

Fhir On Pi - 4



Tomcat opstarten bij boot

In de [vorige aflevering](#) hebben we de Raspberry Pi voorzien van Tomcat, de HTTP-webserver.

Nadat we dat gedaan hadden konden we met de opdracht:

```
sudo systemctl status tomcat
```

zien dat het tomcat proces gestart was en draaide.

Wanneer je daarna je Pi hebt uitgezet of opnieuw hebt gestart ben je misschien geschrokken omdat tomcat het opeens niet meer deed. Geen paniek, als je opnieuw de opdracht:

```
sudo systemctl start tomcat
```

geeft, zal tomcat weer starten. Je kunt dat zien door vanaf je PC met de browser naar het IP-adres van je Raspberry Pi te gaan en daar poort 8080 te kiezen. Dus door -bijvoorbeeld- in je browser het adres: <http://192.168.1.25:8080> te typen. (waarbij je 192.168.1.25 vervangt door het IP-adres van je eigen Pi).

Computers zijn goed in herhaling. Het lukt nog niet om computers echt iets origineels te laten bedenken, computers zijn -in de kern- niet creatief¹. Maar in herhalende activiteiten zijn computers beter dan mensen. We willen dus niet zélf steeds tomcat opnieuw opstarten, maar laten dat de Pi doen. Uiteraard gebruiken we daarvoor de systemctl tool weer:

```
sudo systemctl enable tomcat
```

Nu start tomcat op tegelijk met de rest van de Pi. Probeer maar:

```
sudo reboot now
```

even wachten en dan opnieuw een Putty-sessie opzetten (sudo `systemctl status tomcat`) of je browser refreshen naar de 8080 poort. De computer heeft in ieder geval dit herhalende werk nu van je overgenomen.

¹ “Kunstuitingen” die computers maken zijn voor zover ik weet altijd resultaten van “algoritmes” waarmee *bestaande* kunst (muziek, grafiek, tekst) wordt verwerkt en opnieuw samengesteld.

Tomcat beheerportaal

Nu we toch bezig zijn met tomcat gaan we er ook het beheerportaal (management-portal) voor installeren. De Fhir-server kan ook zónder dat, maar het beheerportaal geeft wat meer inzicht in de opbouw van het geheel en zeker als je nog wat verder wilt dan alleen een Fhir-server laten draaien is het portaal handig. En het is ook niet veel werk.

Tomcat is “out-of-the-box” al redelijk beschermd. We gebruiken voor dit projectje geen HTTPS met certificaten en zo (iets wat zeker wél zou moeten in bijvoorbeeld een ziekenhuis-omgeving) maar ondanks dat is het niet makkelijk mogelijk om van buitenaf de Tomcat-configuratie aan te passen (tenzij je SSH-toegang hebt en root-privilege natuurlijk). Dit betekent óók dat alleen het *installeren* van het Tomcat beheerportaal niet voldoende is, we moeten:

1. Zorgen dat tomcat weet dat er een gebruiker is die via het beheerportaal de configuratie mag wijzigen
2. Zorgen dat de toegang tot het Tomcat-beheerportaal niet alleen vanaf de Raspberry Pi zelf, maar ook van buitenaf (bijvoorbeeld vanaf je PC) is toegestaan.

We beginnen met het aanpassen van het bestand **tomcat-users.xml** . Als je Tomcat hebt geïnstalleerd zoals ik in de vorige aflevering aangaf, dan doe je dat met:

```
sudo nano /opt/tomcat/conf/tomcat-users.xml
```

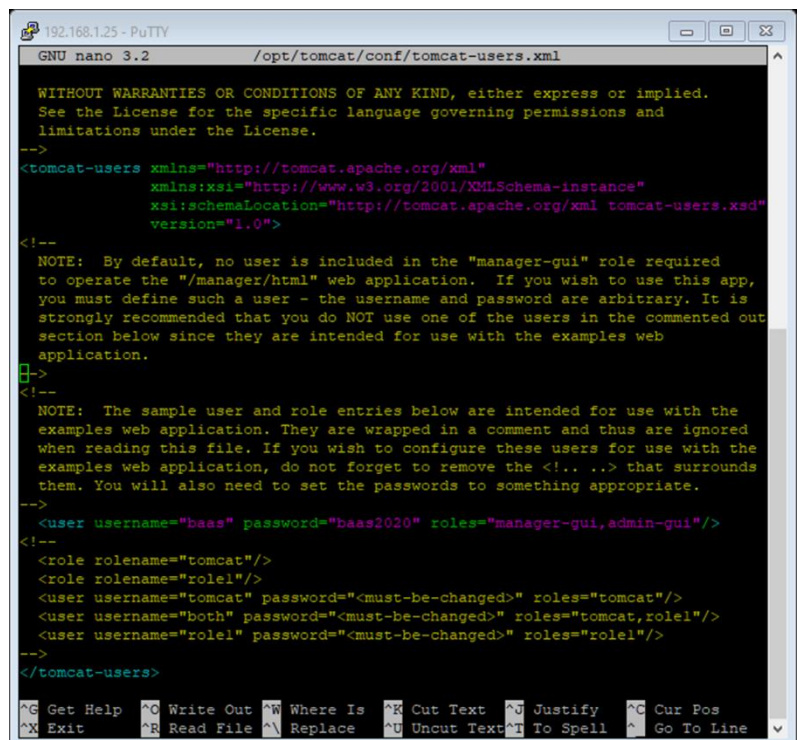
Je ziet dan allerlei blokken tekst, waarbij de meeste tekst tussen `<!--` en `-->` staat (die tekst is in mijn Putty-sessie automatisch geel). Deze tekst is commentaar, de Pi doet er niks mee, maar het vertelt je wel iets over de inhoud van het bestand.

Je ziet ook tekst tussen `<` en `>` tekens (bij mij in blauw), dit zijn “XML-tags”. Tussen de “tags” bevindt zich de informatie die bij de “tag” hoort.

Je ziet boven in het bestand in ieder geval de begin-tag `<tomcat-users` (met daarachter een hoop xmns informatie) `>` en onderin het bestand de eind-tag `</tomcat-users>`.

Zoek een mooi plekje tussen deze twee tags en *buiten* de commentaar-blokken en voeg daar de volgende regel toe:

```
<user username="admin" password="password" roles="manager-gui,admin-gui"/>
```



```
192.168.1.25 - PuTTY
GNU nano 3.2 /opt/tomcat/conf/tomcat-users.xml

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
-->
<tomcat-users xmlns="http://tomcat.apache.org/xml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
  version="1.0">
  <!--
  NOTE: By default, no user is included in the "manager-gui" role required
  to operate the "/manager/html" web application.  If you wish to use this app,
  you must define such a user - the username and password are arbitrary.  It is
  strongly recommended that you do NOT use one of the users in the commented out
  section below since they are intended for use with the examples web
  application.
  -->
  <!--
  NOTE: The sample user and role entries below are intended for use with the
  examples web application.  They are wrapped in a comment and thus are ignored
  when reading this file.  If you wish to configure these users for use with the
  examples web application, do not forget to remove the <!-- ... --> that surrounds
  them.  You will also need to set the passwords to something appropriate.
  -->
  <user username="baas" password="baas2020" roles="manager-gui,admin-gui"/>
  <!--
  <role rolename="tomcat"/>
  <role rolename="role1"/>
  <user username="tomcat" password="<must-be-changed>" roles="tomcat"/>
  <user username="both" password="<must-be-changed>" roles="tomcat,role1"/>
  <user username="role1" password="<must-be-changed>" roles="role1"/>
  -->
</tomcat-users>

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^U Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text ^H To Spell  ^_ Go To Line
```

Waarbij je de username (hier "admin") en het wachtwoord (hier: "password") door iets van jezelf mag vervangen. Schrijf wel even op wat je hier hebt ingevuld zodat je het makkelijk terugvindt. Een voorbeeld zie je in het plaatje (username: "baas" en password "baas2020").

Sla het bestand op met <ctrl> O en verlaat nano met <ctrl> X.

Dat was de eerste stap, je kunt nu inloggen op het beheerportaal, maar op dit moment alleen nog op de Pi zelf en die heeft geen echte webbrowser want daar draait Raspbian Lite. We zijn er dus nog niet!

Tot nu toe had ik het over het Tomcat beheerportaal, maar -**verrassing**- eigenlijk heeft Tomcat twee beheerportalen:

- De tomcat manager app
- De tomcat host manager app

en we moeten ze allebei instellen voor benadering van buitenaf. (Feitelijk bieden de beheerportalen zelfs drie beheerschermen, maar daar kom ik straks nog op terug...)

Eerst de **tomcat manager app**:

```
sudo nano /opt/tomcat/webapps/manager/META-INF/context.xml
```

Hier zie je de begin-tag: <Context antiResourceLocking="false" privileged="true" > en de eind-tag: </Context> .

Daartussen staat de restrictie op de benadering van het portaal. Zet daar <!-- voor en zet aan het einde --> (vóór het begin een "kleiner dan"-teken, uitroepteken, twee mintekens,

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements.  See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License.  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
-->
<Context antiResourceLocking="false" privileged="true" >
  <!-- <CookieProcessor className="org.apache.tomcat.util.http.Rfc6265CookiePro$
        sameSiteCookies="strict" />
  <Valve className="org.apache.catalina.valves.RemoteAddrValve"
        allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" />
  <Manager sessionAttributeValueClassNameFilter="java\.lang\.(?:Boolean|Integer$
  -->
</Context>
```

ná het einde twee mintekens en een “groter dan”-teken. Dit alles dus tussen de <Context> begin- en eindtag. Als het goed is en je gebruikt nano, dan is het hele blok tussen de <Context... begintag en de </Context> eindtag nu geel (want het is commentaar geworden).

Sla het bestand op met <ctrl> O en verlaat nano met <ctrl>X.

Dan de **host manager app**:

```
sudo nano /opt/tomcat/webapps/host-manager/META-INF/context.xml
```

Hier doe je precies hetzelfde. Dus een <!-- net na de Context-starttag en een

--> net voor de </Context> eindtag. <ctrl>O voor opslaan en <ctrl>X voor afsluiten nano.

Nu nog één stapje en we kunnen het beheerportaal gaan bekijken, we starten tomcat opnieuw:

```
sudo systemctl restart tomcat
```

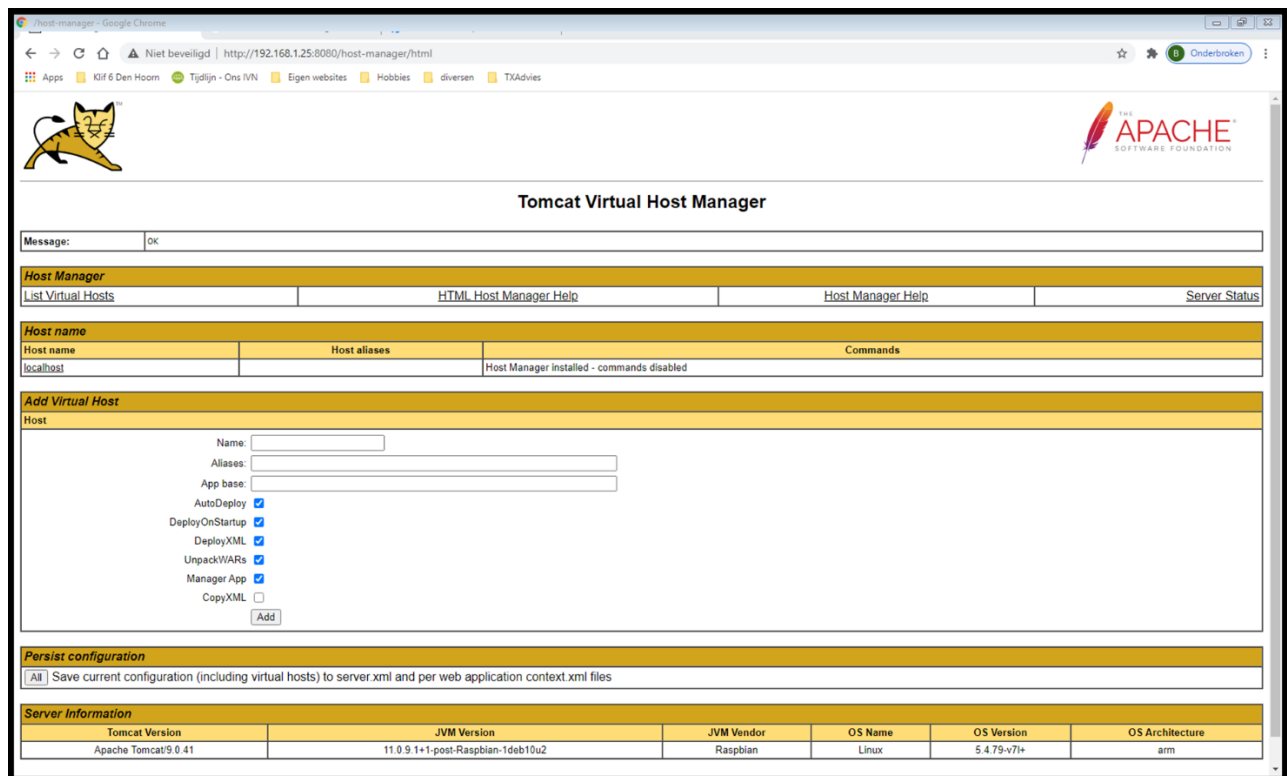
en gaan nu naar het managementportaal toe. Dat doe je door áchter het webadres van je tomcat-installatie het volgende toe te voegen: /manager/html . Op mijn configuratie is het adres dus: <http://192.168.1.25:8080/manager/html> Je zult moeten inloggen met de gebruikersnaam en het wachtwoord dat je in de eerste stap in tomcat-users.xml hebt ingevuld maar dan zie je dan ook het volgende scherm:

The screenshot shows the Tomcat Web Application Manager interface. At the top, there's a message bar and navigation links: Manager, List Applications, HTML Manager Help, Manager Help, and Server Status. Below this is a table of applications:

Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle > 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle > 30 minutes
/examples	None specified	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle > 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle > 30 minutes
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle > 30 minutes

Below the table is a 'Deploy' section with a form to deploy a WAR file or directory. The form includes fields for Context Path, Version (for parallel deployment), XML Configuration file path, and WAR or Directory path, followed by a Deploy button.

Straks leg ik kort uit wat het allemaal betekent, maar kijk eerst nog even naar het host manager scherm. Dat tref je aan als je /host-manager/html achter de :8080 van je Pi's IP-adres zet (in mijn configuratie : <http://192.168.1.25:8080/host-manager/html>)



Omdat je nu zelfs al gefeliciteerd wordt als je via Internet iets van twee euro besteld hebt ga ik dat niet doen, maar je hebt wél met succes. Tomcat én de Tomcat-beheerportalen op je Pi geïnstalleerd en daarmee hebben we (bijna) alle voorbereidingen gedaan om ook de FHIR-server te implementeren.

Misschien **vraag je je af** waarom we Tomcat op zo'n ambachtelijke manier hebben geïnstalleerd terwijl ik het in aflevering 2 uitgebreid over APT heb gehad, de package manager. En daarmee hebben we toch ook Java met succes op de Pi gezet. De reden is dat ik heb gemerkt ("ervaren" zelfs) dat APT, Tomcat verspreid neerzet in allerlei directories. Tomcat is heel slim, en als hij een bestand (bijvoorbeeld een configuratiebestand) niet op de ene plek vindt, dan zoekt hij verder op de andere plek. Maar door de installatie met APT gebeurde het me voortdurend dat ik ergens een configuratiefile aanpasse voor Tomcat (bijvoorbeeld tomcat-users.xml) maar dat de wijziging niet door Tomcat werd opgepikt. Toen ik na herhaalde problemen van deze aard één keer Tomcat installeerde op de manier die ik nu hier heb beschreven waren al die problemen verdwenen en was de installatie eigenlijk heel eenvoudig. Vandaar.

Tomcat beheer. Ik zou nog iets meer vertellen over het Tomcat beheerportaal (het scherm Tomcat Web Application Manager). Wat Tomcat een application noemt is een set aan bij elkaar behorende functionaliteit (dus inderdaad een "applicatie"/"toepassing"). Het beheerportaal maakt het mogelijk om die applicaties los van elkaar te starten , te stoppen, in productie te nemen en te verwijderen.

Van buitenaf heeft iedere applicatie een eigen webadres, herkenbaar aan het deel dat achter de :8080 van het ip-adres (of de domeinnaam) komt. Het beheerportaal is dus een tomcat-applicatie op /manager/html, er is een "examples"-applicatie op /examples en een

Tomcat-documentatie-applicatie op /docs. Bekijk die allemaal trouwens gerust, er zit nú al een enorme berg informatie op je Pi!

In het beheerscherf kun je zo'n applicatie dus even uitzetten. Doe bijvoorbeeld het volgende eens:

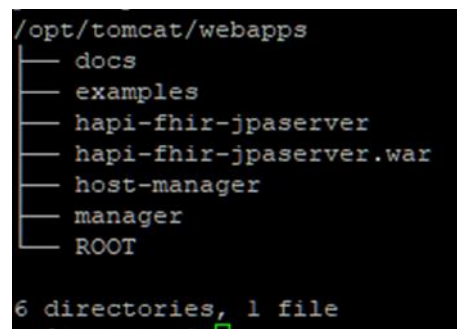
1. Ga naar <het-IP-adres-van-je-Raspberry>:8080/docs en blader even rond.
2. Ga naar het beheerscherf: <het-IP-adres-van-je-Raspberry>:8080/manager/html
3. Druk op de "Stop" knop bij de /docs applicatie. Je krijgt een melding in je beheerscherf (bij Message:)
4. Ga nu weer naar <het-IP-adres-van-je-Raspberry>:8080/docs en doe een "refresh" van je browser. Het duurt een paar seconden, maar dan krijg je een 404-fout te zien: Not Found. Je hebt namelijk de /docs applicatie offline gezet.
5. Ga nu terug naar je beheerscherf, start de /docs applicatie en alles werkt weer als vanouds.

Je snapt nu ook waarom Tomcat standaard zó wordt ingericht dat het beheer van buitenaf niet open staat!

Van binnenuit heeft iedere Tomcat-applicatie een eigen directory onder de /opt/tomcat/webapps directory (zie afbeelding). Dus docs, examples, host-manager en manager kom je daar allemaal tegen.

In de afbeelding hiernaast² zie je ook al de directory van de hapi-fhir-jpaserver staan, die we in de komende twee afleveringen ook op jouw Raspberry Pi gaan installeren.

Tot de volgende!



```
/opt/tomcat/webapps
├── docs
├── examples
├── hapi-fhir-jpaserver
├── hapi-fhir-jpaserver.war
├── host-manager
├── manager
└── ROOT

6 directories, 1 file
```

² Dat plaatje is trouwens de uitvoer van "tree", een handig Unix programmaatje om directory-structuren weer te geven. Het staat niet standaard in de Raspbian omgeving maar je hebt het zó geïnstalleerd met "**sudo apt-get install tree**". Het commando is (hoe kan het anders) "tree" en met de switch -L kun je aangeven hoeveel levels diep je wilt gaan. Het plaatje krijg je met **sudo tree -L 1 /opt/tomcat/webapps**.