Java sources zijn: HTML pagina's, afbeeldingen, CSS pagina's, JSP's...

Aanvullende mappen



Voeg de submappen **img** en **styles** toe aan de **WebContent** folder.

In de map **img** stop je volgende bestanden:
pizzamamamia.png

In de map **styles** stop je volgende bestanden:
style.css

Een statische pagina toevoegen

index.html

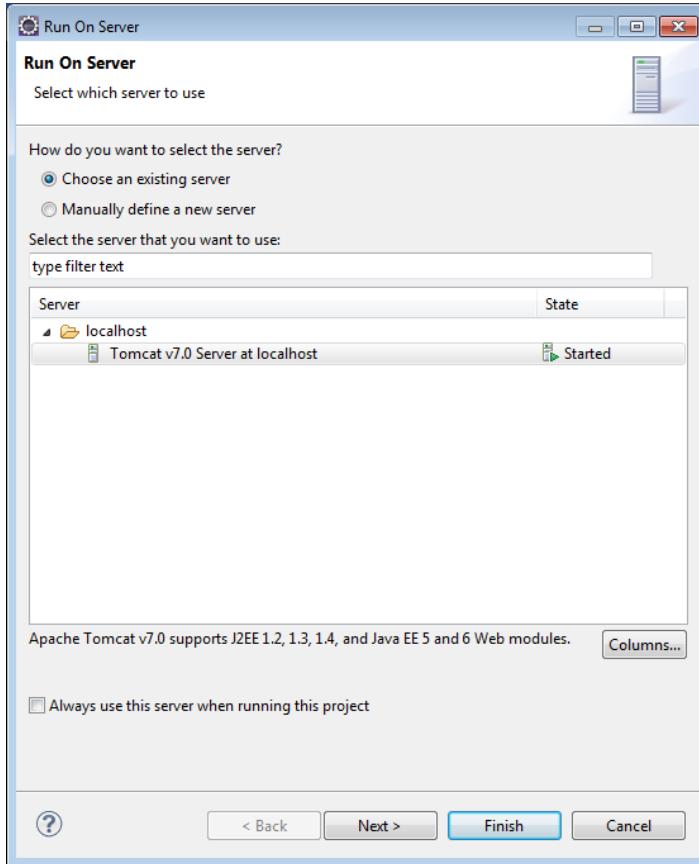
The screenshot shows the Eclipse IDE interface with a Java web project named "PizzaMamaMia".

- Project Explorer:** Shows the project structure:
 - PizzaMamaMia
 - JAX-WS Web Services
 - Deployment Descriptor: PizzaMamaMia
 - Java Resources
 - src
 - Libraries
 - JavaScript Resources
 - build
 - WebContent
 - images
 - pizzamamamia.png
 - ster.png
 - META-INF
 - MANIFEST.MF
 - styles
 - default.css
 - pattern.png
 - WEB-INF
 - lib
- index.html:** Code editor showing the HTML file content:

```
<!DOCTYPE html>
<html>
<head>
<title>startpagina</title>
</head>
<body>
    Hallo!
</body>
</html>
```
- startpagina:** Browser view showing the output of the index.html file: "Hallo!"

Maak in de folder **WebContent** een kleine html pagina met de naam "**index.html**" aan.

Programma op de server laten draaien



1. Klik met de rechtermuisknop op je project in de Project Explorer.
2. Kies **Run As, Run on Server**.
3. Eclipse stelt de Tomcat server voor die je eerder in Eclipse configureerde.
4. Plaats eventueel een vinkje bij **Always use this server** ...
5. Kies **Finish**.

De Deployment Descriptor: web.xml

De Deployment Descriptor: web.xml

Maakt deel uit van de servlet standard voor web applicaties

Wordt gebruikt om aan te duiden

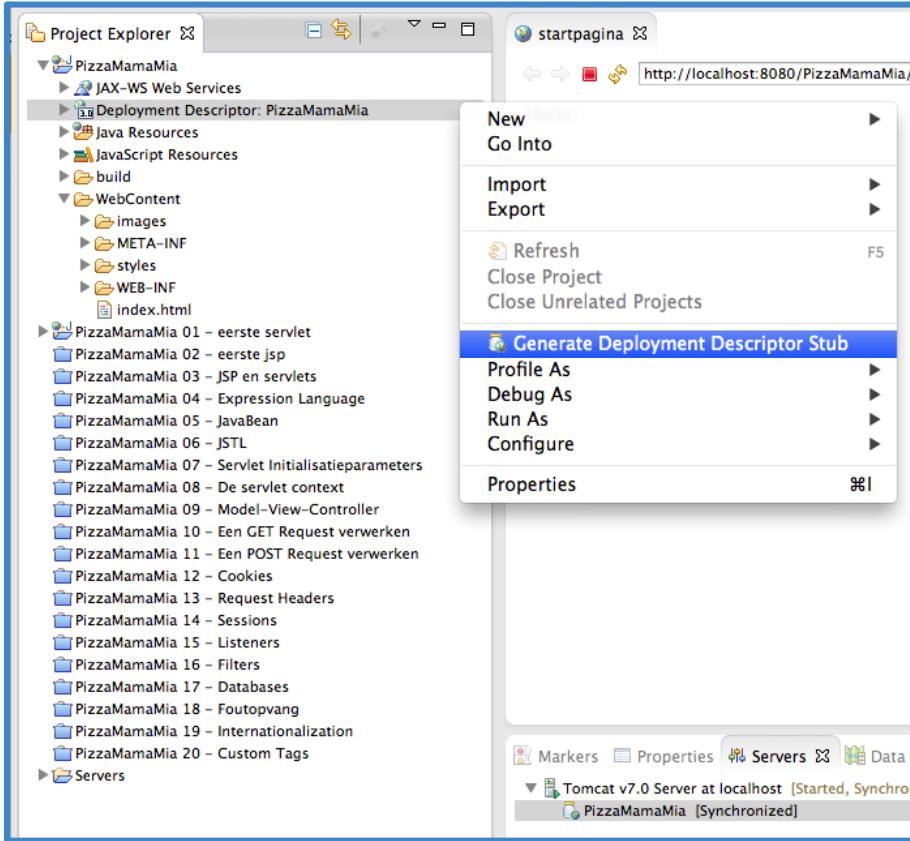
- welke URLs bij welke servlets horen

- welke URLs een autorisatie vereisen

- welke URLs als welkomspagina ingesteld zijn

...

Web.xml aanmaken



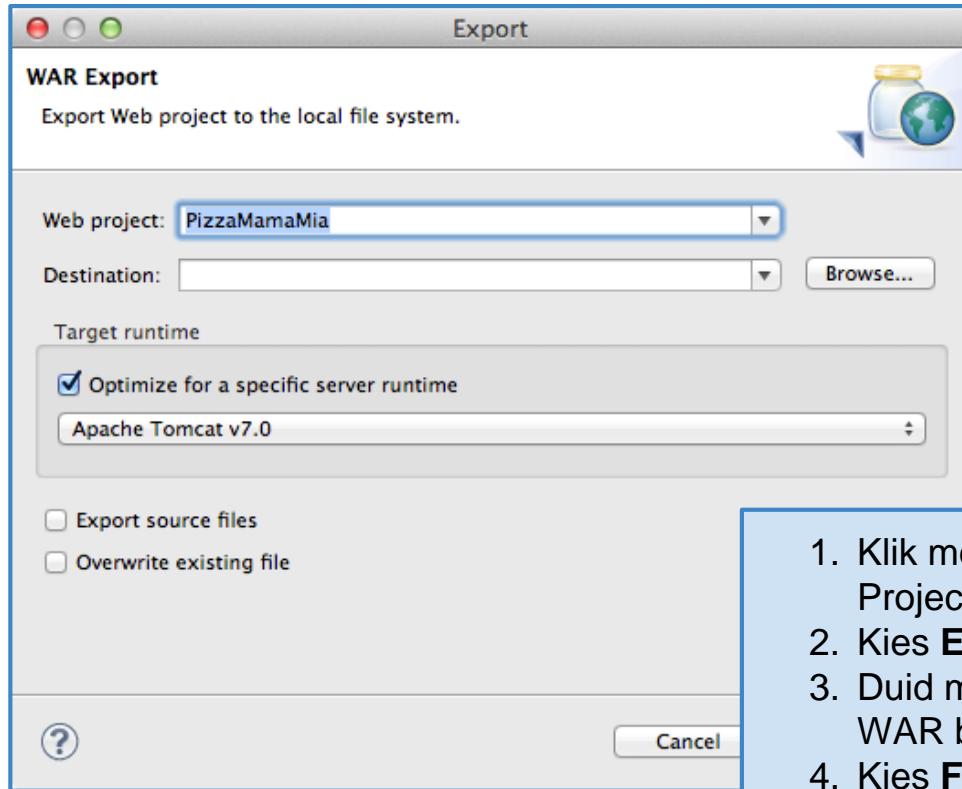
The screenshot shows the content of the generated 'web.xml' file. The XML code defines a web application with a display name 'PizzaMamaMia' and a welcome file list containing multiple entries for 'index.html', 'index.htm', 'index.jsp', and 'default.html' through 'default.jsp'.

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd" display-name="PizzaMamaMia">
    <welcome-file-list>
        <welcome-file>index.html</welcome-file>
        <welcome-file>index.htm</welcome-file>
        <welcome-file>index.jsp</welcome-file>
        <welcome-file>default.html</welcome-file>
        <welcome-file>default.htm</welcome-file>
        <welcome-file>default.jsp</welcome-file>
    </welcome-file-list>
</web-app>
```

Verwijder alle welcome-file's met
uitzondering van de eerste

WAR bestand aanmaken

WAR bestand aanmaken



Een WAR file is een bestand waarin alle bestanden van een website verpakt zitten.

Een WAR file kan op om het even welke Java webserver geïnstalleerd worden

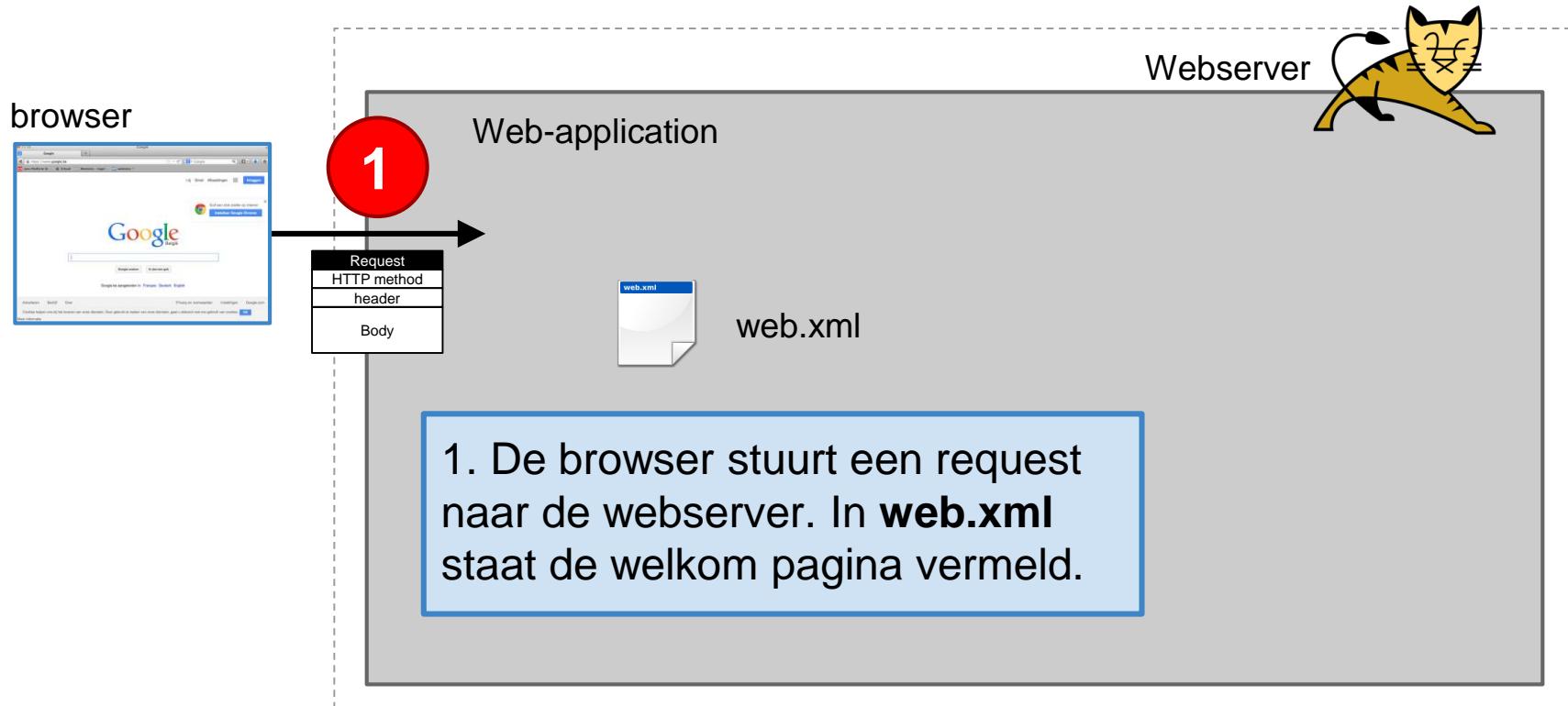
1. Klik met de rechtermuisknop op je project in de Project Explorer.
2. Kies **Export, WAR file**.
3. Duid met **Browse** een directory waarin Eclipse het WAR bestand plaatst.
4. Kies **Finish**.

Servlets

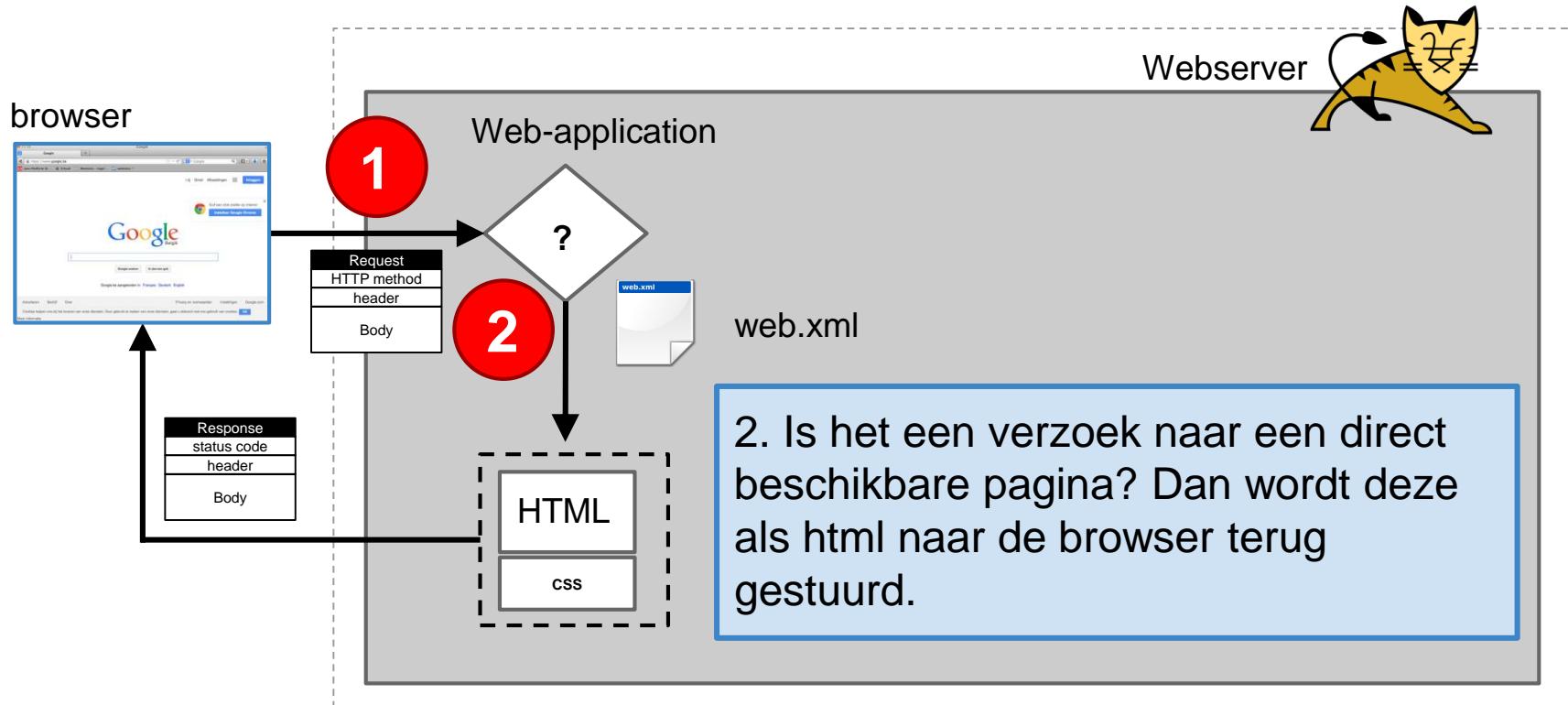
Wat is een servlet ?

- = Een **Java class** die binnen een webcontainer op een server draait. In ons geval is dat de Tomcat server.
- Maakt gebruik van diensten die de webcontainer biedt.
 - o.a. het afhandelen van de communicatie met de client. (meestal via HTTP).
- Kan worden gebruikt om o.a. invulpagina's (formulieren) op een website te verwerken, grafieken te maken of de toegang tot een website te regelen.

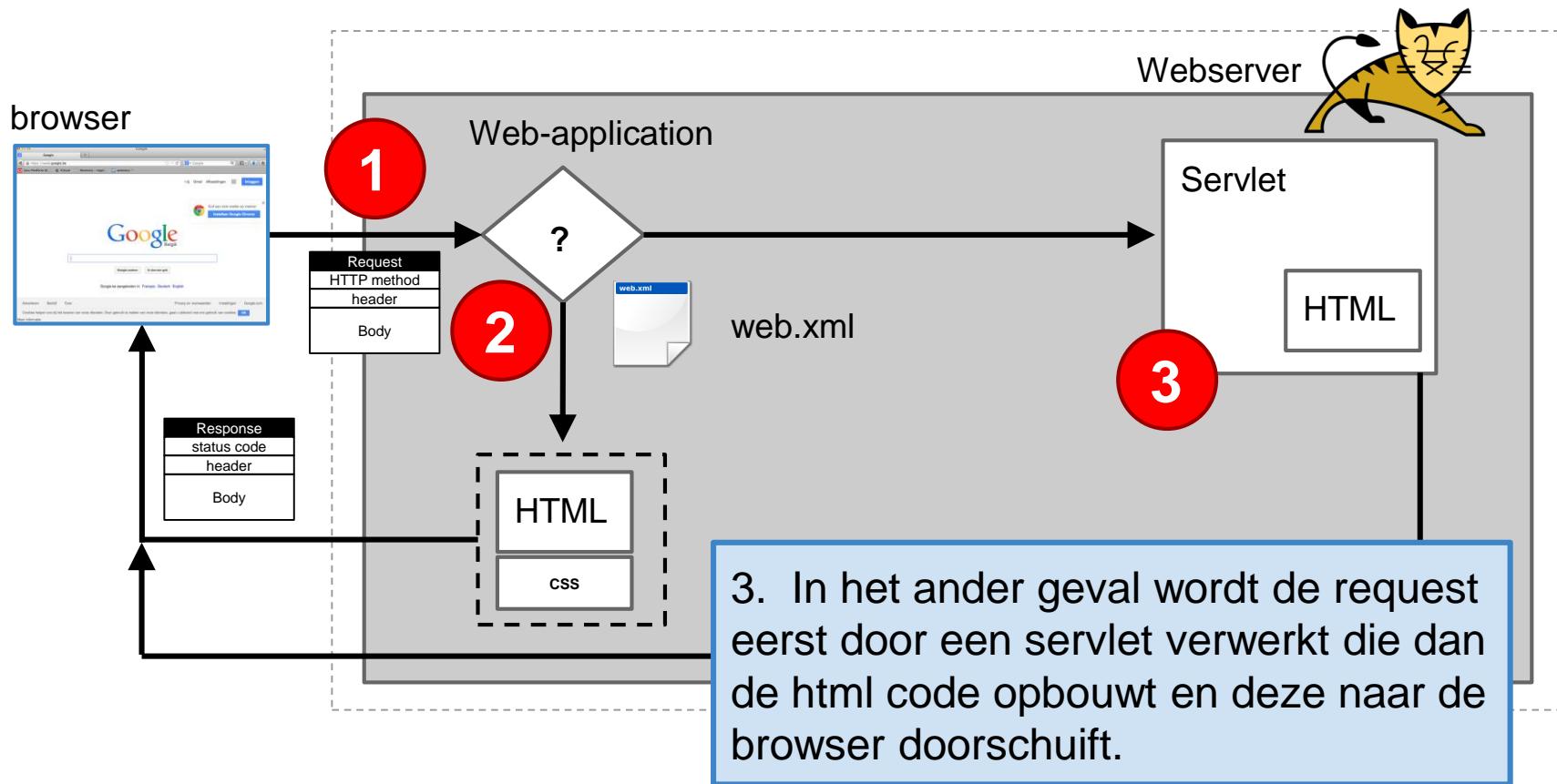
Schematisch overzicht: servlets



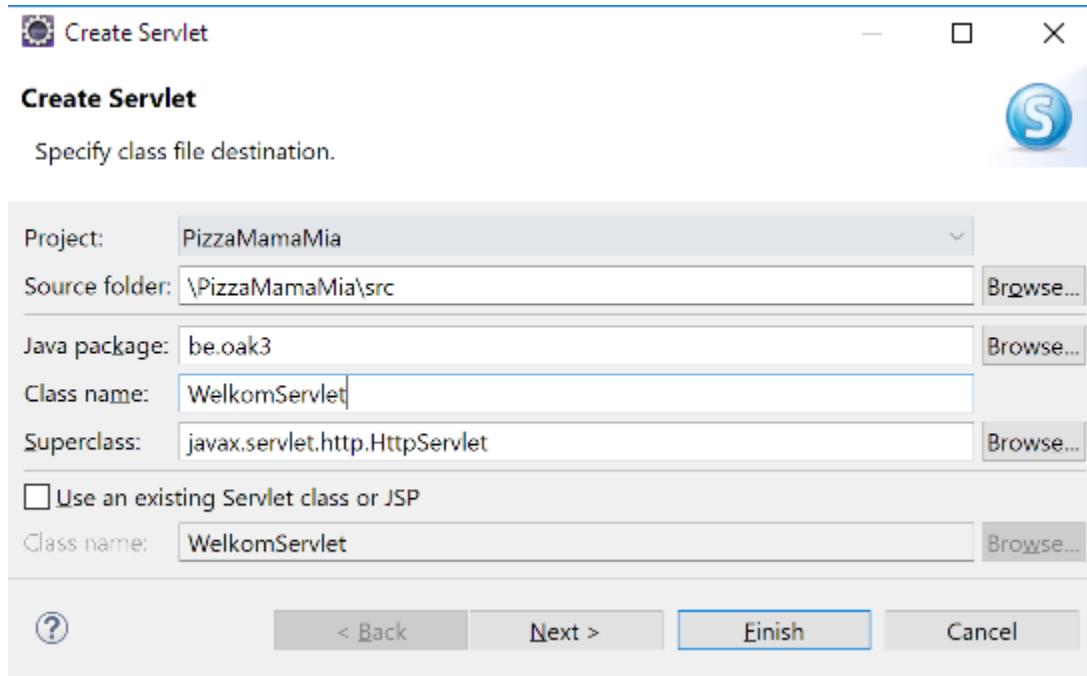
Schematisch overzicht: servlets



Schematisch overzicht: servlets

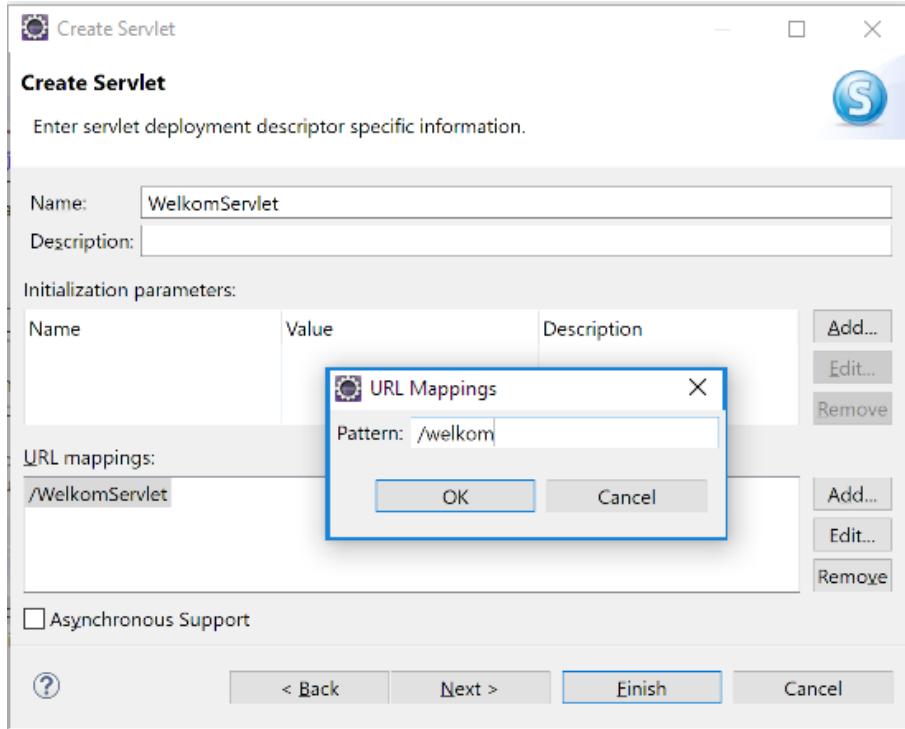


Wizard servlet aanmaken



- Klik met de rechtermuisknop op het project en kies **New, Servlet**
- Tik **be.oak3** bij **Java package**
- Tik **WelkomServlet** bij **Class name** en je kies Next

Wizard servlet aanmaken



- Selecteer **/WelkomServlet** in URL Mappings en kies Edit
- Tik **/welkom** bij Pattern en kies OK.
- Kies Finish.

WelkomServlet

```
package be.intecbrussel;

import java.io.IOException; 

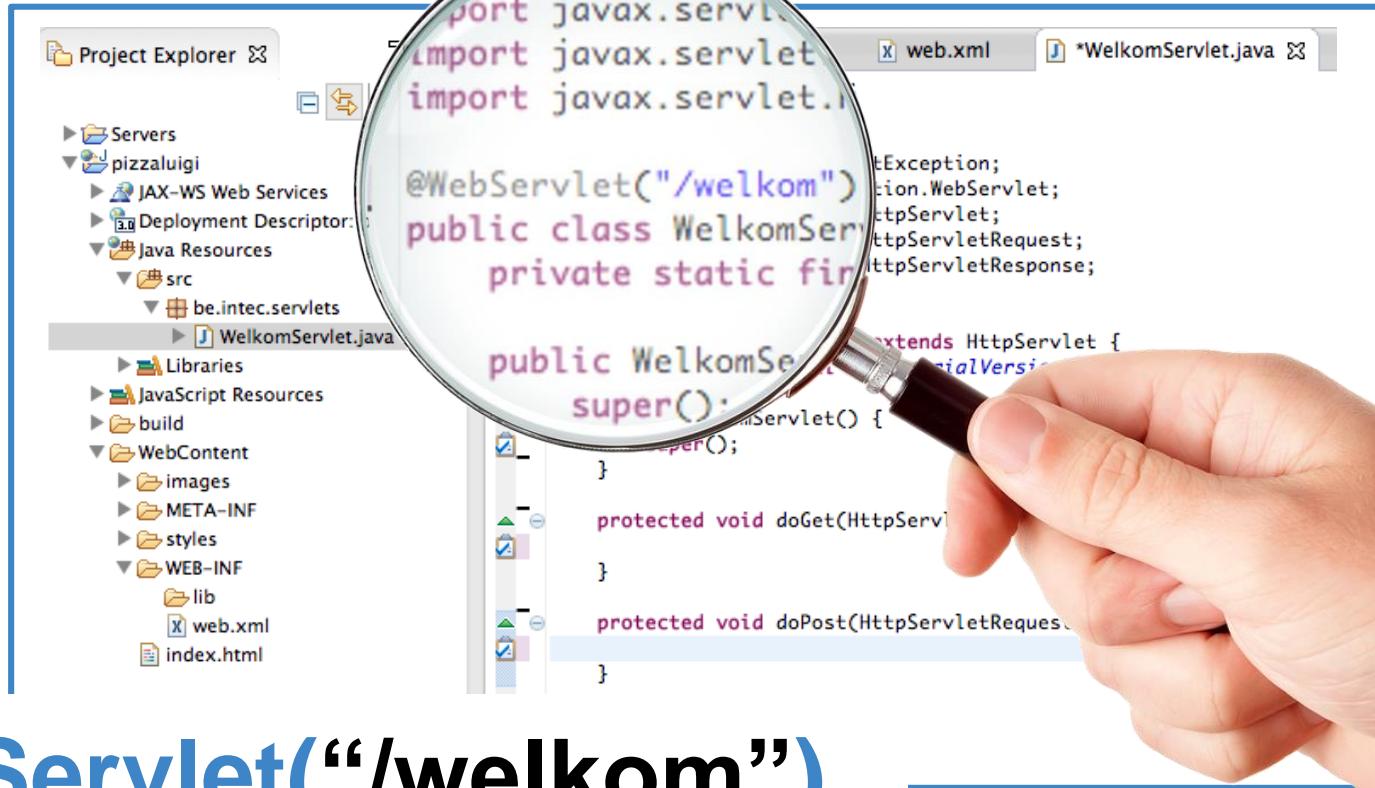
@WebServlet("/welkom")
public class WelkomServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    public WelkomServlet() {
        super();
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    }
}
```

Annotation @WebServlet



@WebServlet("/welkom")

@WebServlet("/welkom")

Annotation

@WebServlet is een annotation, hier met een URL als parameter. De annotation vertelt de webserver dat de volgende class een **Servlet** is die bij deze (relatieve) URL hoort.

@WebServlet("welkom")

Relatieve URL ten opzichte van de base URL
van je website

De webserver stuurt request naar **/welkom** door naar deze servlet.

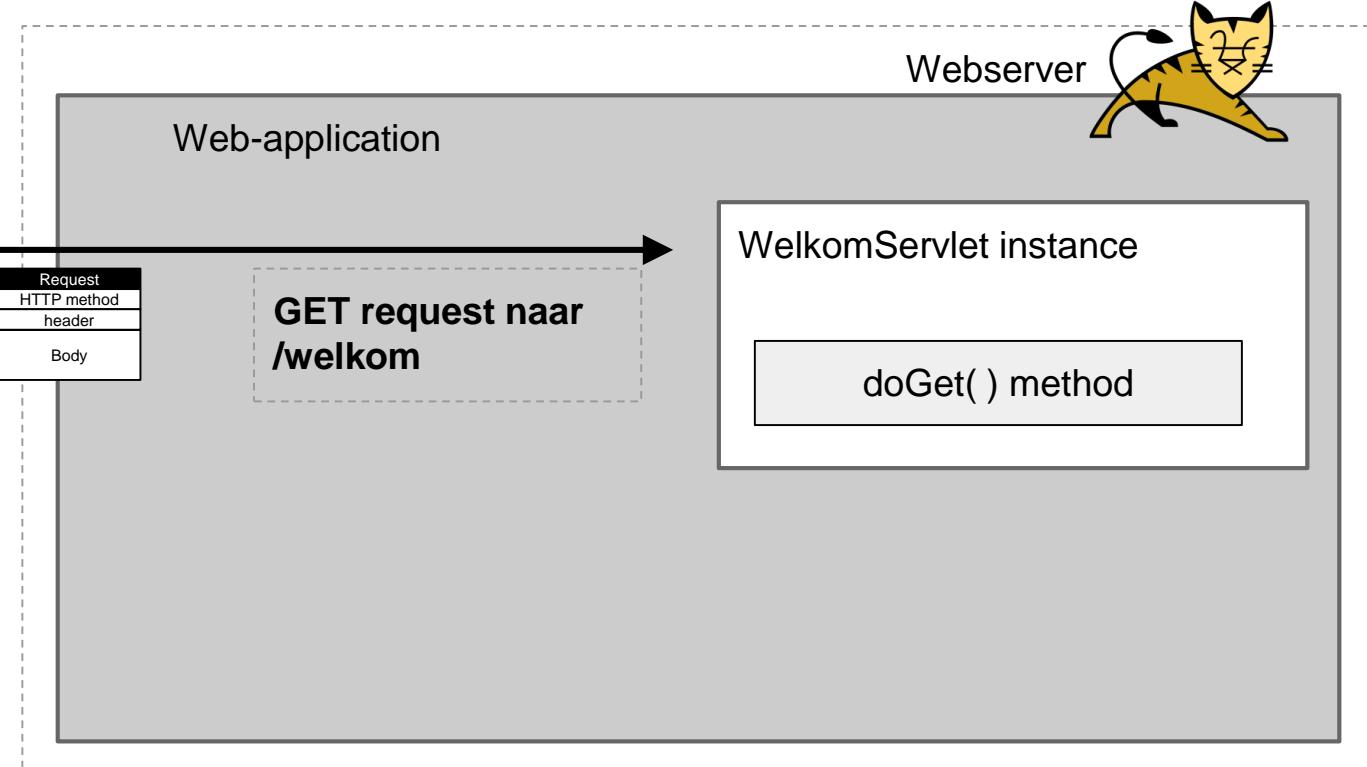
Servlets erven van de abstracte class HttpServlet:

Get request worden verwerkt door de method **doGet()**.

Post request worden verwerkt door de method **doPost()**.

GET request naar /welkom

browser



De class HttpServlet en zijn methods

Method	Omschrijving
doGet()	Vraagt de inhoud van een bepaalde resource op. Eventuele parameters worden toegevoegd aan de URL en zijn daardoor ook beperkt in lengte.
doPost()	Vraagt de inhoud van een bepaalde resource op. Eventuele parameters worden toegevoegd in de body van het verzoek.
doHead()	Vraagt enkel de header op van een bepaalde resource
doPut()	Voegt een bepaalde resource toe op de server
doDelete()	Wist een bepaalde resource op de server
doOptions()	Geeft een lijst van beschikbare methods voor een bepaalde resource
doTrace()	Stuurt gewoon het verzoek terug naar de client.

De parameters van doGet() en doPost()

HttpServletRequest

- Stelt **de browser request** voor
- Wordt gebruikt om de onderdelen van een **request te lezen**
 - De body van een POST request
 - De query string van een GET request
 - De request headers

HttpServletResponse

- Stelt **de response naar de browser** voor
- Wordt gebruikt om de onderdelen van een **response te maken**
 - De body van de response, die je normaal vult met HTML
 - Response headers

De response opbouwen

Stap 1

De browser vraagt de pagina welkom op van onze website. Als deze op het internet is gepubliceerd:

<http://www.pizzamamamia.be/welkom>

Of nu in de ontwikkelingsfase:

<http://localhost:8080/pizzamamamia/welkom>

Stap 2

Tomcat gaat op zoek naar de servlet met de annotation
@WebServlet("welkom") en voert de **method doGet()** uit.

Het is immers een gewone vraag naar html, een request zonder dat er data moet worden aangepast of zo...

Stap 3

In de method `doGet()` doen we eventuele berekeningen en bouwen tenslotte **de response** op onder de vorm van html code. Deze wordt dan door de server naar de browser gestuurd.

Om in onze method `doGet` html te kunnen schrijven hebben we een **PrintWriter object** nodig

```
PrintWriter out = resonse.getWriter( )
```

De response opbouwen

Hoe doe je dat nu?

1. Je vult optioneel de header op met de method **setHeader** (zie voorbeeld)
2. Je roept in de method **doGet** op de response parameter de method **getWriter** op. Je krijgt een object van het type **PrintWriter**.
3. Je roept op dit PrintWriter object de method **print** op. Je geeft de HTML die je naar de browser stuurt mee als parameter.

Opmerking:

Je kunt ook de method **println** oproepen. Deze method werkt zoals de method print maar stuurt een extra Enter teken naar de browser.

De code

```
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {

    // Header instellen:
    response.setContentType("text/html");
    response.setCharacterEncoding("UTF-8");

    // Webpagina: layout:
    PrintWriter out = response.getWriter();
    out.println("<!doctype html>");
    out.println("<html lang='nl'>");
    out.println("<head>");
    out.println("<title>Welkom bij Pizza Mama Mia</title>");
    out.println("</head>");
    out.println("<body>");
    out.println("Hallo Servlet");
    out.println("</body>");
    out.println("</html>");
    out.close();

}
```

De servlet instellen als welkompagina

web.xml

In **web.xml** kunnen we onze servlet als welkompagina instellen:



The screenshot shows a code editor window with the title bar "Welkom bij Pizza Mama Mia" and the tab "web.xml". The content of the editor is the XML configuration for a web application:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xmlns="http://xmlns.jcp.org/xml/ns/javaee"
4   xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
5   id="WebApp_ID" version="3.1">
6   <display-name>PizzaMamaMia</display-name>
7   <welcome-file-list>
8     <welcome-file>welkom</welcome-file>
9   </welcome-file-list>
10 </web-app>
```

Het programma uitvoeren

