**Project Fasten Your Seatbelts**

**Captive portal**

Klas: IN104

Groep: 4

Door: Solaiman Al Bacha en Errol Thielman



# Inhoud

Inhoudsopgave

[Inhoud 2](#_Toc405501039)

[Inleiding 3](#_Toc405501040)

[Tomcat 4](#_Toc405501041)

[Tutorial 4](#_Toc405501042)

[HTML en CSS 8](#_Toc405501043)

[HTML file aanmaken in Tomcat 8](#_Toc405501044)

[Uitleg Index Page: 12](#_Toc405501045)

[Java Servlets 16](#_Toc405501046)

[Servlet aanmaken Tomcat 16](#_Toc405501047)

[MySQL 20](#_Toc405501048)

[IP Tables 23](#_Toc405501049)

[MAC Address 24](#_Toc405501050)

[Exporting Captive portal from VMWare to Raspberry Pi 25](#_Toc405501051)

[Bijlage 26](#_Toc405501052)

[Index.html: 26](#_Toc405501053)

[Error.html: 28](#_Toc405501054)

[Welcome.html 29](#_Toc405501055)

[ToC.html 30](#_Toc405501056)

# Inleiding

Uiteraard willen we niet dat iedere gebruiker zich ongecontroleerd kan aanmelden op het wifi. Hiervoor hebben wij een Captive Portal ontwikkeld. Dit is een webpagina waarop de gebruiker zich moet aanmelden met een gebruikersnaam en wachtwoord. Naast het inloggen staan ook de algemene voorwaardes op de captive portal, dit zodat de gebruiker weet wat hij wel en niet kan doen op het netwerk, ook staat hier wat er bijvoorbeeld met zijn/haar data gebeurt.

Dit is onze handleiding voor de Captive Portal. We gaan in deze handleiding uitleggen hoe alles is gemaakt en hoe en wat sommige specifice code’s of servlets werken.

Wij hebben gebruik gemaakt van de volgende software/programma’s.

* Notepad ++
* Eclipse
* VMWare
* MySQL Server
* Besturingssystemen: Ubuntu en Windows 7 en 8.1

# Tomcat

Voor de captive portal heb je eerst een webserver nodig om alles op te hosten. Hiervoor gebruiken we het opensource programma “Tomcat.” Tomcat is een Java Servlet Container, omdat de captive portal gebruik maakt van Java Servlets, is Tomcat ideaal. Wij maken gebruik van Tomcat v7.0 op Ubuntu Server.

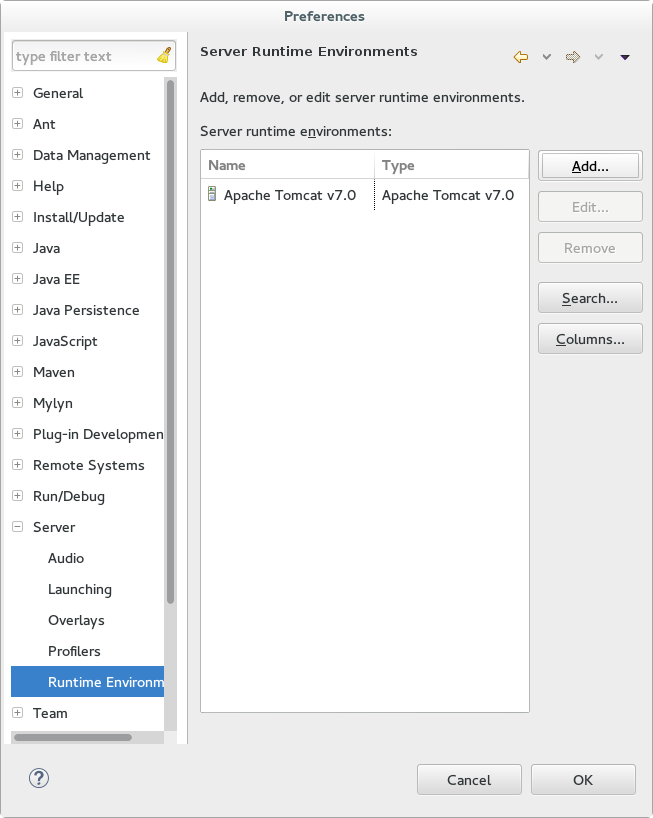
## Tutorial

Zorg er eerst voor dat je Eclipse [hier](https://www.eclipse.org/downloads/) downloadt, zorg er vervolgens ook voor dat je apache-Tomcat [hier](http://tomcat.apache.org/) downloadt. Deze doen we niet met apt-get zodat we geen compatibility issues krijgen met tomcat en eclipse.

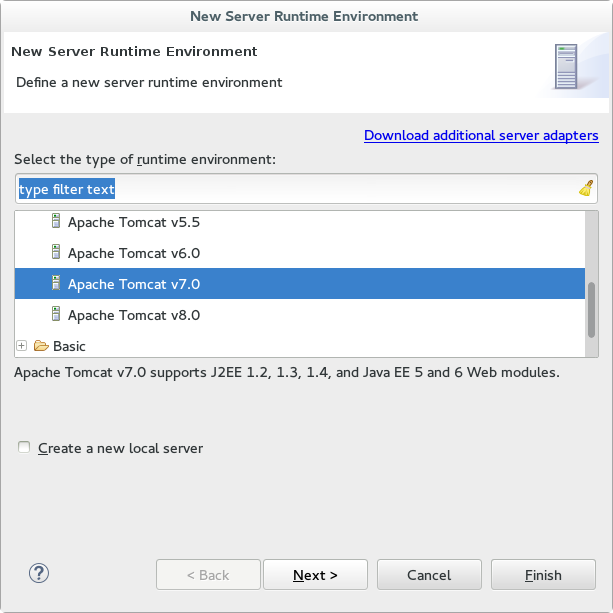
Nadat je dit hebt gedaan pak je Eclipse uit om er gebruik van te kunnen maken, nadat je dit hebt gedaan start je Eclipse voor de eerste keer op.

Nu is het de bedoeling om de server toe te voegen in Eclipse. De eerste stap is om via *windows>preferences>runtime environments* Tomcat toe te voegen.

Als je de schuingedrukte stappen hebt gevolgd kom je bij het volgende scherm (zie afbeelding)

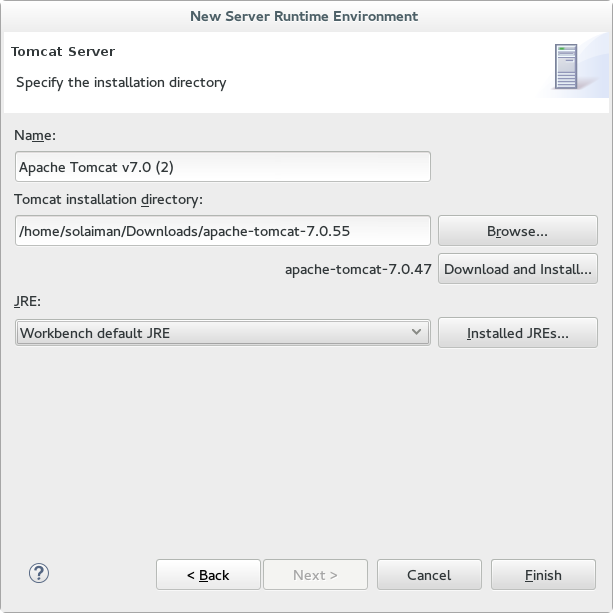


Afbeelding 1 – Toevoegen van een server in Eclipse.

Hier druk je vervolgens op **Add** en selecteer je Apache Tomcat v7.0

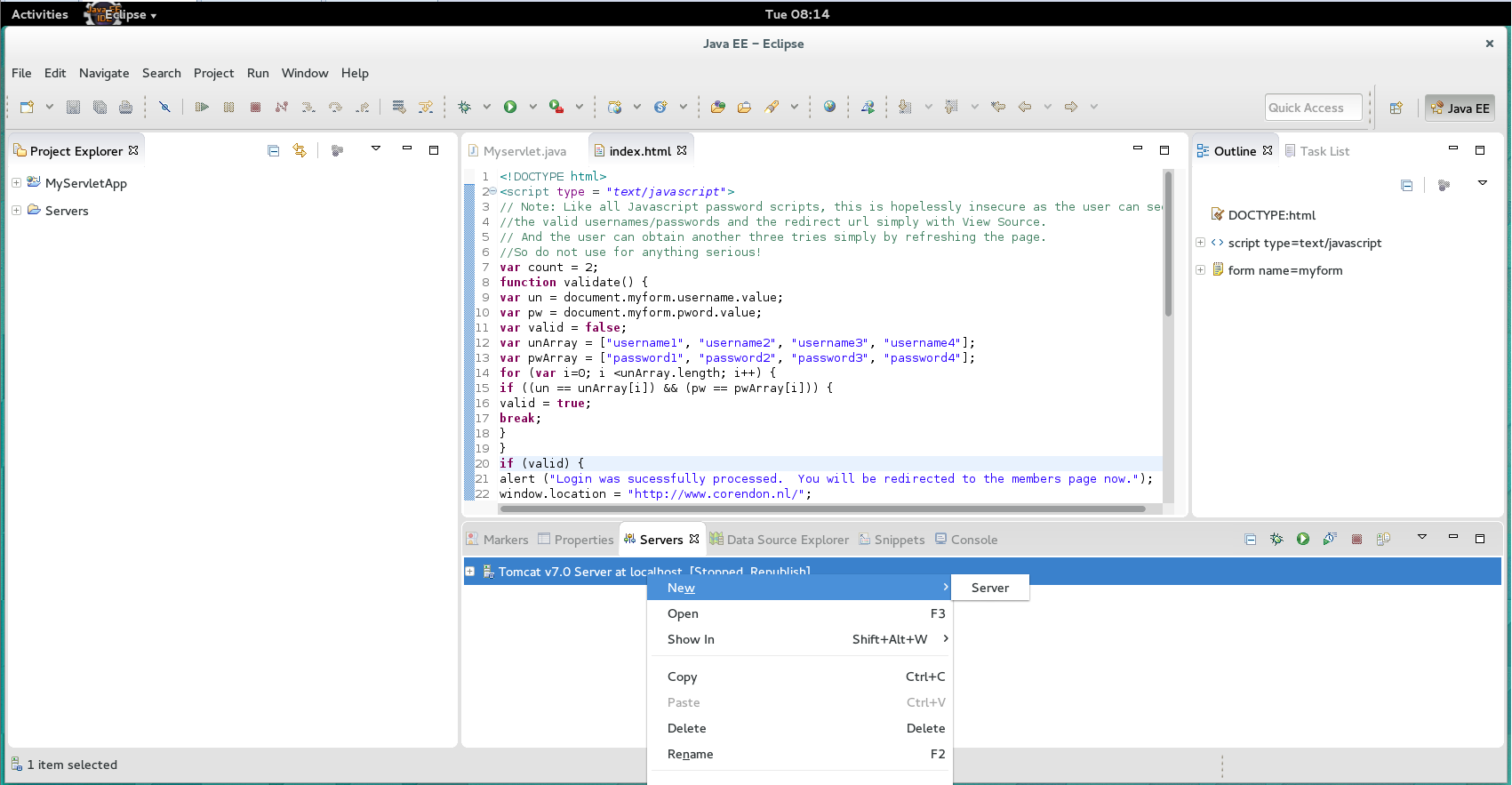
Afbeelding 2 – Selecteer Tomcat.

Nadat je dit hebt gedaan moet je Tomcat een “installation directory” toewijzen, dit is de locatie waar Tomcat zich bevindt, dit heb je bij de eerste stap gedownload. De overige instellingen laat je voor wat het is. Vervolgens druk je op finish.



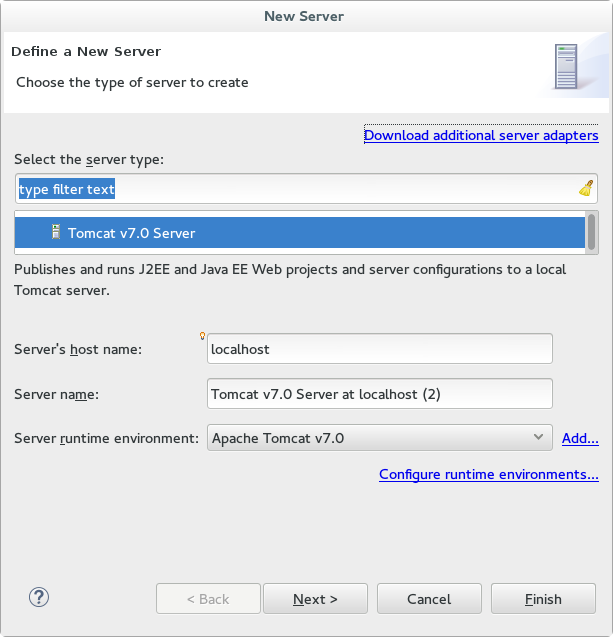
Afbeelding 3 – Afronden van de wizard.

Nu we de Tomcat hebben toegevoegd, kan je een server aanmaken. Dit doe je als volgt. Je doet rechtermuisklik op servers en drukt op new.



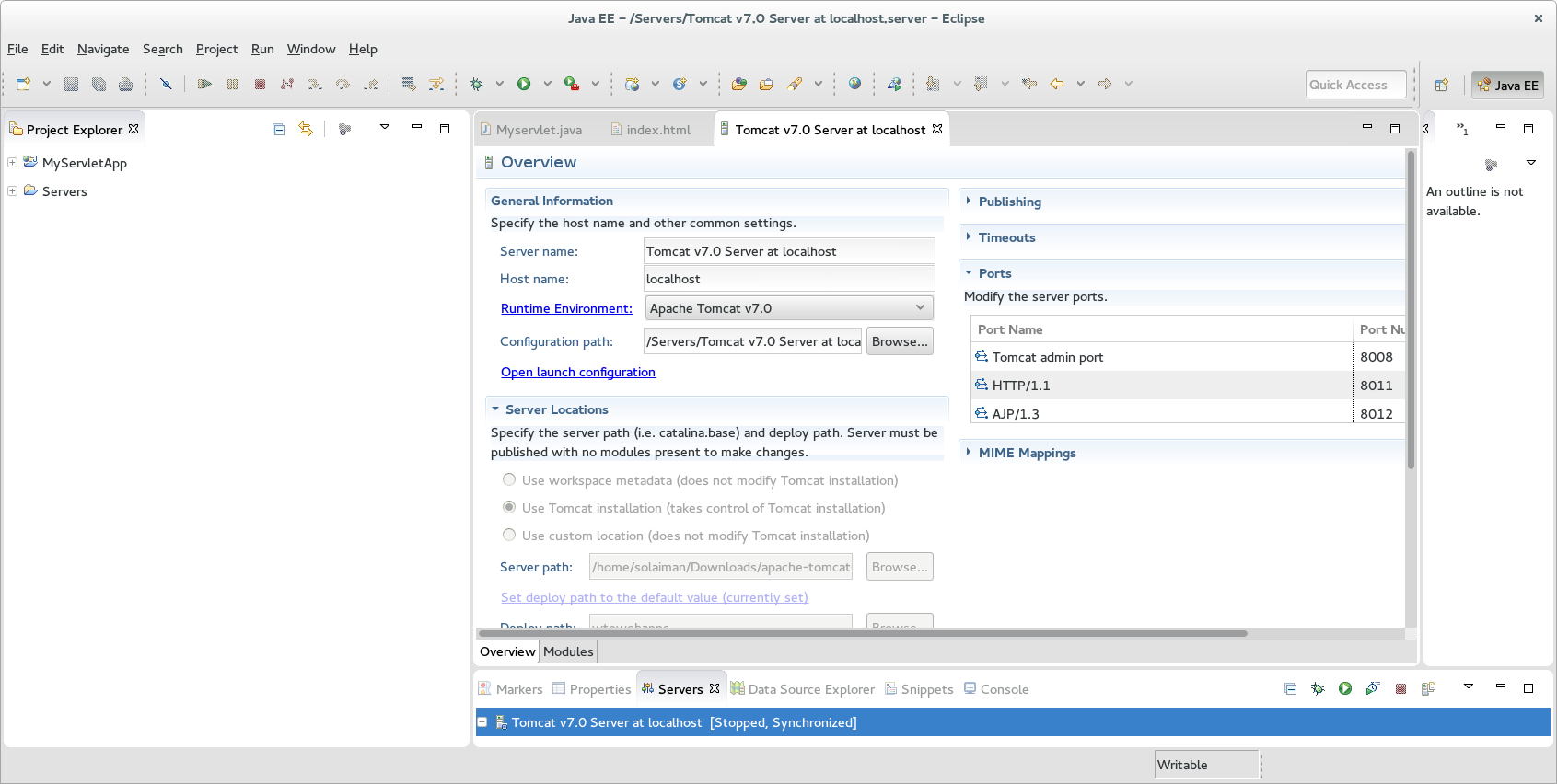
Afbeelding 4 – Het maken van een server

Nadat je dit hebt gedaan, komt het volgende scherm te voorschijn. Hier hoef je niks te veranderen, dus je drukt vervolgens op finish.



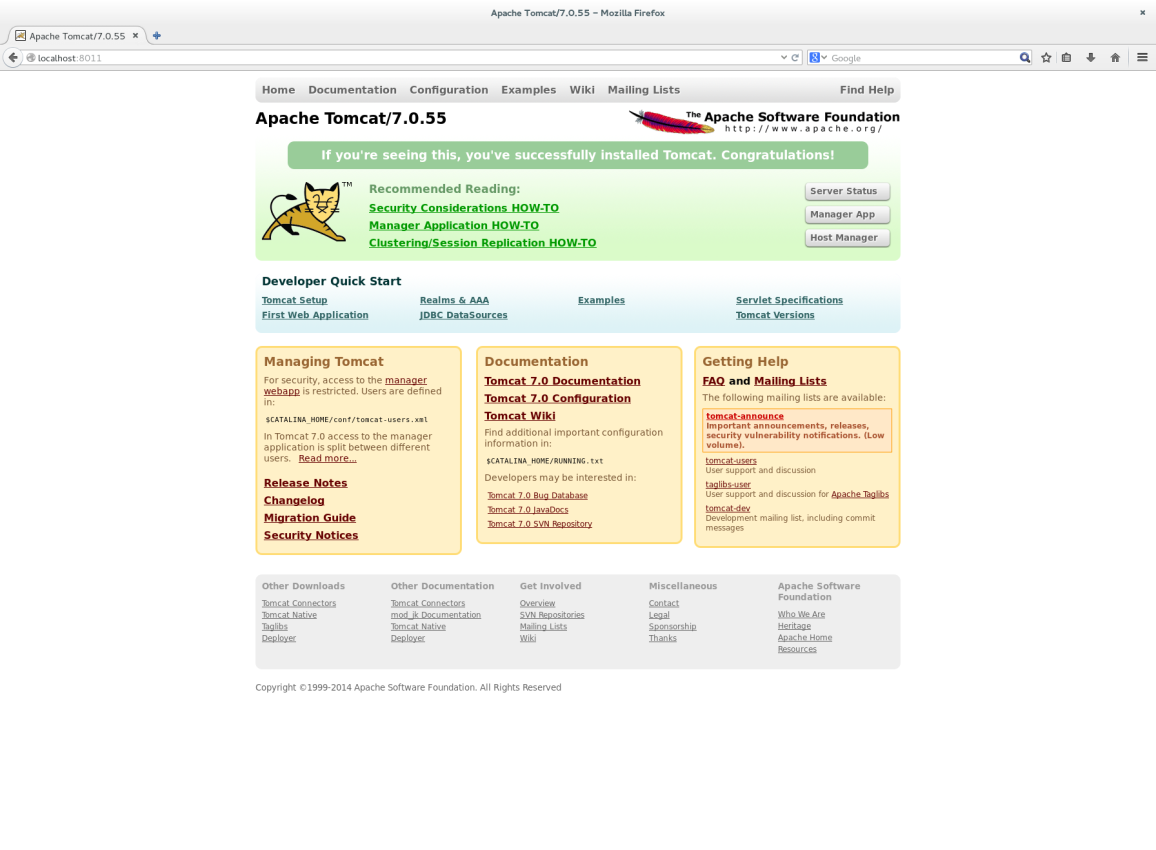
Afbeelding 5 - Afronden van de installatie van de server.

Nu staat, als het goed is, de server in Eclipse en kan je het starten en stoppen. Dubbelklik nu op de Tomcat server om de instellingen te kunnen aanpassen. Het scherm op de volgende pagina hoort nu tevoorschijn te komen



Afbeelding 6 – De instellingen van de Tomcat server.

Verander de “server locations” in “ Use Tomcat installation (takes control of Tomcat installation)”. Nadat je dit gedaan hebt kan de server opnieuw worden gestart. Test de werking door in je webbrowser te navigeren naar <http://localhost:8080>. In mijn geval is het 8011 omdat ik het poortnummer heb veranderd.

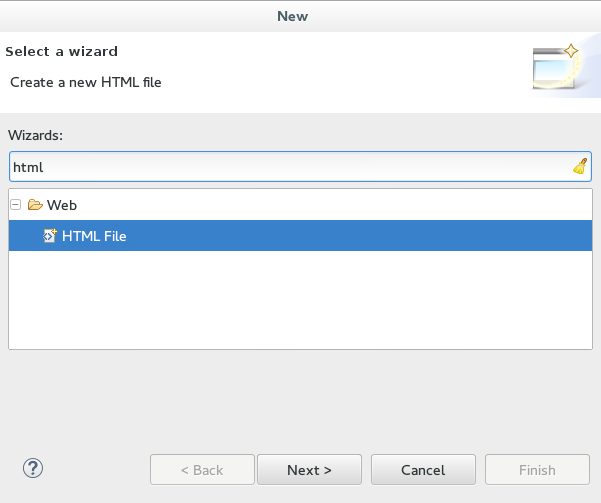
Als alles goed verloopt kom je op de pagina terecht Apache Tomcat.

Afbeelding 7 – De standaardpagina van Tomcat.

# HTML en CSS

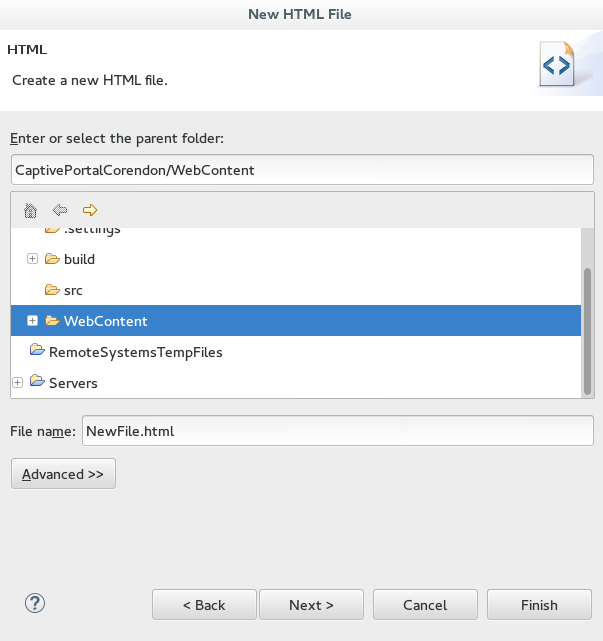
Wij maken gebruik van vier HTML pages en twee CSS stylesheets. HTML is een basis code taal die wordt gebruikt voor het maken van een website. Een CSS style sheet wordt gebruikt om de HTML-pagina een mooie opmaak te geven.

## HTML file aanmaken in Tomcat

1. Het aanmaken van een HTML pagina is vrijwel identiek aan het aanmaken van een servlet. Alleen zit het verschil in de locatie waar je het aanmaakt. Het aanmaken van een HTML pagina gebeurt in “webcontent”. Dit doe je op de volgende manier: new>html. Mits je html er niet tussen ziet doe je het op de volgende manier: new > other>html

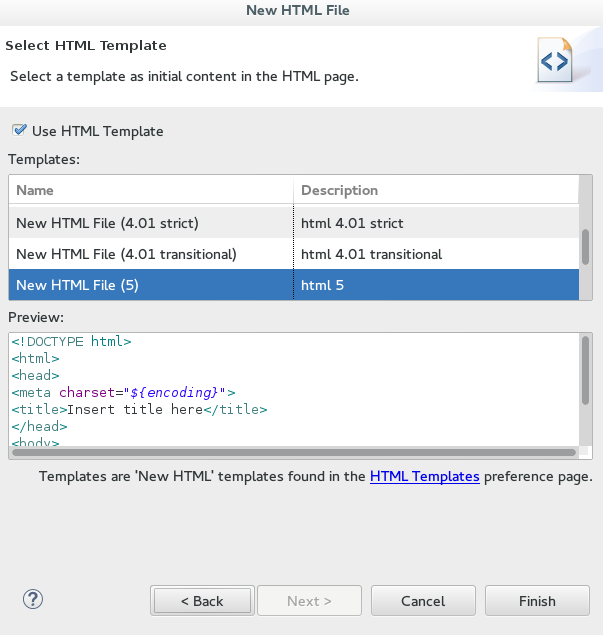
Afbeelding 8 - Het maken van een HTML bestand.

1. Nadat je op html hebt gedrukt, kom je op het volgende venster terecht. Hier moet je een naam en locatie kiezen van je html file. Kies als parent folder WebContent. Bij file name vervang je NewFile door een zelf gekozen naam. Druk op next.

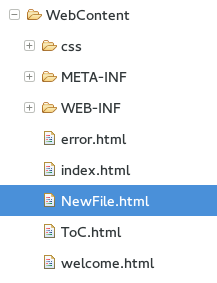


Afbeelding 9 – Het kiezen van de juiste opslagplek.

1. Nadat je een naam hebt gegeven aan je html file, kan je ervoor kiezen om een template te gebruiken. Standaard staat html5 ingesteld, laat dit zo staan en druk op finish.



Afbeelding 10 – De afronding.

1. Nu je op finish hebt gedrukt kan je je

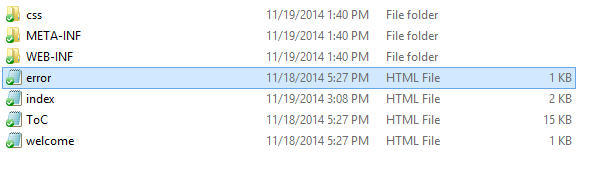
html file vinden onder het kopje

Webcontent, door er dubbel op te

klikken kan je de inhoud aanpassen.

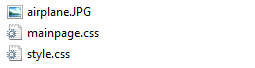
Afbeelding 11 – Het nieuwe html bestand is zichtbaar!

WebContent Folder:

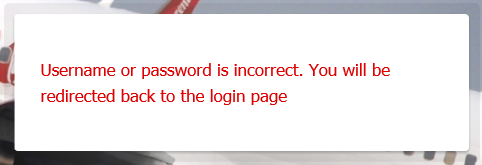


Afbeelding 12 - De inhoud van Webcontent.

Css: Folder met de CSS style sheets.

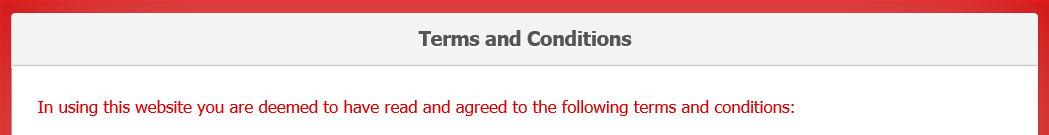


Error.html: Als de username en password niet goed zijn ingevuld geeft hij je de melding en stuurt je terug naar de index page.

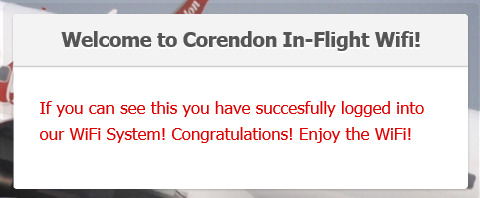


Index.html: Dit is het hoofdpagina van de Captive portal. Verdere uitleg op Pagina 12.

ToC.html: Dit is de page met de Terms and Conditions.



welcome.html: Deze page laat je weten dat je ingelogt bent, daarna verzend je naar [www.corendon.nl](http://www.corendon.nl).



Uitleg Index Page:  
  


Afbeelding 13 – De code van de index pagina.

Uitleg Code:

<link rel="stylesheet" href="css/style.css">

Dit zorgt er voor dat de HTML pagina verwijst naar de correcte style sheet, in deze pagina is het de style sheet “style.css” in het folder “css”. Zo weet het webbrowser waar alles moet worden gezet.

<SCRIPT language=JavaScript>

function checkCheckBox(f){

if (f.agree.checked == false )

{ alert('You have not agreed to the Terms and Conditions, Please try again.');

return false;}else

return true;

}

//-->

</SCRIPT>

Dit is een javascript code. Deze code is verbonden met de check box:

“<b>I agree with the <a href="ToC.html">Terms and Conditions</a> <input type="checkbox" value="0" name="agree"><p>class="submit"><input type="submit" value="Login"></p>”

Met de code “<SCRIPT language=JavaScript>” Geef je aan de html page aan dat er een javascript opdracht moet worden uitgevoerd. Bij deze code geeft het script aan dat als de “CheckBox” niet is aangevinkt er een Alert Message Pop-up Box komt die dan zegt: 'You have not agreed to the Terms and Conditions, Please try again.'. Wanneer de check box wel is aangevinkt laat het je gewoon doorgaan.

<form action="servlet1" method="post" onsubmit="return checkCheckBox(this)">

<section class="container">

<div class="login">

<h1>Sign in to make use of the Onboard WiFi</h1>

<form method="post" action="index.html">

<p><input type="text" name="username" value="" placeholder="Ticket Nummer"></p>

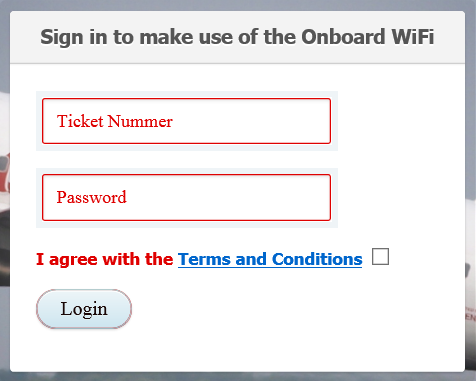
<p><input type="password" name="pass" value="" placeholder="Password"></p>

<label><b>I agree with the <a href="ToC.html">Terms and Conditions</a> <input type="checkbox" value="0" name="agree">

<p class="submit"><input type="submit" value="Login"></p>

</section>

Hierboven staat de code om het form gedeelte te maken:



In het eerste stukje zie je de volgende code:

<form action="servlet1" method="post" onsubmit="return checkCheckBox(this)">.

Het eerste wat er hier gebeurt, is dat het form wordt verbonden met de eerste JavaServlet. In ons geval is deze servlet onze “Log In” servlet. De functie van deze servlet wordt in de “JavaServlet” Sectie besproken. De informatie dat in de forms wordt getypt wordt dan verzonden naar de JavaServlet. Bij dit stukje code wordt ook gezegd dat wanneer het knopje submit wordt ingedrukt dat het eerst naar de Java Script gaat.

<p><input type="text" name="username" value="" placeholder="Ticket Nummer"></p>



Dit is de eerste form. Hier voer je de “Ticketnummer” in end html geeft het de “username” variable. De placeholder code is hier zodat in plaats dat de “Ticketnummer” buiten de form staat, staat het echt in het midden van de form. Dit is zo gedaan om het zo compact mogelijk te houden.

<p><input type="password" name="pass" value="" placeholder="Password"></p>



Dit is de tweede form. Deze is ongeveer hetzelfde als de eerste form. Het verschil is dat hier bij de input type “password” staat. Deze staat als password zodat wat er ingetypt wordt als sterretjes (\*) uitkomt. Andere mensen kunnen dan niet zien wat de password is.

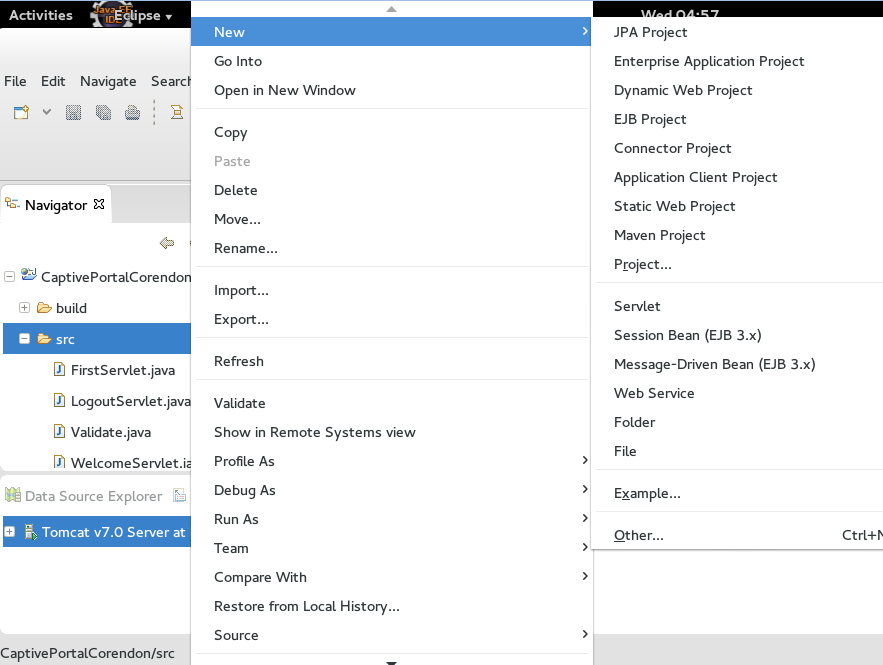
<label><b>I agree with the <a href="ToC.html">Terms and Conditions</a> <input type="checkbox" value="0" name="agree">



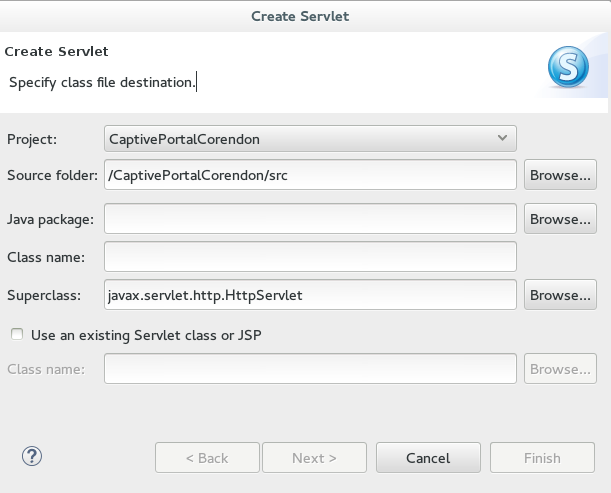
# Java Servlets

## Servlet aanmaken Tomcat

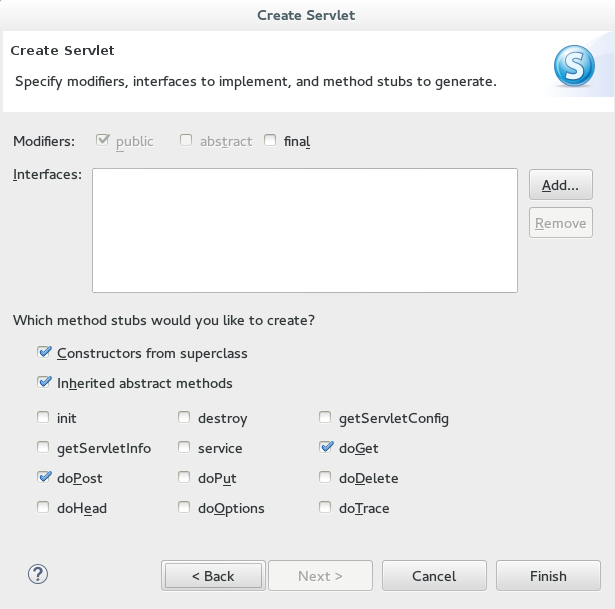
1. Een Java servlet maak je op de volgende manier:

Src > new > Servlet

Afbeelding 14 – Het maken van een nieuwe servlet.

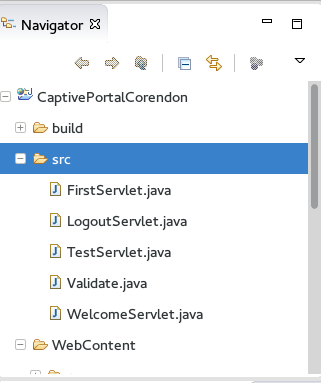
1. Nadat je dit hebt gedaan kom je op het onderstaande venster terecht. Hier kan je zien voor welk project je de servlet aanmaakt. Daarnaast kan je zien waar je de servlet aanmaakt, het kan nodig zijn dat je eerst een package moet aanmaken alvorens je de servlet aanmaakt. Bij Class name geeft je de naam aan van de Servlet. Druk nu op next.

Afbeelding 15 – Naamgeving van de servlet.

1. Druk next totdat je bij de laatste venster bent gekomen. Zoals je kan zien, kan je ervoor kiezen om methods toe te voegen aan je servlet, standaard is de doPost en doGet aangevinkt. Dit zijn ook de methods waar veel gebruik van gemaakt gaat worden. Druk nu Finish.

Afbeelding 16 - De afronding van de servlet.

1. Nadat je op finish hebt gedrukt, komt de servlet te staan in de src map. Het gaat hier om de TestServlet. Door er nu dubbel op te klikken kan je de inhoud ervan aanpassen.



Afbeelding 17 – De net gemaakte servlet in de src map!

1. Zoals je kan zien bestaat de inhoud uit DoPost en DoGet omdat je die hebt aangevinkt tijdens het aanmaken.

Afbeelding 18 – De inhoud van de servlet met DoPost en DoGet

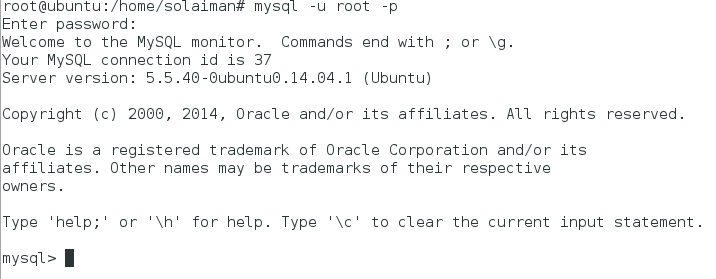
# MySQL

MYSQL database aanmaken en verbinden met de captive portal.

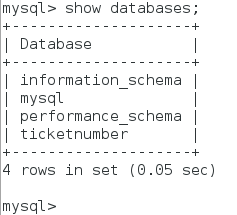
1. Allereerst heb ik mysql server geinstalleerd doormiddel van

sudo apt-get install mysql-server

1. Tijdens de installatie moet je een wachtwoord kiezen voor je mysql server zodat je daar vervolgens meekan inloggen. Nadat de installatie is voltooid kan je verder gaan met de volgende stap.
2. Nu de server is geinstalleerd kan je doormiddel van mysql -u root –p inloggen op de mysql server. Als alles goed verloopt krijg je het onderstaande te zien.



1. Doormiddel van show databases; kan je bekijken welke databases al aanwezig zijn op de server. Dat ziet er als volgt uit :



1. De eerste 3 databases zijn standaard al voorgeinstalleerd, zoals je kan zien is de laatste database door mij aangemaakt. Een database maak je doormiddel van: create database Verzinzelf; . In dit voorbeeld heb ik gebruik gemaakt van create database test;.



1. Nu de database is aangemaakt doe je weer show databases; Nu staat als het goed is de database erbij en kan je gebruik maken van de database doormiddel van use test;

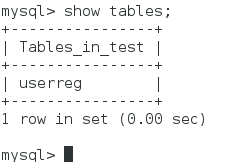


1. Nu je gebruik maakt van de database kan je tables gaan aanmaken in de database. Doormiddel van show tables kan je zien of er al tables aanwezig zijn in de db. Aangezien dit niet het geval is maak je doormiddel van CREATE TABLE een table aan. Als voorbeeld gebruik ik de waarde die in de table voorkomen die ik heb aangemaakt

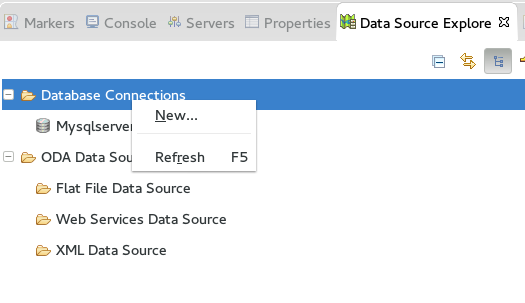
create table userreg(name varchar2(40),pass varchar2(40));



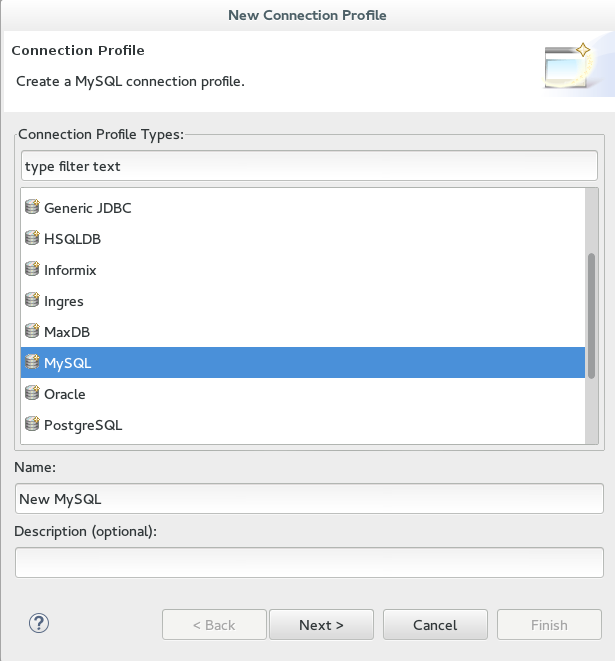
1. Nu doe je nogmaals show tables en zie je dat de table is aangemaakt.



1. Nu je de database en de table hebt aangemaakt ga je in eclipse naar de datasource explorer. Hier kies je er vervolgens voor om een nieuwe connectie aan te maken.



1. Nadat je op new hebt gedrukt kies je voor MYSQL en kies je een passende naam. Zoals je kan zien heb ik die van mij MYSQLserver genoemd.



# IP Tables

# MAC Address

# Exporting Captive portal from VMWare to Raspberry Pi

# Bijlage

## Index.html:

<!DOCTYPE html>

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">

<title>Login Form</title>

<link rel="stylesheet" href="css/style.css">

</head>

<body>

<SCRIPT language=JavaScript>

function checkCheckBox(f){

if (f.agree.checked == false )

{ alert('You have not agreed to the Terms and Conditions, Please try again.');

return false;}else

return true;

}

//-->

</SCRIPT>

<form action="servlet1" method="post" onsubmit="return checkCheckBox(this)">

<section class="container">

<div class="login">

<h1>Sign in to make use of the Onboard WiFi</h1>

<form method="post" action="index.html">

<p><input type="text" name="username" value="" placeholder="Ticket Nummer"></p>

<p><input type="password" name="pass" value="" placeholder="Password"></p>

<p class="remember\_me">

<label>

<b>I agree with the <a href="ToC.html">Terms and Conditions</a> <input type="checkbox" value="0" name="agree">

<p class="submit"><input type="submit" value="Login"></p>

</section>

</body>

</html>

## Error.html:

<!DOCTYPE html>

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">

<title>Terms and Conditions</title>

<link rel="stylesheet" href="css/style.css">

</head>

<body>

<section class="container">

<div class="login">

<p>Username or password is incorrect. You will be redirected back to the login page</p>

<meta http-equiv="refresh" content="3;url=index.html" />

</section>

</body>

</html>

## Welcome.html

<!DOCTYPE html>

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">

<title>Terms and Conditions</title>

<link rel="stylesheet" href="css/style.css">

</head>

<body>

<section class="container">

<div class="login">

<h1>Welcome to Corendon In-Flight Wifi!</h1>

<p>If you can see this you have succesfully logged into our WiFi System! Congratulations!

Enjoy the WiFi!</p>

<meta http-equiv="refresh" content="5;url=http://www.corendon.nl" />

</section>

</body>

</html>

## ToC.html

<!DOCTYPE html>

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">

<title>Terms and Conditions</title>

<link rel="stylesheet" href="css/mainpage.css">

</head>

<body>

<section class="container">

<div class="login">

<h1>Terms and Conditions</h1>

<p>In using this website you are deemed to have read and agreed to the following terms and conditions:

<p>The following terminology applies to these Terms and Conditions, Privacy Statement and Disclaimer Notice and any or all Agreements: "Client", “You” and “Your” refers to you, the person accessing this website and accepting the Company’s terms and conditions. "The Company", “Ourselves”, “We” and "Us", refers to our Company. “Party”, “Parties”, or “Us”, refers to both the Client and ourselves, or either the Client or ourselves. All terms refer to the offer, acceptance and consideration of payment necessary to undertake the process of our assistance to the Client in the most appropriate manner, whether by formal meetings of a fixed duration, or any other means, for the express purpose of meeting the Client’s needs in respect of provision of the Company’s stated services/products, in accordance with and subject to, prevailing English Law. Any use of the above terminology or other words in the singular, plural, capitalisation and/or he/she or they, are taken as interchangeable and therefore as referring to same.</p>

--etc

<FORM><INPUT Type="button" VALUE="Back" onClick="history.go(-1);return true;"></FORM>

</section>

</body>

</html>