Maarten Lindt en Niels Borkes

onderzoeksverslag wicket

Titel : onderzoeksverslag wicket

Auteur : Maarten Lindt 616195 en Niels Borkes 616825

Course: DEA, leerjaar 2019/2020

Docent(en): Meron Brouwer

Plaats: Nijmegen

School: Hogeschool van Arnhem en Nijmegen

Datum: ? maart 2020

|  |  |  |  |
| --- | --- | --- | --- |
| **WIJZIGINGSHISTORIE** | | | |
| *Datum* | *Versie* | *Beschrijving* | *Auteur* |
| 30 maart 2020 | 0.1 | Initiële versie | LDTM |
| 30 maart 2020 | 0.2 | Inleiding geschreven | LDTM |

Contents

[1. Voorwoord 1](#_Toc36454323)

[2. Inleiding 2](#_Toc36454324)

[3. Wat is wicket? 3](#_Toc36454325)

[4. Welke voordelen bied wicket in de Spotitube applicatie? 4](#_Toc36454326)

[5. Hoe werken de componenten van wicket? 5](#_Toc36454327)

[6. Conclusie 6](#_Toc36454328)

[7. Bijlagen 7](#_Toc36454329)

[8. Bronnen 8](#_Toc36454330)

# Voorwoord

# Inleiding

Onze namen zijn Niels Borkes en Maarten Lindt, wij zijn beide studenten aan de HAN. Wij kregen de opdracht om onderzoek te doen naar hoe we Spotitube kunnen verrijken door iets toe te voegen of aan te passen.

Wij hebben ervoor gekozen om in dit verslag onderzoek te doen naar de vraag: Is de aangeleverde client applicatie van Spotitube te vervangen door een wicket applicatie?

Dit gaan we doen door de volgende deelvragen te beantwoorden:

* Wat is Wicket?
* Welke voordelen bied het gebruik van wicket in de Spotitube applicatie?
* Hoe werken de componenten van wicket?

De eerste 2 vragen zullen onderzoeken door online onderzoek. Voor de laatste deelvraag zullen we echter een klein prototype bouwen zodat we voor onszelf een gevoel kunnen krijgen hoe het werkt.

# Wat is wicket?

Apache Wicket is een web application framework voor Java dat zich richt op het versimpelen van het bouwen van web interfaces. Met Wicket is het mogelijk om een web aplicatie te bouwen door alleen gebruik te maken van Java en XHTML, er hoeft dus geen Javascript of XML gebruikt te worden.

## Het design pattern in Wicket

Het framework werkt met het model-view-controller (MVC) pattern. In elke request cycle wordt de het inkomende request gemapped naar een methode op een controller object, die hierop een response genereert, de controller haalt de data hiervoor uit een model om daarmee de view te kunnen aanpassen.

## Versie van het Framework

Het framework is origineel geschreven door Jonathan Locke in april 2004 en de eerste versie kwam in Juni 2005 uit. In de laatste release is er een overstap gemaakt van Java 8 naar Java 11 als minimaal benodigde versie van Java en ondersteund nu ook 12&13. Ook zijn ze over gegaan van jUnit 4 naar 5. Het gebruikmaken van de nieuwste release heeft voor ons voordeel want zo kunnen wij onze geleerde methodes toepassen op wicket.

Wikipedia contributors. (2019, 12 september). *Apache Wicket*. Geraadpleegd op 30 maart 2020, van https://en.wikipedia.org/wiki/Apache\_Wicket

# Welke voordelen bied wicket in de Spotitube applicatie?

In dit hoofdstuk zullen we kijken welke voordelen wicket bied.

## Voordelen van wicket

* De gehele configuratie is in java. Dus niet xml, json o.i.d.
* Informatie staat op de server. Dit is handig voor onderhoud, lagere download tijden.
* Geeft je gemakkelijk een AJAX object om de ui te updaten.
* Je hebt Models.

Models zijn een simpel contract voor het opvragen en het zetten van een waarde. De UI heeft dus Models om makkelijk data op te vragen en de Models zoeken uit hoe ze aan de data moeten komen. En waar ze eventueel data moeten opslaan bij bijvoorbeeld een invoerveld.

De voordelen van wicket Models is dat de Models pas aangeroepen worden als het echt nodig is. Dit voorkomt onnodig inladen van data.

Het werkt namelijk als volgt: de Models worden gemaakt op de server. Voordat de client de HTML opvraagt. Vervolgens wordt de HTML gecreëerd volgens de Models en niet andersom.

* Maakt code reusable, je kan een onderdeel een keer maken en overal implementeren.

# Hoe werken de componenten van wicket?

Om deze deelvraag te beantwoorden hebben we een prototype gebouwd. Vervolgens hebben we hieronder enkele belangrijke ondervindingen van wicket gezet.

Om een project te kunnen starten kun je een quickstart maken dat alles al goed staat door de volgende regel:

mvn archetype:generate -DarchetypeGroupId=org.apache.wicket -DarchetypeArtifactId=wicket-archetype-quickstart -DarchetypeVersion=9.0.0-M4 -DgroupId=nl.han.dea.nielsenmaarten -DartifactId=helloworld -DarchetypeRepository=https://repository.apache.org/ -DinteractiveMode=false

Hier moet je dan nog wel de groupid en artifact id aanpassen.

Dit archetype zet alles al voor je goed. Zo zet hij dependencies in je pom. Maakt hij een correcte web.xml aan. En geeft het je al een startcode om te kijken of alles werkt.

Maar als je het met de hand wilt doen moet je de volgende dependency toevoegen:

<dependency>  
 <groupId>org.apache.wicket</groupId>  
 <artifactId>wicket-core</artifactId>  
 <version>9.0.0-M4</version>  
</dependency>

We hebben hier voor versie 9.0.0-M4 gekozen omdat deze junit 5 en java 12 en 13 ondersteund.

En ook moet je enkele aanpassingen doen in je <build> settings.

Je moet namelijk de volgende regels toevoegen:

<resources>  
 <resource>  
 <filtering>false</filtering>  
 <directory>src/main/resources</directory>  
 </resource>  
 <resource>  
 <filtering>false</filtering>  
 <directory>src/main/java</directory>  
 <includes>  
 <include>\*\*</include>  
 </includes>  
 <excludes>  
 <exclude>\*\*/\*.java</exclude>  
 </excludes>  
 </resource>  
</resources>

Vervolgens moet je dan ook een web.xml aanmaken. Hier een voorbeeld van de gehele web.xml van helloworld.

<?xml version="1.0" encoding="ISO-8859-1"?>  
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://www.oracle.com/webfolder/technetwork/jsc/xml/ns/javaee/web-app\_3\_1.xsd"  
 version="3.1">  
  
 <display-name>helloworld</display-name>  
  
  
 <filter>  
 <filter-name>wicket.helloworld</filter-name>  
 <filter-class>org.apache.wicket.protocol.http.WicketFilter</filter-class>  
 <init-param>  
 <param-name>applicationClassName</param-name>  
 <param-value>nl.han.dea.nielsenmaarten.WicketApplication</param-value>  
 </init-param>  
 </filter>  
  
 <filter-mapping>  
 <filter-name>wicket.helloworld</filter-name>  
 <url-pattern>/\*</url-pattern>  
 </filter-mapping>  
</web-app>

Hier hebben wij de filter en filtermapping aan toegevoegd.  
filters bevatten de volgende attributen die je toe moet voegen:

* Filter name.   
  Hier zet je de naam van de filter neer. Let op deze moet hetzelfde zijn als de filternaam in de filter mapping.
* Filter class  
  Hier zeg je wat voor filter het is. In dit geval is het een wicketFilter.
* Param name  
  Dit geeft aan dat de value die je hierna toekent een applicatieklasse is.
* Param value  
  Waar je je applicatie hebt staan.

Filter mapping bevat het volgende:

* Filter name   
  De naam van de filter die hij moet gaan mappen.(heb je hierboven dus aangemaakt).
* url pattern  
  Naar welke url jee het moet sturen. In dit geval gewoon naar de root.

# Conclusie

# Bijlagen

# Bronnen

AIM\_ HAN. (z.d.). Onderzoeksopdracht Enterprise Applications. Geraadpleegd op 40 maart 2020, van https://onderwijsonline.han.nl/elearning/lesson/QN41kYLy

Del Bene, A., Grigorov, M., Hufe, C., kroemer, C., Bartl, D., Bors, P., … Rhode, J. (z.d.). 3 Why should I learn Wicket? 6.x. Geraadpleegd op 30 maart 2020, van https://ci.apache.org/projects/wicket/guide/6.x/guide/whyLearn.html

ProgramJava. (2010). Wicket Introduction [Videobestand]. Geraadpleegd van https://www.youtube.com/watch?v=Vt1v1P6deA4

Wikipedia contributors. (2019, 12 september). Apache Wicket. Geraadpleegd op 30 maart 2020, van https://en.wikipedia.org/wiki/Apache\_Wicket

Working with Wicket models. (2011, 14 november). Geraadpleegd op 30 maart 2020, van https://cwiki.apache.org/confluence/display/WICKET/Working+with+Wicket+models#WorkingwithWicketmodels-WhatareWicketModels?