

Apache Wicket and Java EE Sitting in a Tree

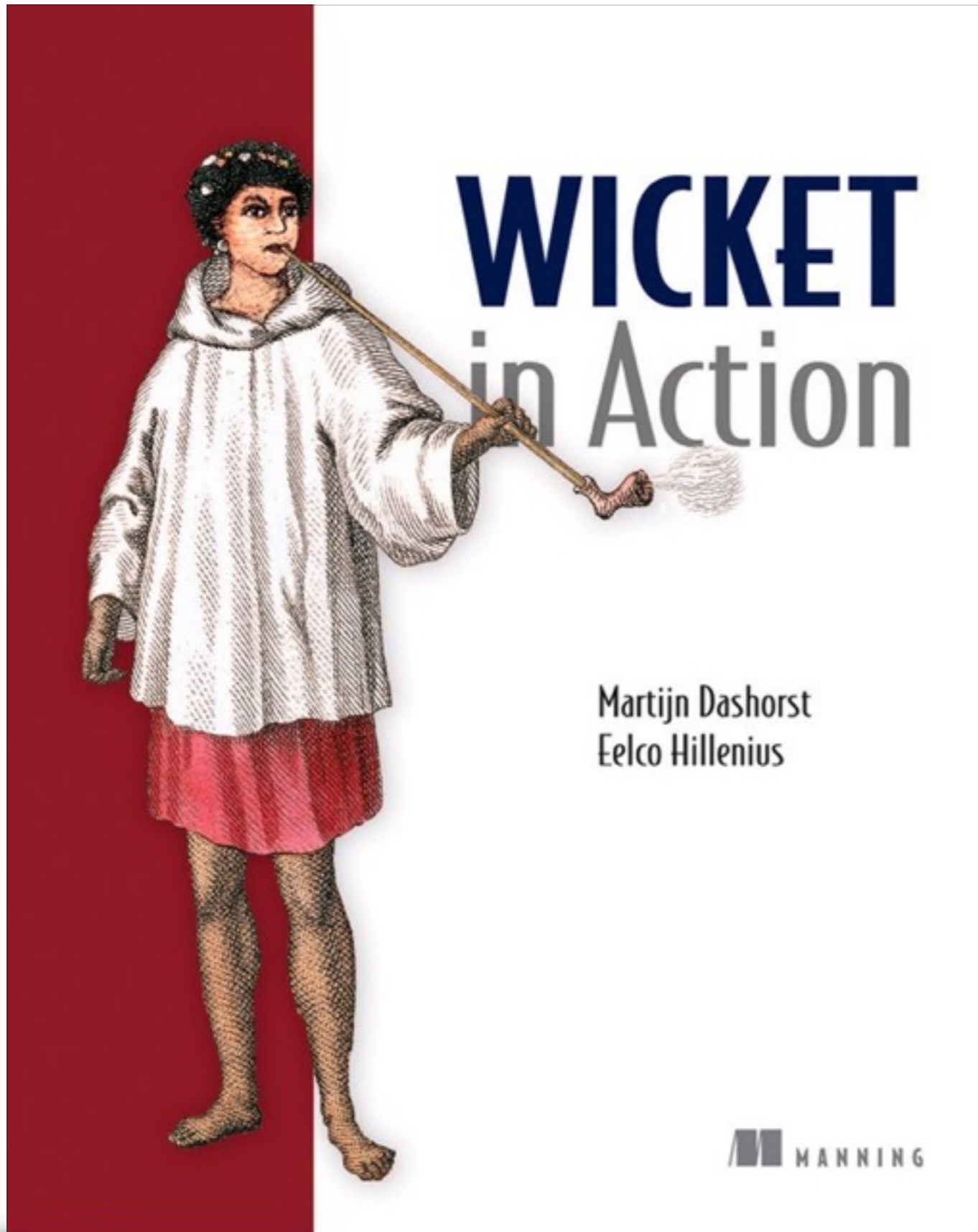
Martijn Dashorst



APACHECON
EUROPE

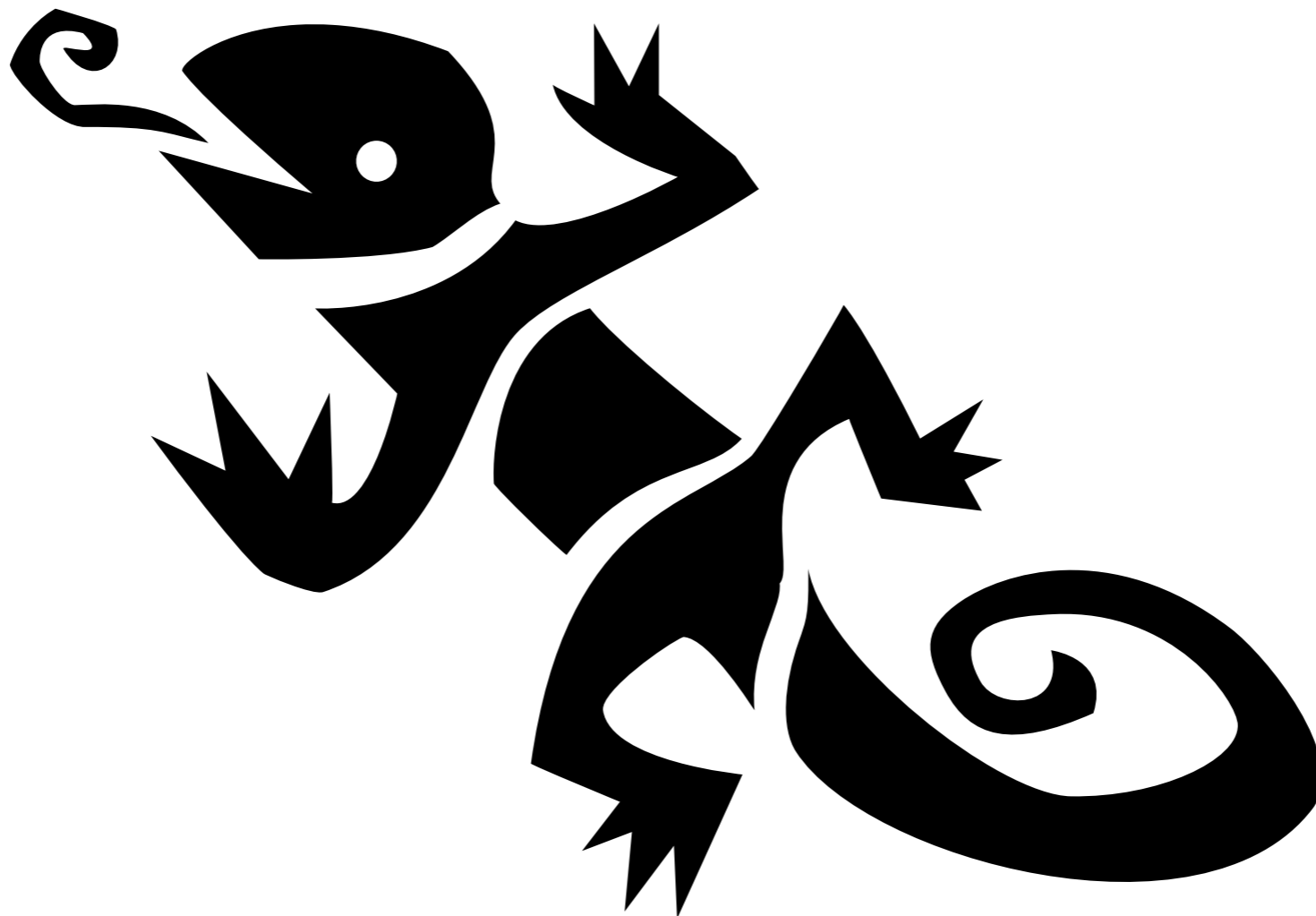
CORINTHIA HOTEL
BUDAPEST, HUNGARY
— NOVEMBER 17-21, 2014 —





co-author

10 year contributor to **Wicket**



topicus

(11 years)

Wicket and Java EE



Java EE

web profile

Servlet 3

JSP

JAX-WS

JAX-RS

JASPIC

JSR 88

JSF 2

CDI

JAX-RPC

JAXB

JACC

JSR 77

EJB 3.1

JTA

JAXR

JMS

JCA

RMI

JPA 2

Bean
Validation

SAAJ

JAAS

Java Mail

JNDI

Java EE

web profile

Servlet 3



JAX-WS

JAX-RS

JASPIC

JSR 88



CDI

JAX-RPC

JAXB

JACC

JSR 77

EJB 3.1

JTA

JAXR

JMS

JCA

RMI

JPA 2

Bean
Validation

SAAJ

JAAS

Java Mail

JNDI

Wicket's mission statement from 2004

s/[JSF|JSP]/Wicket/🌍

What is Wicket?

in 5 minutes or less



wicket | 'wɪkɪt |

noun

1 Cricket each of the sets of three stumps with two bails across the top at either end of the pitch, defended by a batsman.

- the prepared strip of ground between two sets of stumps. *when they inspected the wicket, they found it being rolled by some prisoners.*

- the dismissal of a batsman; each of ten dismissals regarded as marking a division of a side's innings: *Darlington won by four wickets.*

2 (also **wicket door** or **wicket gate**) a small door or gate, especially one beside or in a larger one.

- N. Amer. an opening in a door or wall, often fitted with glass or a grille and used for selling tickets or a similar purpose.

3 N. Amer. a croquet hoop.

PHRASES

at the wicket Cricket **1** batting: *the batsman remained at the wicket.* **2** by the wicketkeeper: *he was **caught at the wicket** chasing a wide one.*

keep wicket Cricket be a wicketkeeper.

lose a wicket Cricket (of the batting side) have a batsman dismissed. *the tourists lost their last seven wickets for 94.*

a sticky wicket Cricket a pitch that has been drying after rain and is difficult to bat on. • informal a tricky or awkward situation: *I might be **on a sticky wicket** if I used that line.*

over the wicket Cricket (referring to which side of the wicket a bowler runs when bowling) to the left of the wicket if a right-handed bowler and the right of the wicket if a left-handed bowler.

round the wicket Cricket (referring to which side of the wicket a bowler runs when bowling) to the right of the wicket if a right-handed bowler and the left of the wicket if a left-handed bowler.

take a wicket Cricket (of a bowler or a fielding side) dismiss a batsman.

ORIGIN Middle English (in the sense 'small door or grille'): from Anglo-Norman French and Old Northern French *wiket*; origin uncertain, usually referred to the Germanic root of Old Norse *vīkja* 'to turn, move'. Cricket senses date from the late 17th cent.

wicket |'wɪkɪt|

noun

1 a component oriented, open source, Java web application framework

components everywhere

The screenshot displays the EDUCUS web application interface. At the top, there is a navigation bar with tabs for 'home', 'deelnemer', 'groep', 'medewerker', 'onderwijs', 'relatie', 'financieel', and 'beheer'. The 'deelnemer' tab is active. Below the navigation bar, the user's name '106947 Emre Vegt' and a dropdown menu for '2. 94262BOL4 - Machinebouw mechatronica (Mechatronicus)' are visible. A secondary navigation bar contains various menu items like 'Deelnemerkaart', 'Personalia', 'Verbintenissen', etc. The main content area features a table titled 'Verbintenissen en plaatsingen - 1 t/m 2 (van 2)'. Below this table, there are two detailed sections: 'Verbintenis' and 'Vrije velden - 1 t/m 1 (van 1)'. The 'Verbintenis' section lists various attributes such as Status, BRON-status, Opleiding, and Naam. The 'Vrije velden' section shows a table with 'Naam' and 'Waarde' columns. On the right side, there is a sidebar with sections for 'Recente deelnemers', 'Mededelingen', 'Favorieten', 'Rapportages', and 'Geschiedenis'. At the bottom, there are several buttons: 'Proces starten', 'Nieuwe verbintenis', 'Her-intake', 'Nieuwe plaatsing', and 'Bewerken'. The footer contains performance metrics: 'Server tijd: 0.001s, Client tijd: 0.198s (0.08s parsen + 0.118s renderen)'.

Volgnr.	Opleiding	Schooljaar	Leerjaar	Praktijkjaar	Plaatsingsgroep	Code	Status	Bekostigd	Van	T/m	Fase	Leerweg	Reden beëindigen
2	Machinebouw mechatronica (Mechatronicus)	2014/2015				94262	Voorlopig	Nee	01-08-2014	(verwacht 31-07-2016)		BOL	
								Nee	01-08-2014				

Naam	Waarde
Studietaal	Nederlands

Attribuut	Waarde
Status	Voorlopig
BRON-status	Geen
BRON-datum	
Opleiding	94262BOL4 - Machinebouw mechatronica (Mechatronicus)
Crebo-code	94262
Indicatie gehandicapt	Nee
Naam	Machinebouw mechatronica (Mechatronicus)
Code	94262BOL4
Leerweg	BOL
Niveau	Niveau 4
Intensiteit	Voltijd
Organisatie-eenheid	R4 Techniek
Locatie	Plantsoen 72, Wageningen

components everywhere

The screenshot displays the EDUCUS web application interface. At the top, there is a navigation menu with tabs for 'home', 'deelnemer', 'groep', 'medewerker', 'onderwijs', 'relatie', 'financieel', and 'beheer'. The 'deelnemer' tab is active. The user is identified as '106947 Emre Vegt' and is viewing the course '2. 94262BOL4 - Machinebouw mechatronica (Mechatronica)'. The interface includes a progress bar showing '0 Credits' out of a total of 120. Below this, there is a table showing progress per phase for 'Leerjaar 1' and 'Leerjaar 2', both at 0% completion. A table of 'Productregels' lists various courses with their status and credits. The right sidebar contains sections for 'Recente deelnemers', 'Mededelingen', 'Favorieten', 'Rapportages', and 'Geschiedenis'. The footer shows server and client performance metrics.

current.ontwikkel.eduarte.nl

home deelnemer groep medewerker onderwijs relatie financieel beheer

106947 Emre Vegt 2. 94262BOL4 - Machinebouw mechatronica (Mechatronica)

Deelnemerkaart Personalia Verbindingen CGO Begeleiding Financieel Onderwijs Resultaten Examen BRON Aanwezigheid

HUIDIGE FASE TOTALE STUDIE

0 Credits 30 60 90 120

Tip: ga met uw muis over de onderdelen om te zien waar deze voor staan.

Fase... Formatief

		Voortgang per fase (behaald / afgenomen / benodigd)			Voorf.cr.
Leerjaar 1		0 / 60 / 60	= 0%	(0 vrijstellingen)	0
Leerjaar 2		0 / 0 / 60	= 0%	(0 vrijstellingen)	0

Productregels - 1 t/m 6 (van 6)

Soort	Afkorting	Naam	Keuze	Resultaat	Cohort	Vrijstelling	Formatief	Cr.	Cr.ond.pr.	Beh.cr.
Fase: Leerjaar 1 (6)										
BG	BG1	Beroepsgericht vak 1	BG1 - Beroepsgericht vak 1		2014/2015	Geen	Ja	12	12	0
BG	BG2	Beroepsgericht vak 2	BG2 - Beroepsgericht vak 2		2014/2015	Geen	Ja	12	12	0
BG	BG3	Beroepsgericht vak 3	BG3 - Beroepsgericht vak 3		2014/2015	Geen	Ja	15	15	0
AVO	AVO1	Algemeen vak 1	AVO1 - Algemeen vak 1		2014/2015	Geen	Ja	7	7	0
AVO	AVO2	Algemeen vak 2	AVO2 - Algemeen vak 2		2014/2015	Geen	Ja	8	8	0
Stage	ST1	Stagevak 1	ST1 - Stagevak 1		2014/2015	Geen	Ja	6	6	0

Collectief cohort/vrijstelling aanpassen

Bewerken

Server tijd: 0.002s, Client tijd: 0.247s (0.082s parsen + 0.165s renderen)

components everywhere

The screenshot displays the EDUCUS web application interface. At the top, there is a navigation bar with tabs for 'home', 'deelnemer', 'groep', 'medewerker', 'onderwijs', 'relatie', 'financieel', and 'beheer'. A search bar and the EDUCUS logo are also present. Below the navigation bar, there is a 'Menu...' dropdown and a 'Sorteren op' section set to 'Categorie'. The main content area is divided into several sections: 'recente pagina's' (recent pages), 'Beheer' (Management) with a list of tasks like 'Activiteitenoverzicht' and 'Email-templates', 'Organisatie' (Organization) with options like 'Locaties' and 'Organisatie algemeen', 'Rapportage' (Reporting) with 'Analyse' and 'Analyseopdrachten', 'Resultaten' (Results) with 'Formuletemplates' and 'Schalen', and 'Systeem' (System) with 'Afgenomen modules'. A sidebar on the right contains sections for 'Recente deelnemers' (Recent participants), 'Mededelingen' (Notifications), 'Documenten' (Documents), 'Favorieten' (Favorites), 'Rapportages' (Reports), and 'Geschiedenis' (History). The footer shows performance metrics: 'Server tijd: 0.001s, Client tijd: 0.226s (0.083s parsen + 0.143s renderen)'.

Injection (EJB, CDI)

Conversations (CDI)

JPA

Bean validation

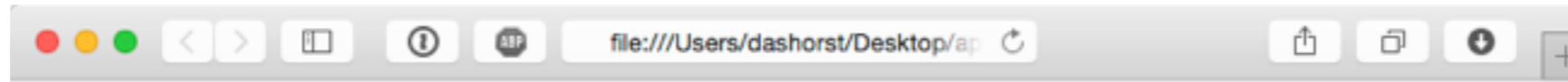
Arquillian



Cheese Store

Example





Gouda

Named after the Dutch town of Gouda, just outside Rotterdam. It accounts for more than 60% of the cheese produced in Holland and it has a very long history. Gouda is a traditional, creamery, hard cheese. It is round with very smooth, yellow, waxed rind. The flavor is sweet and fruity. As time passes, the taste intensifies and becomes more complex. Mature Gouda (18 months plus) is coated in black wax which provides a stark contrast to the deep yellow interior. Gouda is considered to be one of the world's great cheeses. It is both a table cheese and a dessert cheese, excellent with fruit and wine. Gouda is now made globally in a style similar to the creation of Edam.

\$1.99 [Add to cart](#)

Edam

This is a pressed, semi-hard to hard cheese, made from cow's milk. It comes in a shape of ball covered with distinctive red wax. Edam is produced from skimmed or semi-skimmed milk. It is usually consumed young, when the texture is elastic and supple and the flavor is smooth, sweet and nutty. Black-wax coating means that Edam has been matured for at least 17 weeks. The cheese tastes delicious with a glass of Pinot Noir.

\$2.99 [Add to cart](#)

Camembert

A very famous French cheese, Camembert dates back to the 18th century and is named for a Norman village in which there is a statue of the creator of this particular variety (Marie Harel). Originally, this cheese was dry and yellow-brown, but after a few modifications it

Your Selection

Gouda \$1.99 [remove](#)

Total \$1.99

[Check out](#)



- Name is required

Check out

Please enter your billing address.

Name

Street

City

State

Your Selection

Gouda \$1.99 [remove](#)
Total \$1.99

First steps

1. Setup project
2. Add 'domain layer'
3. Add components



1. setup the project





Creating the project - with Maven

To create your project, copy and paste the command line generated after typing in the groupId, artifactId and version.

GroupId: (?)
 ArtifactId: (?)
 Version: (?)
 Server: (?)

```
mvn archetype:generate -DarchetypeGroupId=org.apache.wicket -
DarchetypeArtifactId=wicket-archetype-quickstart -
DarchetypeVersion=6.18.0 -DgroupId=com.cheesr -DartifactId=cheesr -
DarchetypeRepository=https://repository.apache.org/ -
Dlog4j.properties=wildfly-doesnt-need-log4j.properties -
DinteractiveMode=false
```

Command Line:

Results

This will produce the following project structure/files:

```
.\myproject
| pom.xml
|
|---src
|   +---main
|   |   +---java
|   |   |   \---com
|   |   |       \---mycompany
|   |   |           HomePage.html
|   |   |           HomePage.java
|   |   |           WicketApplication.java
|   |   |
|   |   +---resources
|   |   |   log4j.properties
|   |   |
|   |   \---webapp
|   |       \---WEB-INF
|   |           web.xml
|   |
|   \---test
```

- g Lists
- Learn
- Guide
- amples
- onents
- rojects
- Wiki
- guide
- Books
- IDEs
- ases
- et 6.18
- ket 1.5
- ket 1.4
- ket 1.3
- ket 1.2
- ket 1.1
- ket 1.0
- Docs
- ket 6.x
- ket 1.5
- ket 1.4
- ket 1.3
- et 7.x
- oad M4
- guide
- ocs 7.x
- ibute
- g docs
- Wicket
- atch
- Wicket
- isheye



GroupId: (?)

ArtifactId: (?)

Version: (?)

Server: (?)

Command Line:

```
mvn archetype:generate -DarchetypeGroupId=org.apache.wicket -DarchetypeArtifactId=wicket-archetype-quickstart -DarchetypeGroupId=com.cheesr -DartifactId=cheesr -DarchetypeRepository=https://repository.apache.org/ -Ddoesnt-need-log4j.properties -DinteractiveMode=false
```

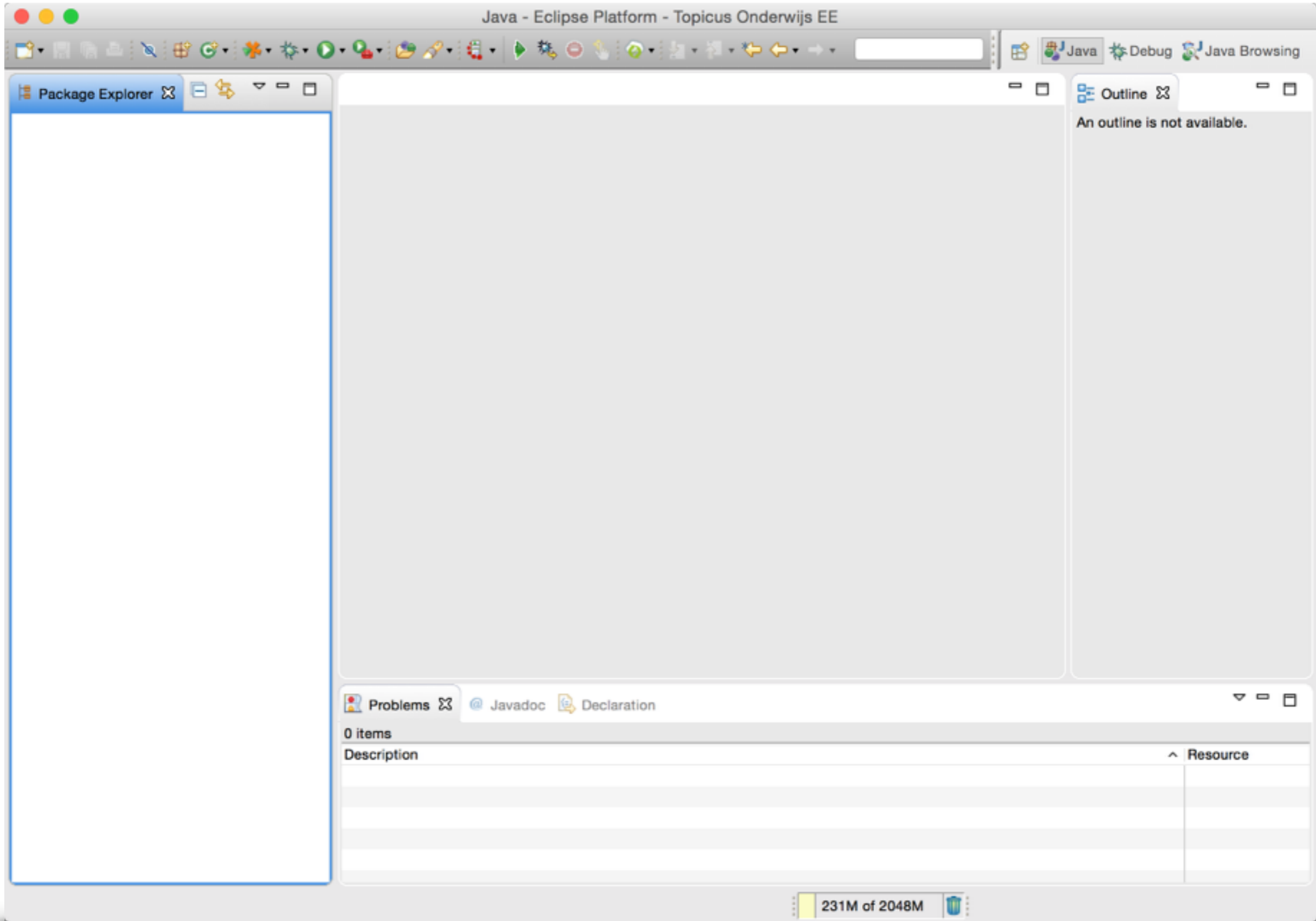


```
dashorst@Martijns-MacBook-Pro-2 ~/Workspace$ █
```



```
dashorst@Martijns-MacBook-Pro-2 ~/Workspace$ mvn archetype:generate -DarchetypeGroupId=org.apache.wicket -DarchetypeArtifactId=wicket-archetype-quickstart -DarchetypeVersion=6.18.0 -DgroupId=com.cheesr -DartifactId=cheesr -DarchetypeRepository=https://repository.apache.org/ -Dlog4j.properties=wildfly-doesnt-need-log4j.properties -DinteractiveMode=false
```

```
dashorst@Martijns-MacBook-Pro-2 ~/Workspace$ mvn archetype:generate -DarchetypeGroupId=org.apache.wicket -DarchetypeArtifactId=wicket-archetype-quickstart -DarchetypeVersion=6.18.0 -DgroupId=com.cheesr -DartifactId=cheesr -DarchetypeRepository=https://repository.apache.org/ -Dlog4j.properties=wildfly-doesnt-need-log4j.properties -DinteractiveMode=false
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building Maven Stub Project (No POM) 1
[INFO] -----
[INFO]
[INFO] >>> maven-archetype-plugin:2.2:generate (default-cli) > generate-sources @ standalone-pom >>>
[INFO]
[INFO] <<< maven-archetype-plugin:2.2:generate (default-cli) < generate-sources @ standalone-pom <<<
[INFO]
[INFO] --- maven-archetype-plugin:2.2:generate (default-cli) @ standalone-pom ---
[INFO] Generating project in Batch mode
[INFO] Archetype defined by properties
[INFO]
[INFO] -----
[INFO] Using following parameters for creating project from Archetype: wicket-archetype-quickstart:6.18.0
[INFO] -----
[INFO] Parameter: groupId, Value: com.cheesr
[INFO] Parameter: artifactId, Value: cheesr
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Parameter: package, Value: com.cheesr
[INFO] Parameter: packageInPathFormat, Value: com/cheesr
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Parameter: package, Value: com.cheesr
[INFO] Parameter: groupId, Value: com.cheesr
[INFO] Parameter: log4j.properties, Value: wildfly-doesnt-need-log4j.properties
[INFO] Parameter: artifactId, Value: cheesr
[INFO] project created from Archetype in dir: /Users/dashorst/Workspace/cheesr
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO]
[INFO] Total time: 0.885 s
[INFO] Finished at: 2014-11-15T00:02:40+01:00
[INFO] Final Memory: 11M/246M
[INFO] -----
dashorst@Martijns-MacBook-Pro-2 ~/Workspace$ █
```



Import

Select

Import Existing Maven Projects



Select an import source:

- ▶ EJB
- ▶ Git
- ▼ Install
 - From Existing Installation
 - Install Software Items from File
- ▶ Java EE
- ▼ Maven
 - Check out Maven Projects from SCM
 - Existing Maven Projects**
 - Install or deploy an artifact to a Maven repository
 - Materialize Maven Projects from SCM
- ▶ Plug-in Development
- ▶ Run/Debug
- ▶ Team
- ▶ Web
- ▶ Web services
- ▶ XML



< Back

Next >

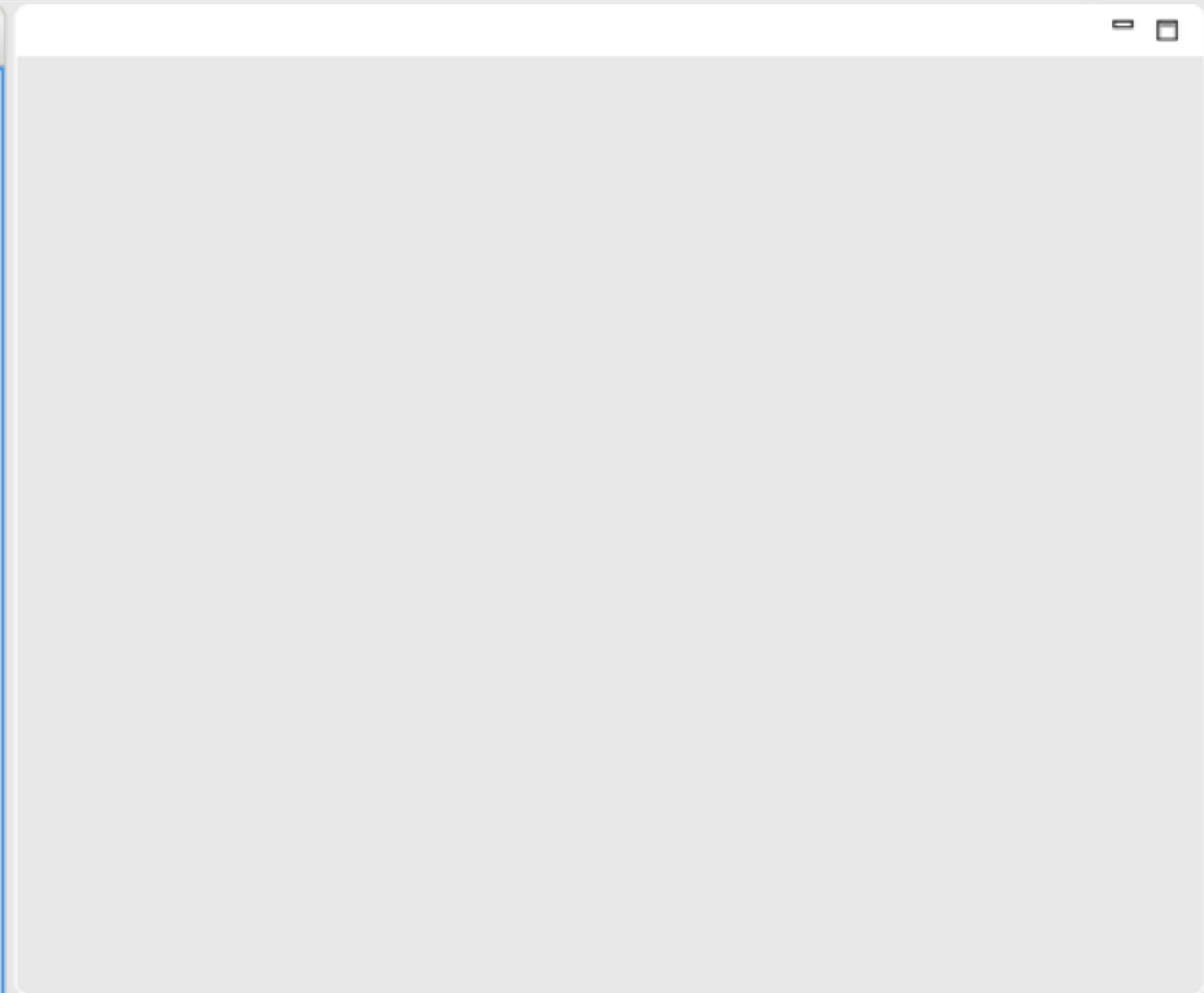
Cancel

Finish



Package Explorer

- cheesr
 - src/main/java
 - com.cheesr
 - HomePage.java
 - WicketApplication.java
 - HomePage.html
 - src/main/resources
 - src/test/java
 - src/test/resources
 - src
 - main
 - webapp
 - WEB-INF
 - web.xml
 - logo.png
 - style.css
 - test
 - target
 - pom.xml



Outline

An outline is not available.

Problems Javadoc Declaration

0 items

Description	Resource



Package Explorer

- cheesr
 - src/main/java
 - com.cheesr
 - WicketApplication.java
 - src/main/resources
 - src/test/java
 - com.cheesr
 - src/test/resources
 - src
 - main
 - webapp
 - WEB-INF
 - web.xml
 - test
 - target
 - pom.xml

Outline

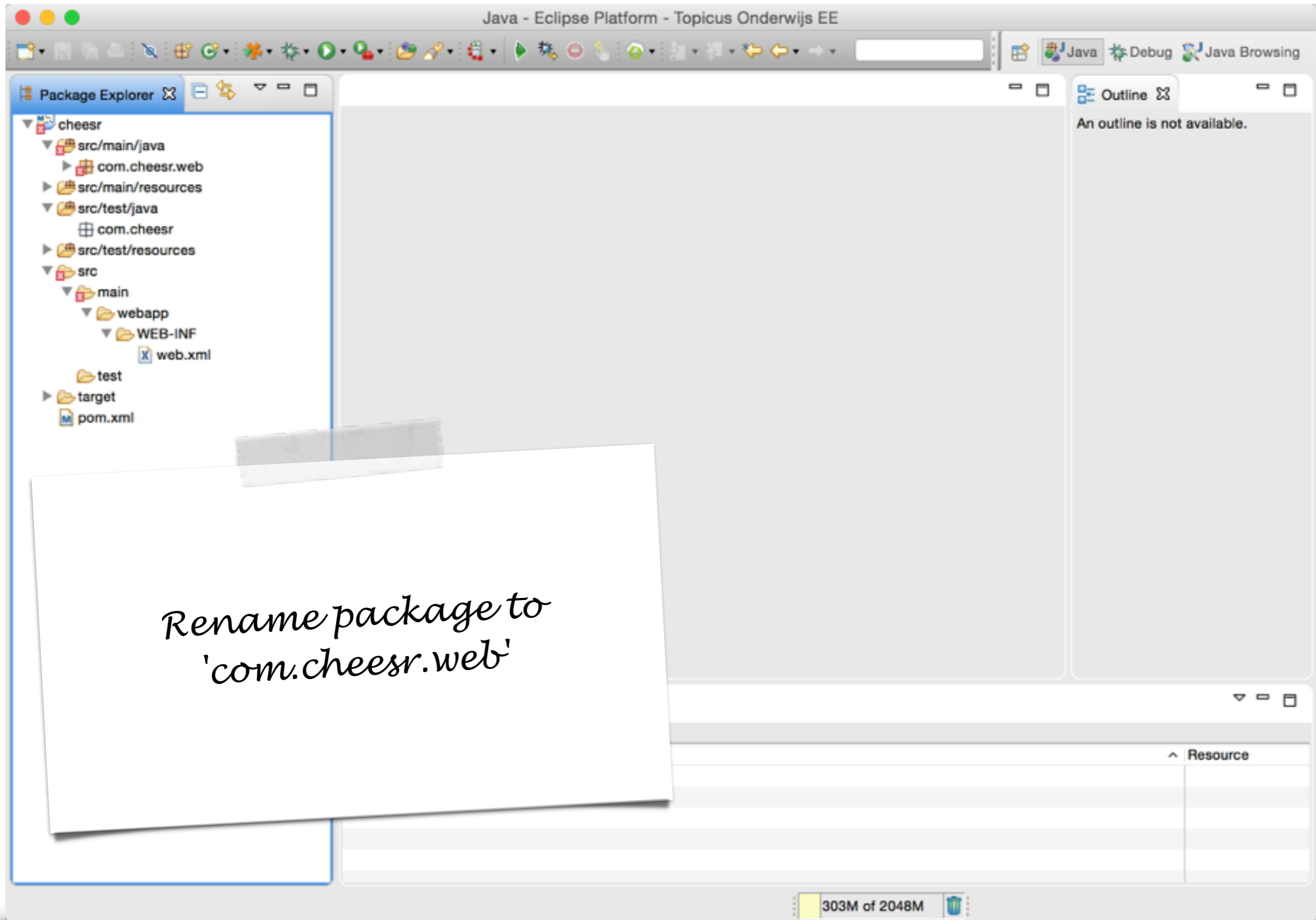
An outline is not available.

Problems

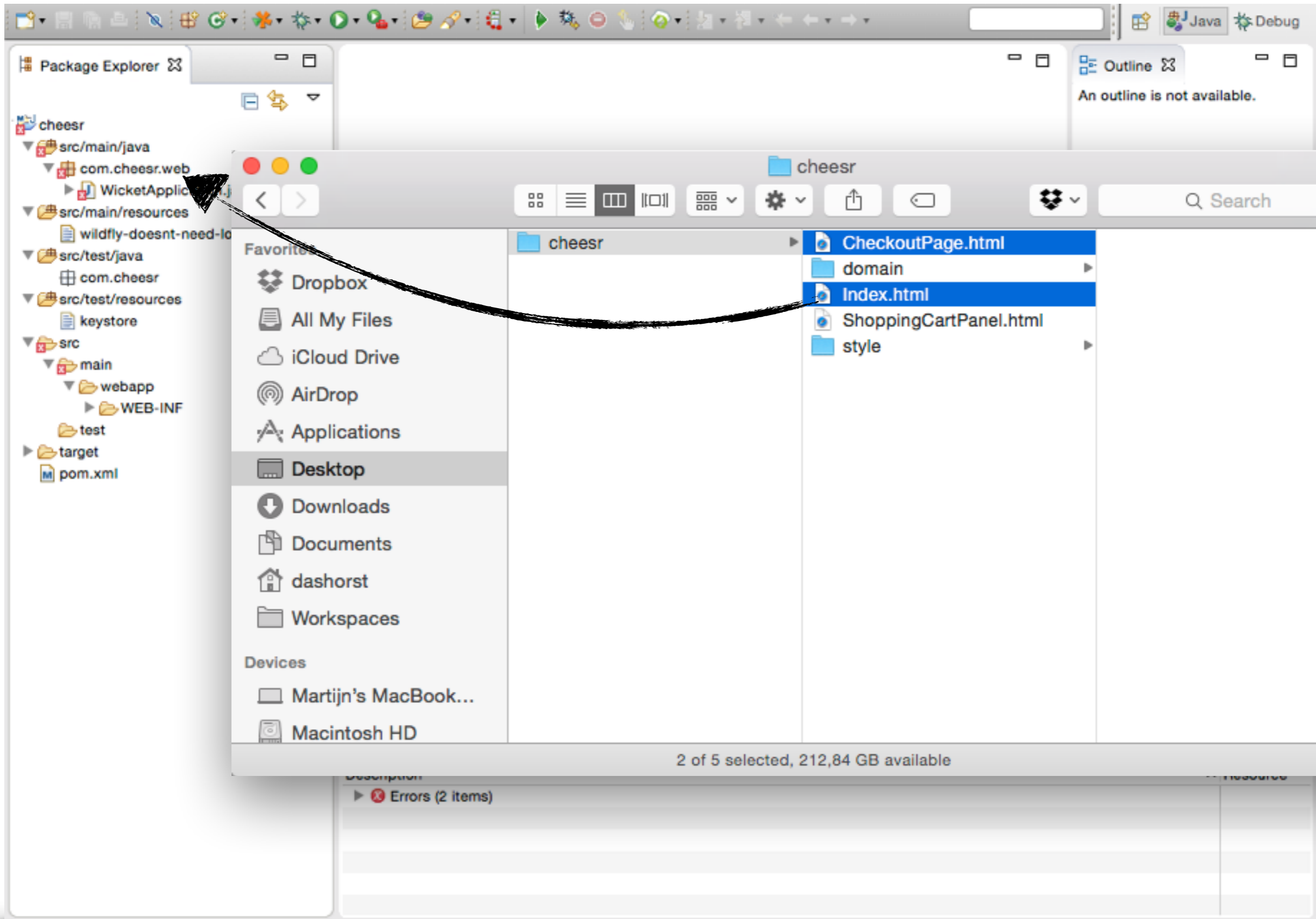
2 errors, 0 warnings, 0 others

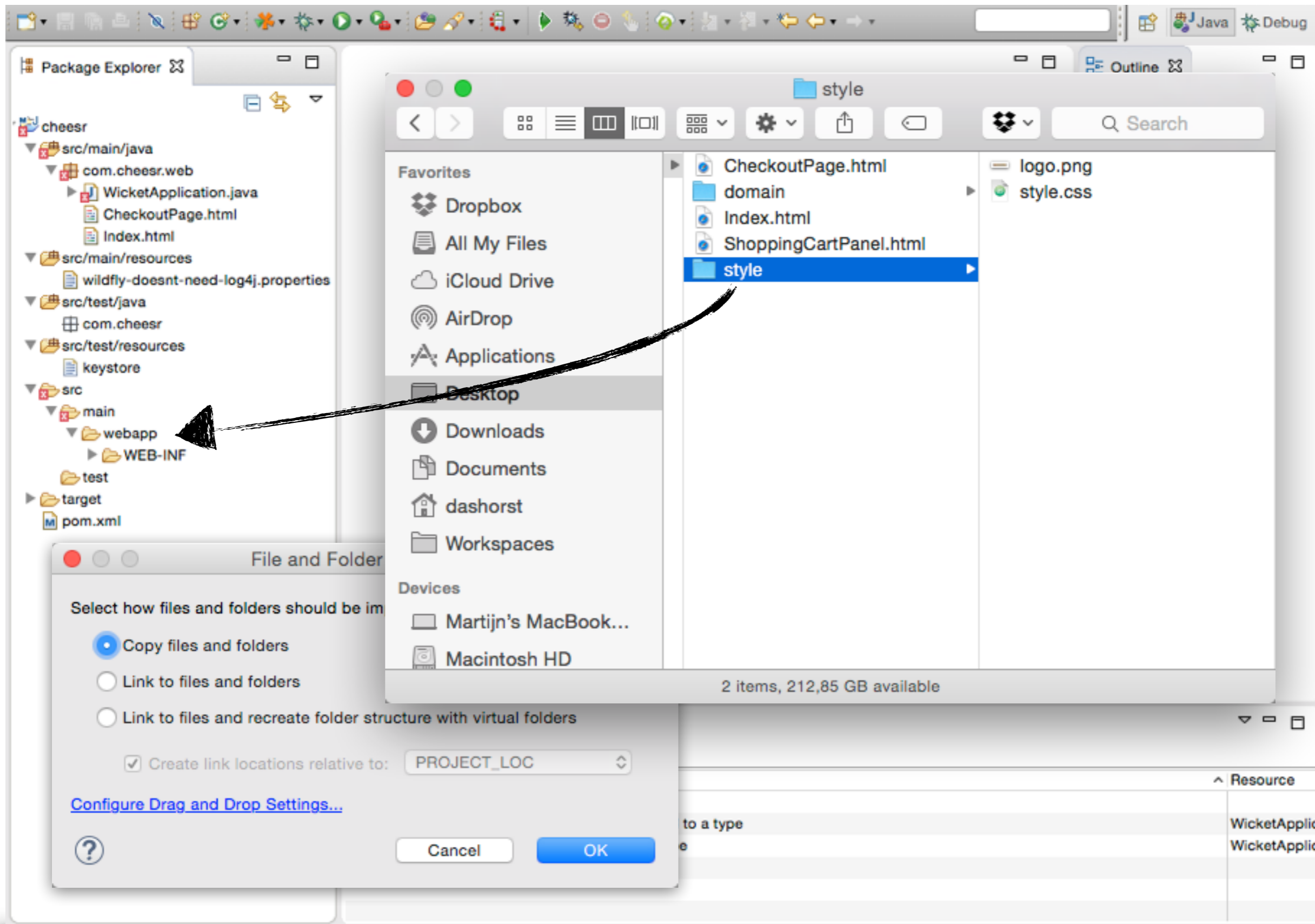
Description	Resource
▶ Errors (2 items)	

Remove unnecessary files



*Rename package to
'com.cheesr.web'*





New Java Class

Java Class
Create a new Java class.

Source folder:

Package:

Enclosing type:

Name:

Modifiers: public package private protected
 abstract final static

Superclass:

Interfaces:

Which method stubs would you like to create?

public static void main(String[] args)
 Constructors from superclass
 Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))

Generate comments

Package Explorer

- cheers
 - src/main/java
 - com.cheersr.web
 - WicketApplication.java
 - CheckoutPage.html
 - Index.html
 - src/main/resources
 - wildfly-doesnt-need-log4j.pro
 - src/test/java
 - com.cheersr
 - src/test/resources
 - keystore
 - src
 - main
 - webapp
 - style
 - logo.png
 - style.css
 - WEB-INF
 - test
 - target
 - pom.xml

Outline

An outline is not available.

Resource

- WicketApplic
- WicketApplic

```
package com.cheesr.web;

import org.apache.wicket.markup.html.WebPage;

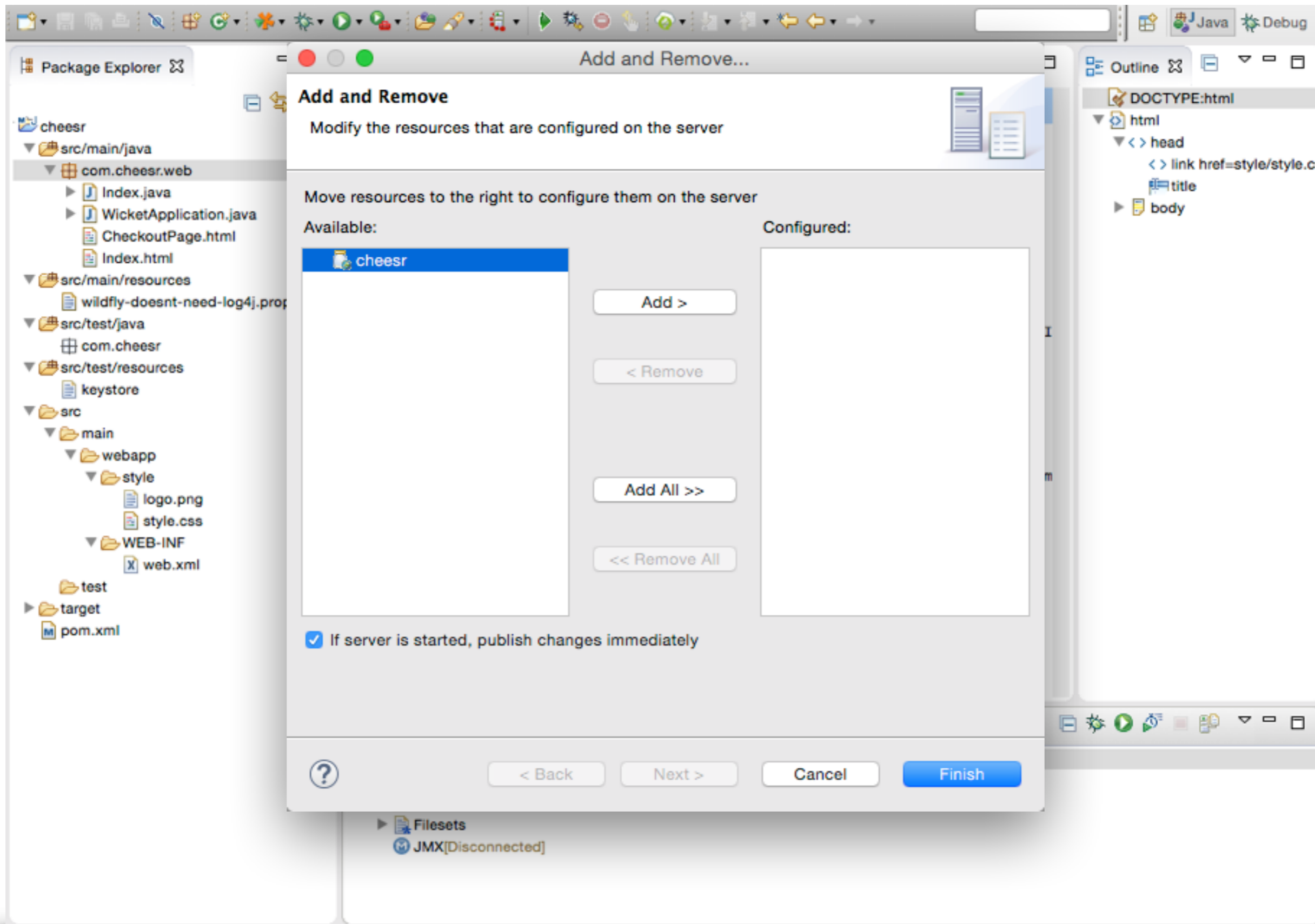
public class Index extends WebPage {
    public Index() {
    }
}
```

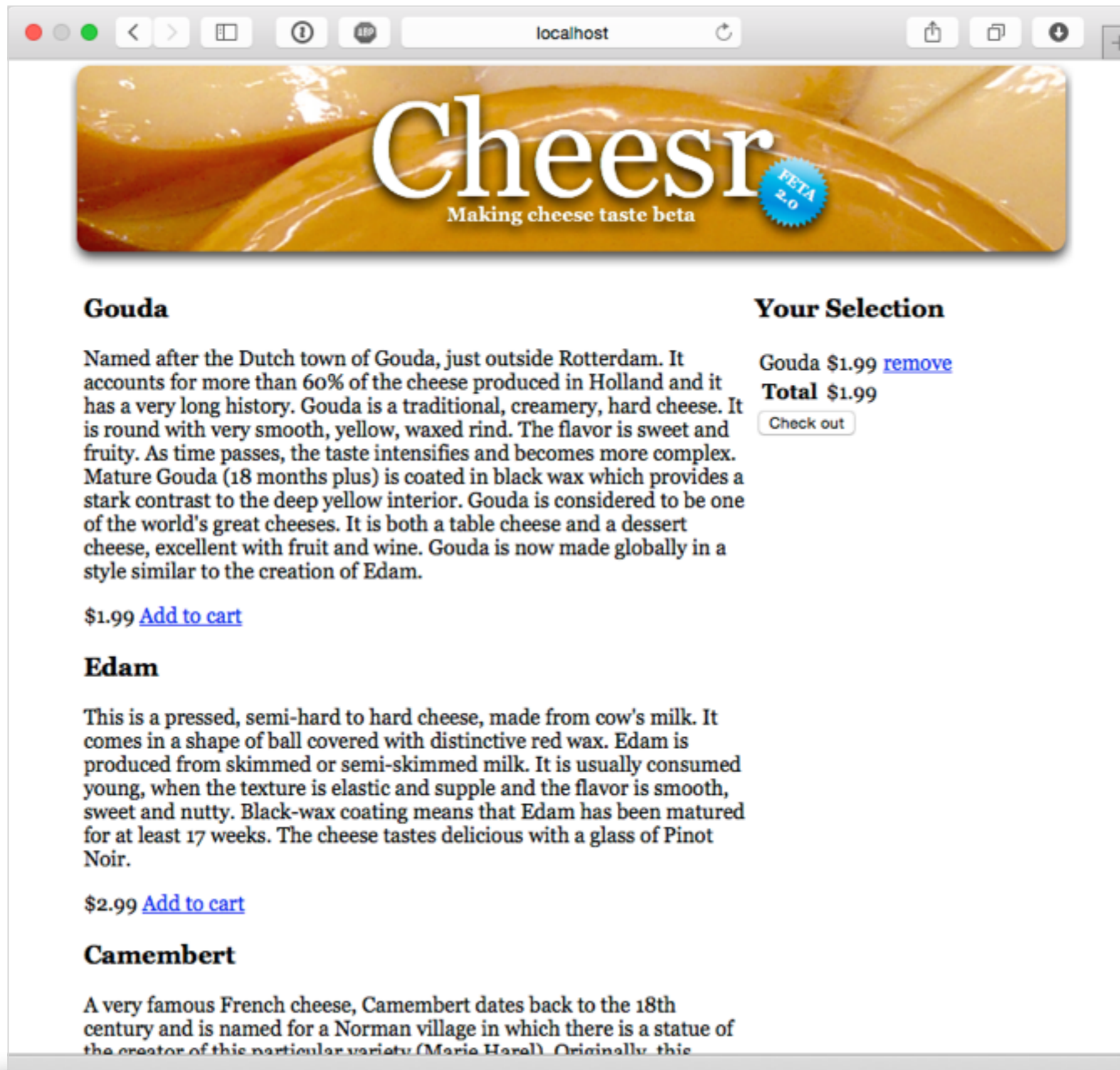
```
package com.cheesr.web;

import org.apache.wicket.markup.html.WebPage;
import org.apache.wicket.protocol.http.WebApplica

public class WicketApplication extends WebApplica
    @Override
    public Class<? extends WebPage> getHomePage() {
        return Index.class;
    }

    @Override
    public void init() {
        super.init();
    }
}
```





Cheesr

Making cheese taste beta



Gouda

Named after the Dutch town of Gouda, just outside Rotterdam. It accounts for more than 60% of the cheese produced in Holland and it has a very long history. Gouda is a traditional, creamery, hard cheese. It is round with very smooth, yellow, waxed rind. The flavor is sweet and fruity. As time passes, the taste intensifies and becomes more complex. Mature Gouda (18 months plus) is coated in black wax which provides a stark contrast to the deep yellow interior. Gouda is considered to be one of the world's great cheeses. It is both a table cheese and a dessert cheese, excellent with fruit and wine. Gouda is now made globally in a style similar to the creation of Edam.

\$1.99 [Add to cart](#)

Edam

This is a pressed, semi-hard to hard cheese, made from cow's milk. It comes in a shape of ball covered with distinctive red wax. Edam is produced from skimmed or semi-skimmed milk. It is usually consumed young, when the texture is elastic and supple and the flavor is smooth, sweet and nutty. Black-wax coating means that Edam has been matured for at least 17 weeks. The cheese tastes delicious with a glass of Pinot Noir.

\$2.99 [Add to cart](#)

Camembert

A very famous French cheese, Camembert dates back to the 18th century and is named for a Norman village in which there is a statue of the creator of this particular variety (Marie Harel). Originally, this

Your Selection

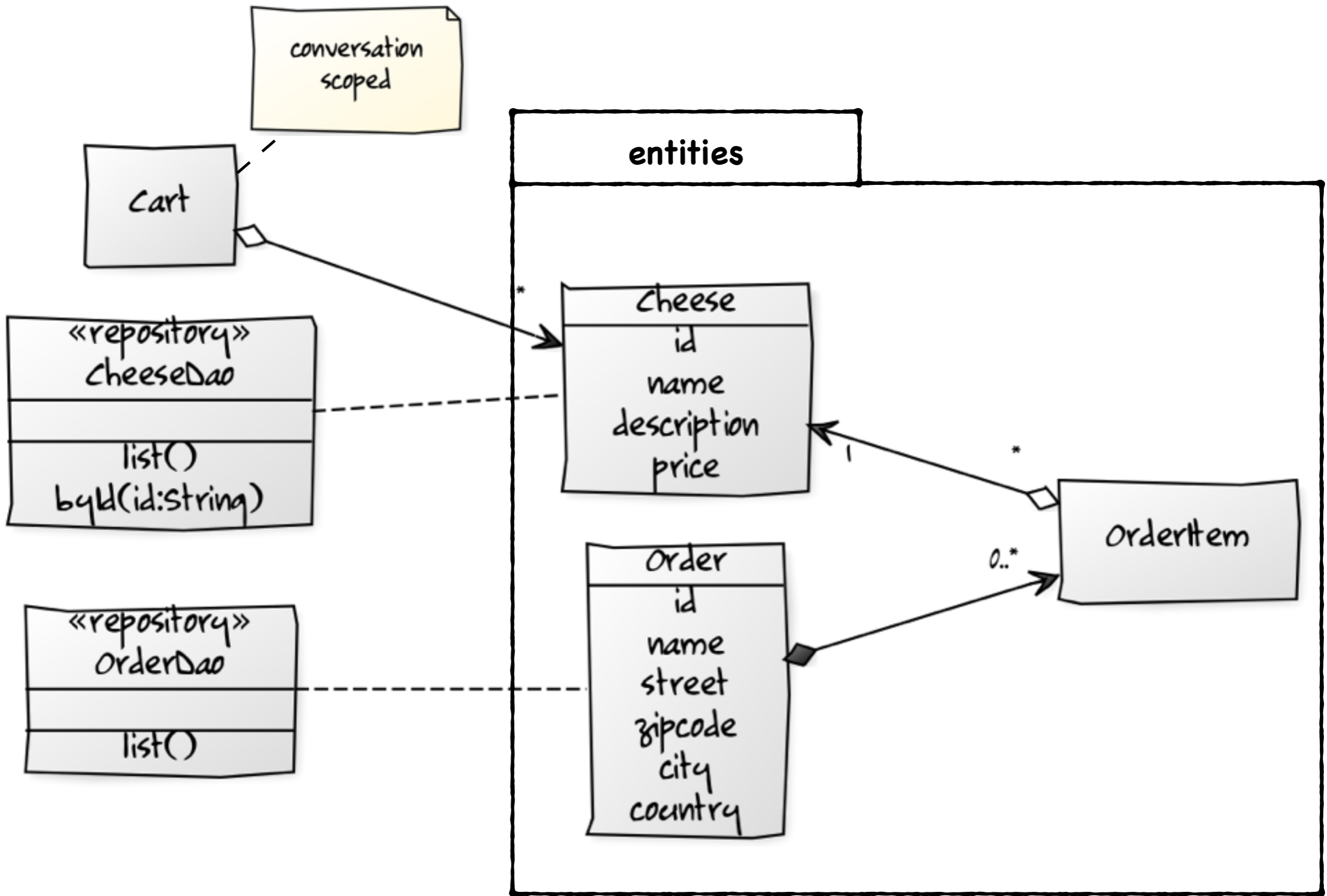
Gouda \$1.99 [remove](#)

Total \$1.99

[Check out](#)

2. Add 'domain model'






```
package com.cheesr.dao;
```

```
public class CheeseDao {  
}
```

```
package com.cheesr.dao;
```

```
public class CheeseDao {  
    public List<Cheese> getCheeses() {  
    }  
}
```

```
package com.cheesr.dao;  
/*...*/  
public class CheeseDao {  
    public List<Cheese> getCheeses() {  
        return asList();  
    }  
}
```

```
package com.cheesr.dao;  
/*...*/  
public class CheeseDao {  
    public List<Cheese> getCheeses() {  
        return asList(  
            new Cheese());  
    }  
}
```

```
package com.cheesr.dao;
/*...*/
public class CheeseDao {
    public List<Cheese> getCheeses() {
        return asList(
            new Cheese(
                "Gouda",
                "Named after the Dutch town of Gouda, j",
                1.99));
    }
}
```

```
package com.cheesr.dao;
/*...*/
public class CheeseDao {
    public List<Cheese> getCheeses() {
        return asList(
            new Cheese(
                "Gouda",
                "Named after the Dutch town of Gouda, j",
                1.99),
            new Cheese(
                "Edam",
                "This is a pressed, semi-hard to hard c",
                2.99));
    }
}
```

```
package com.cheesr.dao;
/*...*/
public class CheeseDao {
    public List<Cheese> getCheeses() {
        return asList(
            new Cheese(
                "Gouda",
                "Named after the Dutch town of Gouda, j",
                1.99),
            new Cheese(
                "Edam",
                "This is a pressed, semi-hard to hard c",
                2.99),
            new Cheese(
                "Camembert",
                "A very famous French cheese, Camembert",
                3.99));
    }
}
```

```
package com.cheesr.dao;
```

```
public class OrderDao {  
}
```



```
package com.cheesr.dao;  
  
public class OrderDao {  
    public void save(Order order) {  
    }  
}
```

```
package com.cheesr.dao;

public class OrderDao {
    private List<Order> orders = new ArrayList<>();

    public void save(Order order) {
    }
}
```

```
package com.cheesr.dao;

public class OrderDao {
    private List<Order> orders = new ArrayList<>();

    public void save(Order order) {
        orders.add(order);
    }
}
```

```
package com.cheesr.dao;

public class OrderDao {
    private List<Order> orders = new ArrayList<>();

    public void save(Order order) {
        orders.add(order);
    }

    public List<Order> list() {
    }
}
```

```
package com.cheesr.dao;

public class OrderDao {
    private List<Order> orders = new ArrayList<>();

    public void save(Order order) {
        orders.add(order);
    }

    public List<Order> list() {
        return Collections.unmodifiableList(orders);
    }
}
```

3. Add components





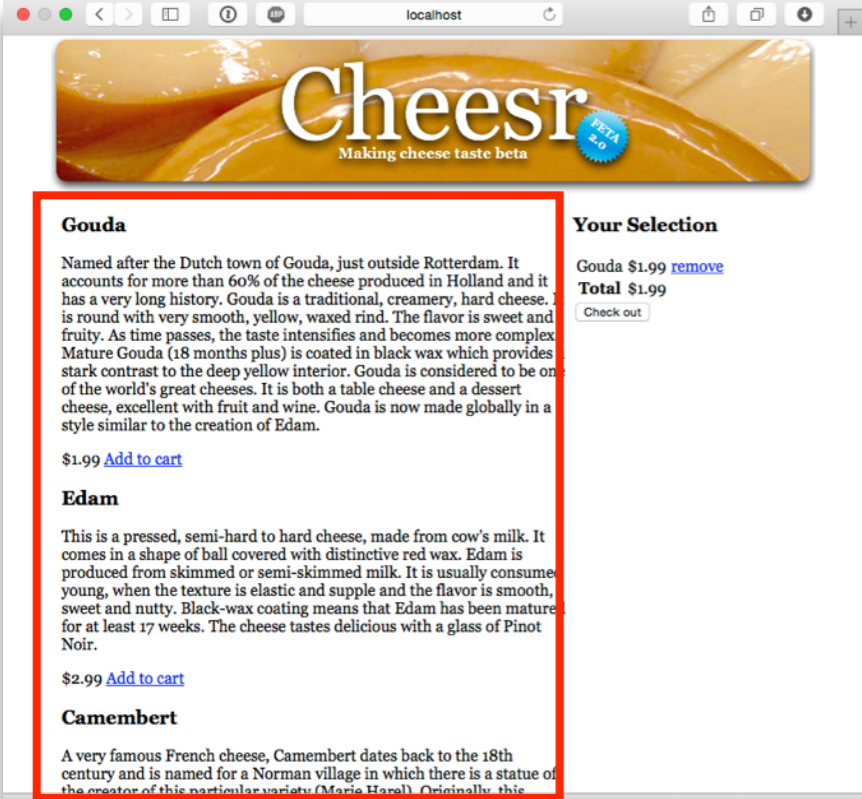
Package Explorer

- cheesr
 - src/main/java
 - com.cheesr
 - dao

```

5 <title>Cheesr - Making cheese taste beta</title>
6 </head>
7 <body>
8 <div id="container">
9 <div id="header"><h1>Cheesr</h1></div>
10 <div id="content">
11 <div id="main">
    <div class="cheese">
      <h3>Gouda</h3>
      <p>Named after the Dutch town of Gouda, just outside Rotterdam. It accounts for more than 60%
      <p>
        <span>$1.99</span>
        <a href="#">Add to cart</a>
      </p>
    </div>
    <div class="cheese">
      <h3>Edam</h3>
      <p>This is a pressed, semi-hard to hard cheese, made from cow's milk. It comes in a shape of
      <p>
        <span>$2.99</span>
        <a href="#">Add to cart</a>
      </p>
    </div>
    <div class="cheese">
      <h3>Camembert</h3>
      <p>A very famous French cheese, Camembert dates back to the 18th century and is named for a M
      <p>
        <span>$0.99</span>
        <a href="#">Add to cart</a>
      </p>
    </div>
  </div>
  <div id="cart">
    <div>
  </div>

```



Package Explorer

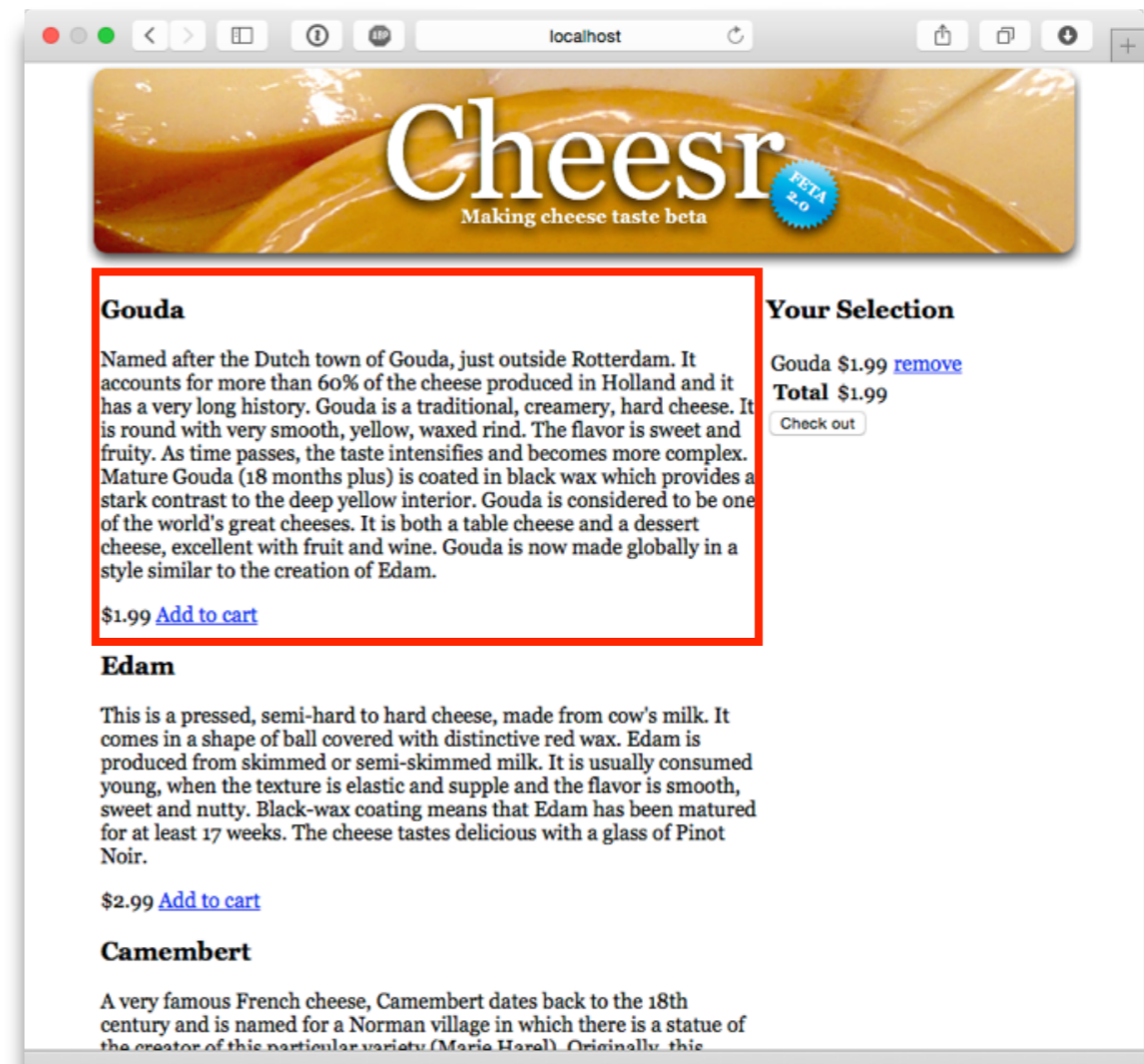
- style
 - logo.png
 - style.css
- WEB-INF
 - web.xml
- test
 - target
 - pom.xml
- cheesr-javac
 - src/main/java
 - src/main/resources
 - src/test/java
 - src/test/resources

Problems @ Javadoc Declaration Servers Console

0 errors, 5 warnings, 0 others

Description	Resource
Warnings (5 items)	

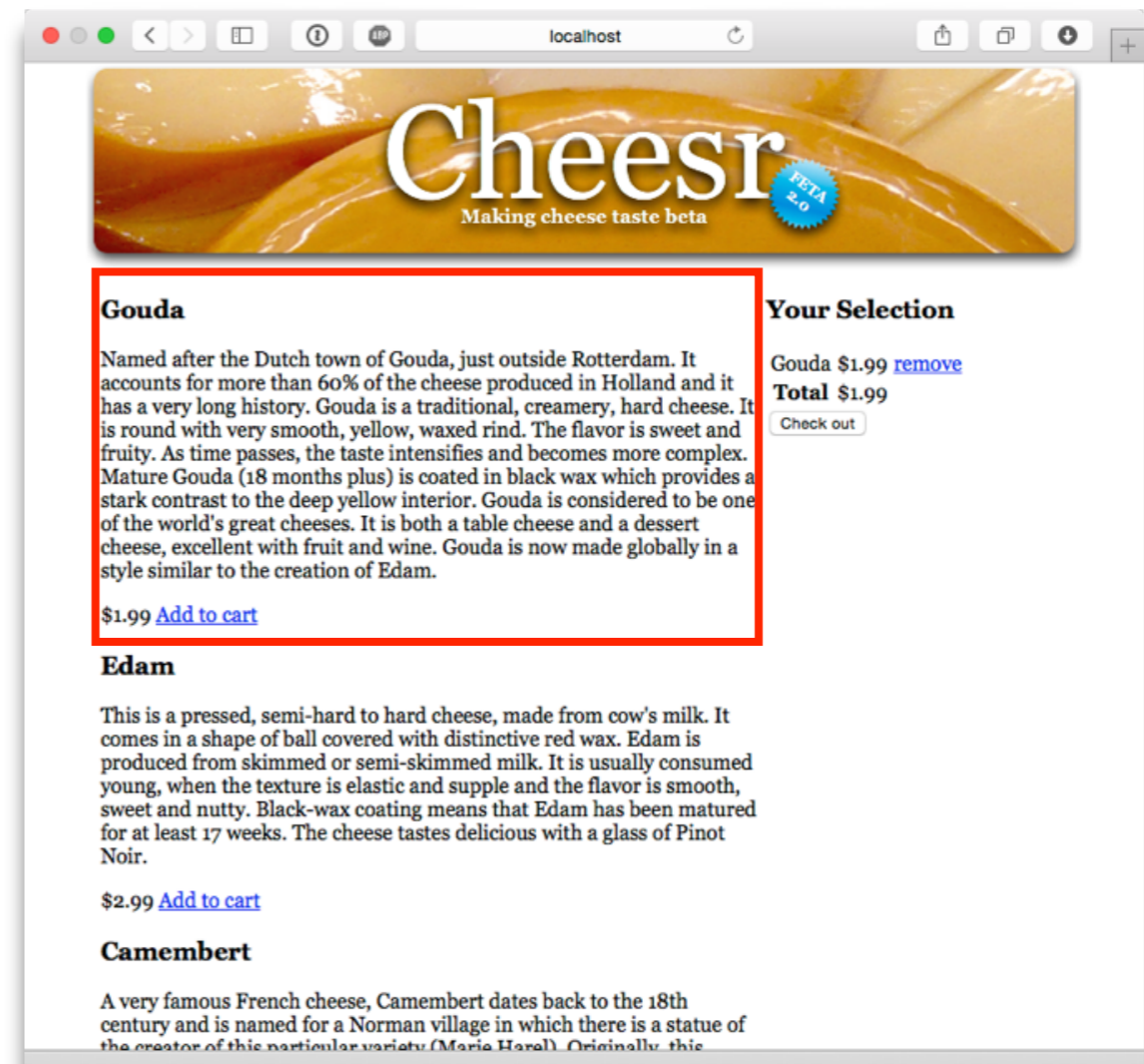
```
<div class="cheese">
  <h3>Gouda</h3>
  <p>Named after the Dutch town of Gouda, just ou
  <p>
    <span>$1.99</span>
    <a href="#">Add to cart</a>
  </p>
</div>
```




```

<div wicket:id="cheese" class="cheese">
  <h3 wicket:id="name">Gouda</h3>
  <p wicket:id="description">Named after the Dutch
  <p>
    €<span wicket:id="price">1.99</span>
    <a href="#">Add to cart</a>
  </p>
</div>

```



```
public class Index extends WebPage {  
    public Index() {  
    }  
}
```

```
public class Index extends WebPage {  
    private CheeseDao cheeseDao = new CheeseDao();  
  
    public Index() {  
    }  
}
```

```
public class Index extends WebPage {  
    private CheeseDao cheeseDao = new CheeseDao();  
  
    public Index() {  
        List<Cheese> cheeses = cheeseDao.getCheeses();  
    }  
}
```

```
public class Index extends WebPage {  
    private CheeseDao cheeseDao = new CheeseDao();  
  
    public Index() {  
        List<Cheese> cheeses = cheeseDao.getCheeses();  
  
        add(new ListView<Cheese>("cheese", cheeses) {  
        });  
    }  
}
```

```
public class Index extends WebPage {
    private CheeseDao cheeseDao = new CheeseDao();

    public Index() {
        List<Cheese> cheeses = cheeseDao.getCheeses();

        add(new ListView<Cheese>("cheese", cheeses) {
            @Override
            protected void populateItem(ListItem<Cheese> item) {
            }
        });
    }
}
```

```
public class Index extends WebPage {
    private CheeseDao cheeseDao = new CheeseDao();

    public Index() {
        List<Cheese> cheeses = cheeseDao.getCheeses();

        add(new ListView<Cheese>("cheese", cheeses) {
            @Override
            protected void populateItem(ListItem<Cheese> item) {
                Cheese c = item.getModelObject();
                item.add(new Label("name", c.getName()));
                item.add(new Label("description", c.getDescription()));
                item.add(new Label("price", c.getPrice()));
            }
        });
    }
}
```

Package Explorer

- cheesr
 - src/main/java
 - com.cheesr
 - dao

```
14     <p wicket:id="description">Named after the Dutch town of Gouda, just outside Rotterdam. It ac
15     <p>
16         €<span wicket:id="price">1.99</span>
17         <a href="#">Add to cart</a>
18     </p>
19 </div>
20 </div>
```

Cheesr
Making cheese taste beta

Gouda
Named after the Dutch town of Gouda, just outside Rotterdam. It accounts for more than 60% of the cheese produced in Holland and it has a very long history. Gouda is a traditional, creamery, hard cheese. It is round with very smooth, yellow, waxed rind. The flavor is sweet and fruity. As time passes, the taste intensifies and becomes more complex. Mature Gouda (18 months plus) is coated in black wax which provides a stark contrast to the deep yellow interior. Gouda is considered to be one of the world's great cheeses. It is both a table cheese and a dessert cheese, excellent with fruit and wine. Gouda is now made globally in a style similar to the creation of Edam.

\$1.99 [Add to cart](#)

Edam
This is a pressed, semi-hard to hard cheese, made from cow's milk. It comes in a shape of ball covered with distinctive red wax. Edam is produced from skimmed or semi-skimmed milk. It is usually consumed young, when the texture is elastic and supple and the flavor is smooth, sweet and nutty. Black-wax coating means that Edam has been matured for at least 17 weeks. The cheese tastes delicious with a glass of Pinot Noir.

\$2.99 [Add to cart](#)

Camembert
A very famous French cheese, Camembert dates back to the 18th century and is named for a Norman village in which there is a statue of the creator of this particular variety (Marie Harel). Originally, this

Your Selection
Gouda \$1.99 [remove](#)
Total \$1.99
[Check out](#)

```
21 <div id="cart">
22 <div>
23 <h3>Your Selection</h3>
24 <table>
25 <tbody>
26 <tr >
27 <td>Gouda</td>
28 <td>$1.99</td>
29 <td><a href="#">remove</a></td>
30 </tr>
31 </tbody>
32 <tfoot>
33 <tr class="total">
34 <th>Total</th>
35 <td>$1.99</td>
36 <td>&nbsp;</td>
37 </tr>
38 </tfoot>
39 </table>
40 </div>
41 <input type="button" value="Check out" />
42 </div>
43 </div>
44 </div>
```

Package Explorer

- WEB-INF
 - web.xml
- test
- target
- pom.xml

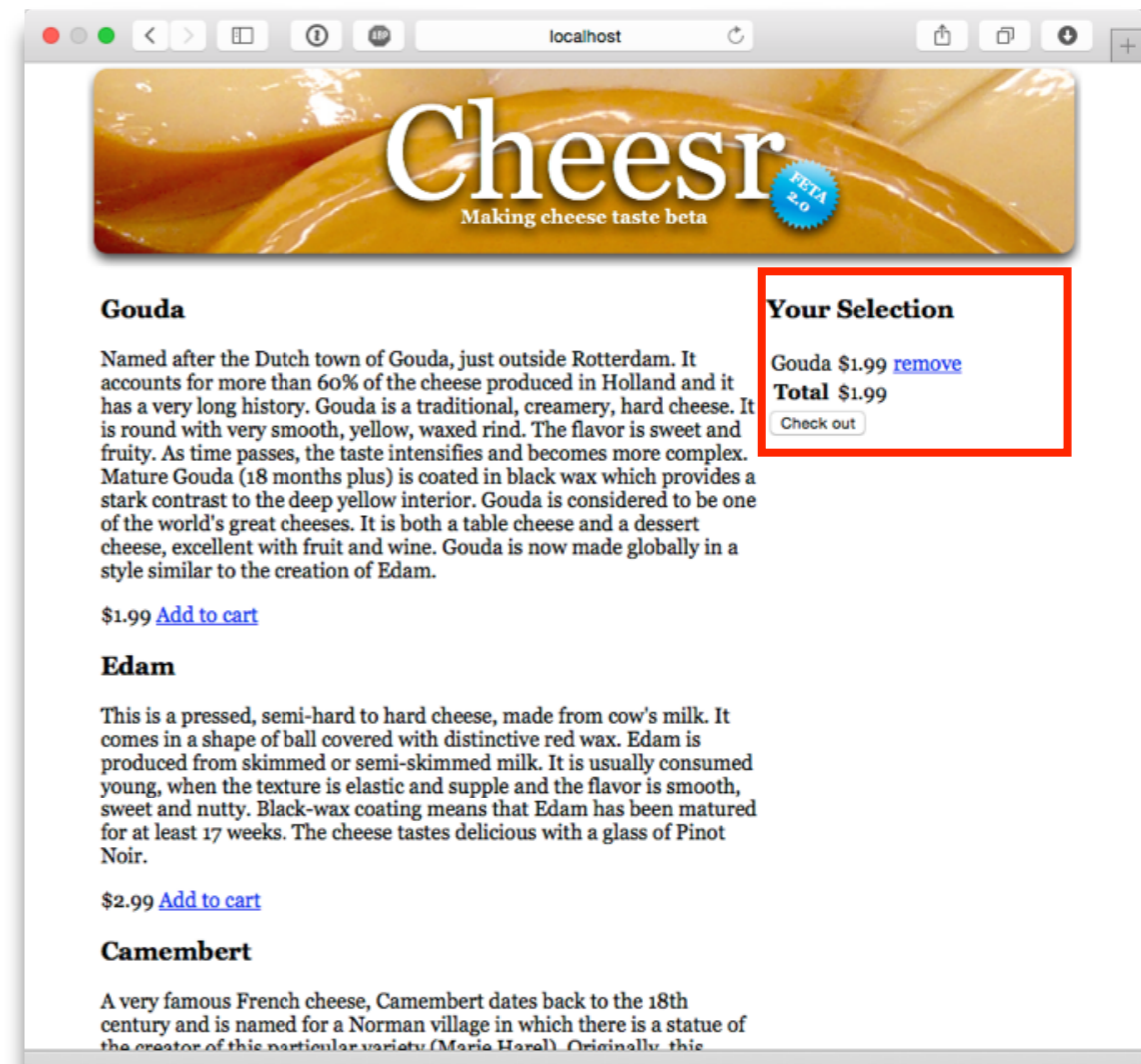
Problems @ **Javadoc** **Declaration** **Servers** **Console**

- WildFly 8.x [Debugging, Republish]
 - cheesr [Started, Republish]
 - XML Configuration
 - Server Details
 - Filesets
 - JMX[Disconnected]


```

<h3>Your Selection</h3>
<table>
<tbody>
  <tr>
    <td>Gouda</td>
    <td>€<span>1.99</span></td>
    <td><a href="#">remove</a></td>
  </tr>
</tbody>
<tfoot>
  <tr class="total">
    <th>Total</th>
    <td>$1.99</td>
    <td>&nbsp;</td>
  </tr>
</tfoot>
</table>

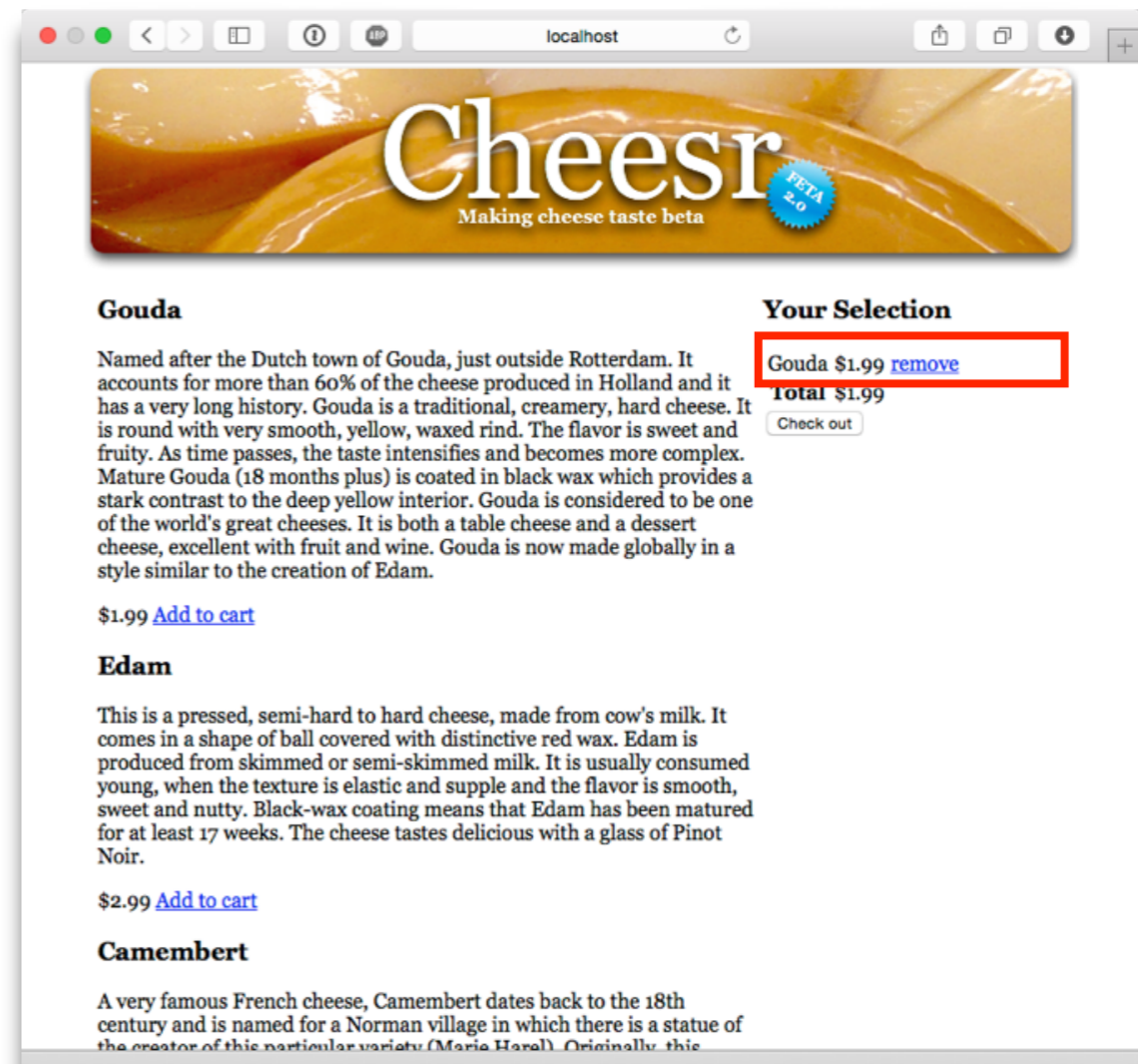
```



```

<h3>Your Selection</h3>
<table>
<tbody>
  <tr wicket:id="item">
    <td wicket:id="name">Gouda</td>
    <td>€<span wicket:id="price">1.99</span></td>
    <td><a href="#">remove</a></td>
  </tr>
</tbody>
<tfoot>
  <tr class="total">
    <th>Total</th>
    <td>$1.99</td>
    <td>&nbsp;</td>
  </tr>
</tfoot>
</table>

```



```
public class Index extends WebPage {  
    private CheeseDao cheeseDao = new CheeseDao();  
  
    public Index() {  
        List<Cheese> cheeses = cheeseDao.getCheeses();  
  
        add(new ListView<Cheese>("cheese", cheeses) {...});  
    }  
}
```

```
public class Index extends WebPage {  
    private CheeseDao cheeseDao = new CheeseDao();  
    private Cart cart = new Cart();  
  
    public Index() {  
        List<Cheese> cheeses = cheeseDao.getCheeses();  
  
        add(new ListView<Cheese>("cheese", cheeses) {...});  
    }  
}
```

```
public class Index extends WebPage {  
    private CheeseDao cheeseDao = new CheeseDao();  
    private Cart cart = new Cart();  
  
    public Index() {  
        List<Cheese> cheeses = cheeseDao.getCheeses();  
  
        add(new ListView<Cheese>("cheese", cheeses) {...});  
  
        add(new ListView<Cheese>("item", cart.getItems()) {  
            });  
        }  
    }  
}
```

```
public class Index extends WebPage {
    private CheeseDao cheeseDao = new CheeseDao();
    private Cart cart = new Cart();

    public Index() {
        List<Cheese> cheeses = cheeseDao.getCheeses();

        add(new ListView<Cheese>("cheese", cheeses) {...});

        add(new ListView<Cheese>("item", cart.getItems()) {
            @Override
            protected void populateItem(ListItem<Cheese> item) {
            }
        });
    }
}
```

```
public class Index extends WebPage {
    private CheeseDao cheeseDao = new CheeseDao();
    private Cart cart = new Cart();

    public Index() {
        List<Cheese> cheeses = cheeseDao.getCheeses();

        add(new ListView<Cheese>("cheese", cheeses) {...});

        add(new ListView<Cheese>("item", cart.getItems()) {
            @Override
            protected void populateItem(ListItem<Cheese> item) {
                Cheese c = item.getModelObject();
            }
        });
    }
}
```

```
public class Index extends WebPage {
    private CheeseDao cheeseDao = new CheeseDao();
    private Cart cart = new Cart();

    public Index() {
        List<Cheese> cheeses = cheeseDao.getCheeses();

        add(new ListView<Cheese>("cheese", cheeses) {...});

        add(new ListView<Cheese>("item", cart.getItems()) {
            @Override
            protected void populateItem(ListItem<Cheese> item) {
                Cheese c = item.getModelObject();
                item.add(new Label("name", c.getName()));
                item.add(new Label("price", c.getPrice()));
            }
        });
    }
}
```




Package Explorer

- cheesr
 - src/main/java
 - com.cheesr
 - dao

```
Index.html | Index.java
2 <html>
3 <head>
4 <link href="style/style.css" rel="stylesheet" media="screen" />
5 <title>Cheesr - Making cheese taste beta</title>
6 </head>
7 <body>
8 <div id="container">
9 <div id="header"><h1>Cheesr</h1</div>
10 <div id="content">
11 <div id="main">
12 <div wicket:id="cheese" class="cheese">
13 <h3 wicket:id="name">Gouda</h3>
14 <p wicket:id="description">Named after the Dutch town of Gouda, just outside Rotterdam. It ac
15 <p>
16 <span wicket:id="price">1.99</span>
17 <a href="#">Add to cart</a>
18 </p>
19 </div>
20 </div>
21 </div>
22 <div id="cart">
23 <div>
24 <h3>Your Selection</h3>
25 <table>
26 <tbody>
27 <tr wicket:id="item">
28 <td wicket:id="name">Gouda</td>
29 <td wicket:id="price">$1.99</td>
30 <td><a href="#">remove</a></td>
31 </tr>
32 </tbody>
33 </table>
34 </div>
35 <div class="total">
36 <th>Total</th>
37 <td>$1.99</td>
38 </div>
39 </div>
40 </div>
41 </div>
42 </div>
43 </div>
44 </div>
45 </div>
46 </div>
47 </div>
48 </div>
49 </div>
50 </div>
51 </div>
52 </div>
53 </div>
54 </div>
55 </div>
56 </div>
57 </div>
58 </div>
59 </div>
60 </div>
61 </div>
62 </div>
63 </div>
64 </div>
65 </div>
66 </div>
67 </div>
68 </div>
69 </div>
70 </div>
71 </div>
72 </div>
73 </div>
74 </div>
75 </div>
76 </div>
77 </div>
78 </div>
79 </div>
80 </div>
81 </div>
82 </div>
83 </div>
84 </div>
85 </div>
86 </div>
87 </div>
88 </div>
89 </div>
90 </div>
91 </div>
92 </div>
93 </div>
94 </div>
95 </div>
96 </div>
97 </div>
98 </div>
99 </div>
100 </div>
```

localhost

Gouda

Named after the Dutch town of Gouda, just outside Rotterdam. It accounts for more than 60% of the cheese produced in Holland and it has a very long history. Gouda is a traditional, creamery, hard cheese. It is round with very smooth, yellow, waxed rind. The flavor is sweet and fruity. As time passes, the taste intensifies and becomes more complex. Mature Gouda (18 months plus) is coated in black wax which provides a stark contrast to the deep yellow interior. Gouda is considered to be one of the world's great cheeses. It is both a table cheese and a dessert cheese, excellent with fruit and wine. Gouda is now made globally in a style similar to the creation of Edam.

\$1.99 [Add to cart](#)

Edam

This is a pressed, semi-hard to hard cheese, made from cow's milk. It comes in a shape of ball covered with distinctive red wax. Edam is produced from skimmed or semi-skimmed milk. It is usually consumed young, when the texture is elastic and supple and the flavor is smooth, sweet and nutty. Black-wax coating means that Edam has been matured for at least 17 weeks. The cheese tastes delicious with a glass of Pinot Noir.

\$2.99 [Add to cart](#)

Camembert

A very famous French cheese, Camembert dates back to the 18th century and is named for a Norman village in which there is a statue of the creator of this particular variety (Marie Harel). Originally, this

WEB-INF

- web.xml

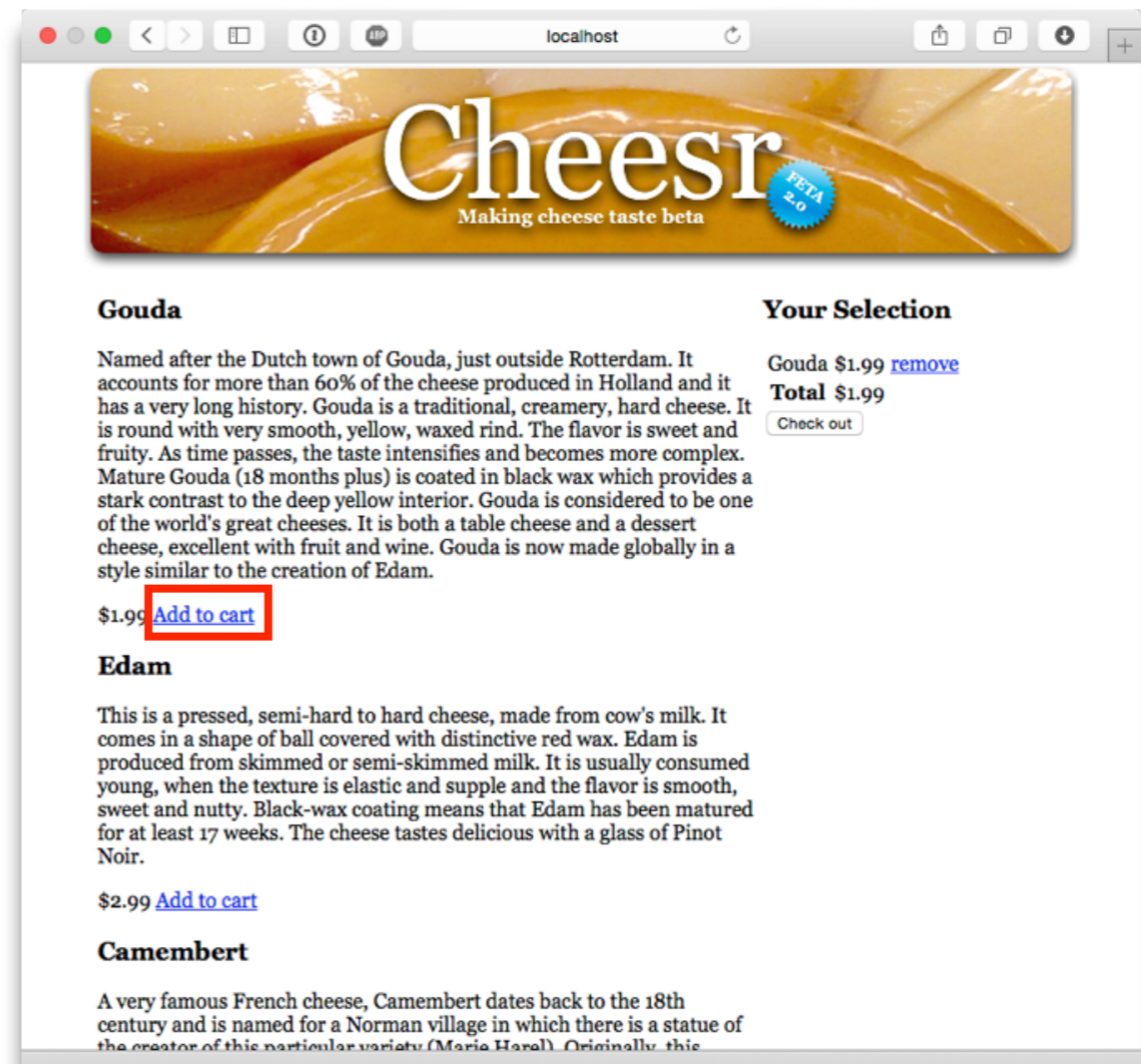
test

- target
- pom.xml

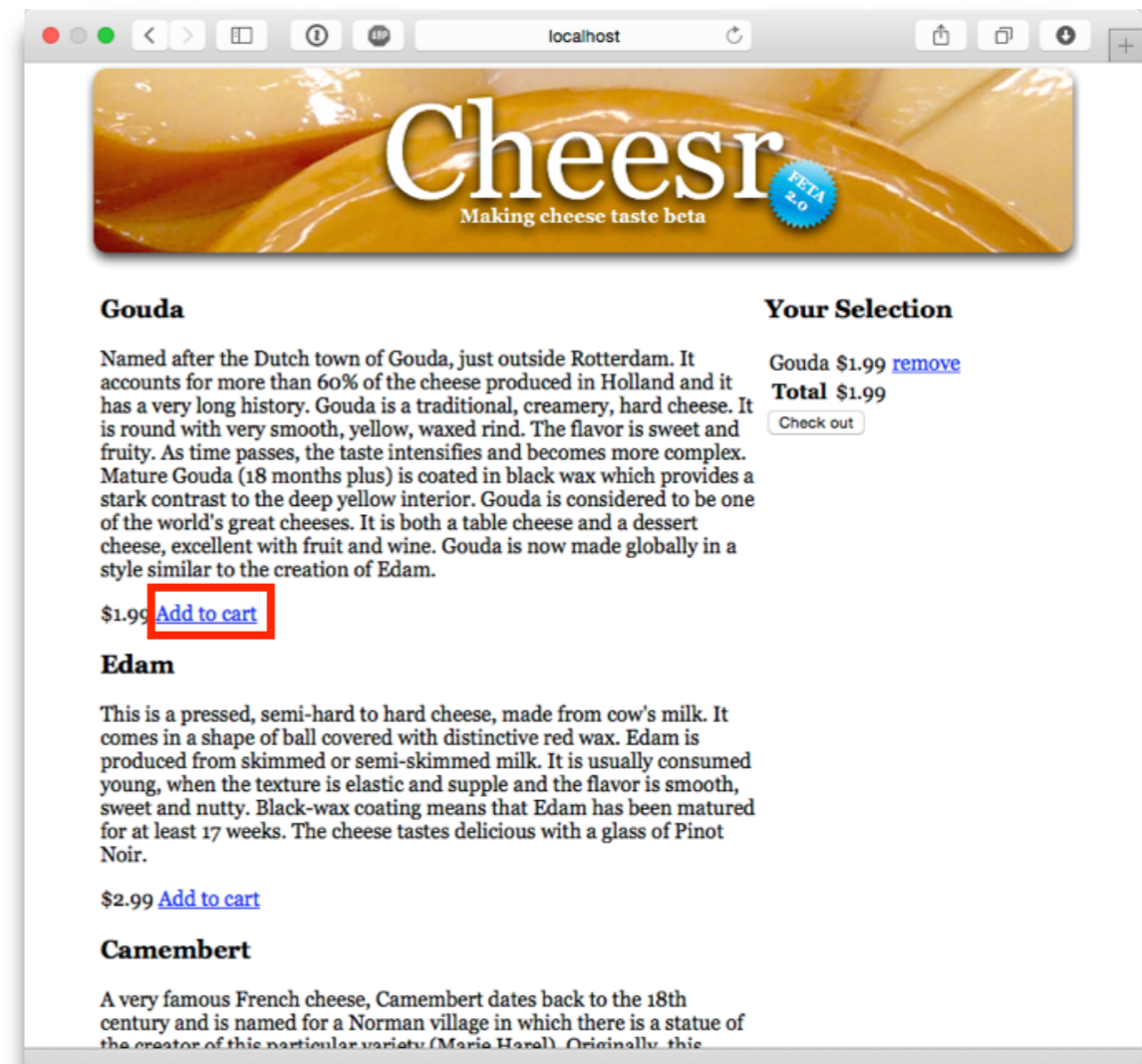
Problems | Javadoc | Declaration | Servers | Console

- WildFly 8.x [Debugging, Republish]
 - cheesr [Started, Republish]
 - XML Configuration
 - Server Details
 - Filesets
 - JMX[Disconnected]

```
<div wicket:id="cheese" class="cheese">
  <h3 wicket:id="name">Gouda</h3>
  <p wicket:id="description">Named after the Dutch
  <p>
    €<span wicket:id="price">1.99</span>
    <a href="#">Add to cart</a>
  </p>
</div>
```



```
<div wicket:id="cheese" class="cheese">
  <h3 wicket:id="name">Gouda</h3>
  <p wicket:id="description">Named after the Dutch
  <p>
    €<span wicket:id="price">1.99</span>
    <a wicket:id="price" href="#">Add to cart</a>
  </p>
</div>
```



```
add(new ListView<Cheese>("cheese", cheeses) {  
    @Override  
    protected void populateItem(ListItem<Cheese> item) {  
        Cheese c = item.getModelObject();  
        item.add(new Label("name", c.getName()));  
        item.add(new Label("description", c.getDescription()));  
        item.add(new Label("price", c.getPrice()));  
    }  
});
```

```
add(new ListView<Cheese>("cheese", cheeses) {
    @Override
    protected void populateItem(ListItem<Cheese> item) {
        Cheese c = item.getModelObject();
        item.add(new Label("name", c.getName()));
        item.add(new Label("description", c.getDescription()));
        item.add(new Label("price", c.getPrice()));
        item.add(new Link<Cheese>("price", item.getModel()));
    }
});
```

```
add(new ListView<Cheese>("cheese", cheeses) {  
    @Override  
    protected void populateItem(ListItem<Cheese> item) {  
        Cheese c = item.getModelObject();  
        item.add(new Label("name", c.getName()));  
        item.add(new Label("description", c.getDescription()));  
        item.add(new Label("price", c.getPrice()));  
        item.add(new Link<Cheese>("price", item.getModel()) {  
        });  
    }  
});
```

```
add(new ListView<Cheese>("cheese", cheeses) {
    @Override
    protected void populateItem(ListItem<Cheese> item) {
        Cheese c = item.getModelObject();
        item.add(new Label("name", c.getName()));
        item.add(new Label("description", c.getDescription()));
        item.add(new Label("price", c.getPrice()));
        item.add(new Link<Cheese>("price", item.getModel()) {
            @Override
            public void onClick() {
            }
        });
    }
});
```

```
add(new ListView<Cheese>("cheese", cheeses) {
    @Override
    protected void populateItem(ListItem<Cheese> item) {
        Cheese c = item.getModelObject();
        item.add(new Label("name", c.getName()));
        item.add(new Label("description", c.getDescription()));
        item.add(new Label("price", c.getPrice()));
        item.add(new Link<Cheese>("price", item.getModel()) {
            @Override
            public void onClick() {
                Cheese cheese = getModelObject();
            }
        });
    }
});
```



```
add(new ListView<Cheese>("cheese", cheeses) {
    @Override
    protected void populateItem(ListItem<Cheese> item) {
        Cheese c = item.getModelObject();
        item.add(new Label("name", c.getName()));
        item.add(new Label("description", c.getDescription()));
        item.add(new Label("price", c.getPrice()));
        item.add(new Link<Cheese>("price", item.getModel()) {
            @Override
            public void onClick() {
                Cheese cheese = getModelObject();
                cart.getItems().add(cheese);
            }
        });
    }
});
```

Injection (EJB, CDI)

Conversations (CDI)

JPA

Bean validation

Testing with Arquillian



Using EJBs



Steps

**Add EJB dependencies
to project**

Make DAOs EJBs

Inject DAOs



```
<dependencies>
  <!-- EJB DEPENDENCIES -->
  <dependency>
    <groupId>org.jboss.spec.javax.ejb</groupId>
    <artifactId>jboss-ejb-api_3.2_spec</artifactId>
    <scope>provided</scope>
  </dependency>
  ...
</dependencies>
```

Steps

Add EJB dependencies
to project

Make DAOs EJBs

Inject DAOs



```
public class CheeseDao {  
    public List<Cheese> getCheeses() {...}  
}  
  
public class OrderDao {  
    public void save(Order order) {...}  
    public List<Order> list() {...}  
}  
  
public class Index extends WebPage {  
    private CheeseDao cheeseDao = new CheeseDao();  
  
    public Index() {...}  
}
```

```
public class CheeseDao {  
    public List<Cheese> getCheeses() {...}  
}
```

```
public class OrderDao {  
    public void save(Order order) {...}  
    public List<Order> list() {...}  
}
```

```
public class Index extends WebPage {  
    private CheeseDao cheeseDao = new CheeseDao();  
  
    public Index() {...}  
}
```



```
@Stateless
public class CheeseDao {
    public List<Cheese> getCheeses() {...}
}

public class OrderDao {
    public void save(Order order) {...}

    public List<Order> list() {...}
}

public class Index extends WebPage {
    private CheeseDao cheeseDao = new CheeseDao();

    public Index() {...}
}
```

```
@Stateless
public class CheeseDao {
    public List<Cheese> getCheeses() {...}
}
```

```
@Stateless
public class OrderDao {
    public void save(Order order) {...}

    public List<Order> list() {...}
}
```

```
public class Index extends WebPage {
    private CheeseDao cheeseDao = new CheeseDao();

    public Index() {...}
}
```

```
@Stateless  
public class CheeseDao {  
    public List<Cheese> getCheeses() {...}  
}
```

```
@Stateless  
public class OrderDao {  
    public void save(Order order) {...}  
  
    public List<Order> list() {...}  
}
```

```
public class Index extends WebPage {  
    private CheeseDao cheeseDao;  
  
    public Index() {...}  
}
```

Steps

Add EJB dependencies
to project

Make DAOs EJBs

Inject DAOs



```
@Stateless
public class CheeseDao {
    public List<Cheese> getCheeses() {...}
}
```

```
@Stateless
public class OrderDao {
    public void save(Order order) {...}

    public List<Order> list() {...}
}
```

```
public class Index extends WebPage {
    @EJB
    private CheeseDao cheeseDao;

    public Index() {...}
}
```

21:20:55,305 Processing weld deployment cheesr.war
21:20:55,313 JNDI bindings for session bean named CheeseDao
in deployment unit deployment "cheesr.war" are as follows:

```
java:global/cheesr/CheeseDao!com.cheesr.dao.CheeseDao  
java:app/cheesr/CheeseDao!com.cheesr.dao.CheeseDao  
java:module/CheeseDao!com.cheesr.dao.CheeseDao  
java:global/cheesr/CheeseDao  
java:app/cheesr/CheeseDao  
java:module/CheeseDao
```

21:20:55,335 Starting Services for CDI deployment:
cheesr.war
21:20:55,340 Starting weld service for deployment
cheesr.war

Unexpected RuntimeException

Last cause: null
WicketMessage: Can't instantiate page using constructor 'public com.cheesr.web.Index()'. An exception has been

Stacktrace

Root cause:

```
java.lang.NullPointerException
  at com.cheesr.web.Index.<init>(Index.java:27)
  at sun.reflect.GeneratedConstructorAccessor684.newInstance(Unknown Source)
  at java.lang.reflect.Constructor.newInstance(Constructor.java:408)
  at org.apache.wicket.session.DefaultPageFactory.newPage(DefaultPageFactory.java:175)
  at org.apache.wicket.session.DefaultPageFactory.newPage(DefaultPageFactory.java:67)
  at org.apache.wicket.DefaultMapperContext.newPageInstance(DefaultMapperContext.java:133)
  at org.apache.wicket.core.request.handler.PageProvider.resolvePageInstance(PageProvider.java:268)
  at org.apache.wicket.core.request.handler.PageProvider.getPageInstance(PageProvider.java:166)
  at org.apache.wicket.request.handler.render.PageRenderer.getPage(PageRenderer.java:78)
  at org.apache.wicket.request.handler.render.WebPageRenderer.respond(WebPageRenderer.java:279)
  at org.apache.wicket.core.request.handler.RenderPageRequestHandler.respond(RenderPageRequestHandler.java:114)
  at org.apache.wicket.request.cycle.RequestCycle$HandlerExecutor.respond(RequestCycle.java:862)
  at org.apache.wicket.request.RequestHandlerStack.execute(RequestHandlerStack.java:64)
  at org.apache.wicket.request.cycle.RequestCycle.execute(RequestCycle.java:261)
  at org.apache.wicket.request.cycle.RequestCycle.processRequest(RequestCycle.java:218)
  at org.apache.wicket.request.cycle.RequestCycle.processRequestAndDetach(RequestCycle.java:289)
  at org.apache.wicket.protocol.http.WicketFilter.processRequestCycle(WicketFilter.java:259)
  at org.apache.wicket.protocol.http.WicketFilter.processRequest(WicketFilter.java:201)
  at org.apache.wicket.protocol.http.WicketFilter.doFilter(WicketFilter.java:282)
  at io.undertow.servlet.core.ManagedFilter.doFilter(ManagedFilter.java:60)
  at io.undertow.servlet.handlers.FilterHandler$FilterChainImpl.doFilter(FilterHandler.java:132)
  at io.undertow.servlet.handlers.FilterHandler.handleRequest(FilterHandler.java:85)
  at io.undertow.servlet.handlers.security.ServletSecurityRoleHandler.handleRequest(ServletSecurityRoleHandler.java:42)
  at io.undertow.servlet.handlers.ServletDispatchingHandler.handleRequest(ServletDispatchingHandler.java:36)
  at org.wildfly.extension.undertow.security.SecurityContextAssociationHandler.handleRequest(SecurityContextAssociationHandler.java:53)
  at io.undertow.server.handlers.PredicateHandler.handleRequest(PredicateHandler.java:25)
  at io.undertow.servlet.handlers.security.SSLInformationAssociationHandler.handleRequest(SSLInformationAssociationHandler.java:43)
```

```
public class Index extends WebPage
{
    @EJB
    private CheeseDao cheeseDao;

    private Cart cart = new Cart();

    public Index()
    {
        List<Cheese> cheeses = cheeseDao.getCheeses();

        add(new ListView<Cheese>("cheese", cheeses) {...});
        add(new ListView<Cheese>("item", cart.getItems()) {...})
    }
}
```



```
public class Index extends WebPage
{
    @EJB
    private CheeseDao cheeseDao;

    private Cart cart = new Cart();

    public Index()
    {
        List<Cheese> cheeses = cheeseDao.getCheeses();

        add(new ListView<Cheese>("cheese", cheeses) {...});
        add(new ListView<Cheese>("item", cart.getItems()) {...})
    }
}
```



Wicket is unmanaged, container doesn't know about Pages, Components, etc.

This repository Search Explore Gist Blog Help

wicketstuff / core Watch 84

Java EE Inject

Michael O'Cleirigh edited this page on Apr 12, 2011 · 1 revision

JavaEE Inject is a wicket module that provides integration through Java EE 5 resource injection. With JavaEE Inject you can use in your wicket components three annotations:

- @EJB
- @PersistenceUnit
- @Resource

Documentation

- [How to use EJB annotation](#)
- [How to use PersistenceUnit annotation](#)
- [How to use Resource annotation](#)

The old example projects with Wicket 1.3-compatible source:

- [A sample application demonstrating the use of @EJB and @Resource](#)
- [A sample application demonstrating the use of @PersistenceUnit and @Resource](#)

The new examples are in the javaee-inject-parent project as javaee-inject-examples module. This example project is always maintained with the current version of Wicket

Pages 83

Find a Page...

- Annotation
- Artwork
- Async tasks
- Autocomplete
- BrowserId
- ClientAndServ
- Console
- Context
- DataStores
- Datatable Auto
- DataTables
- DDCalendar
- Editable Grid

```
<dependencies>
  <!-- EJB DEPENDENCIES -->
  <dependency>
    <groupId>org.jboss.spec.jboss.ejb</groupId>
    <artifactId>jboss-ejb-api_3.2_spec</artifactId>
    <scope>provided</scope>
  </dependency>
  <dependency>
    <groupId>org.wicketstuff</groupId>
    <artifactId>wicketstuff-javaee-inject</artifactId>
    <version>6.17.0</version>
  </dependency>
</dependencies>
```

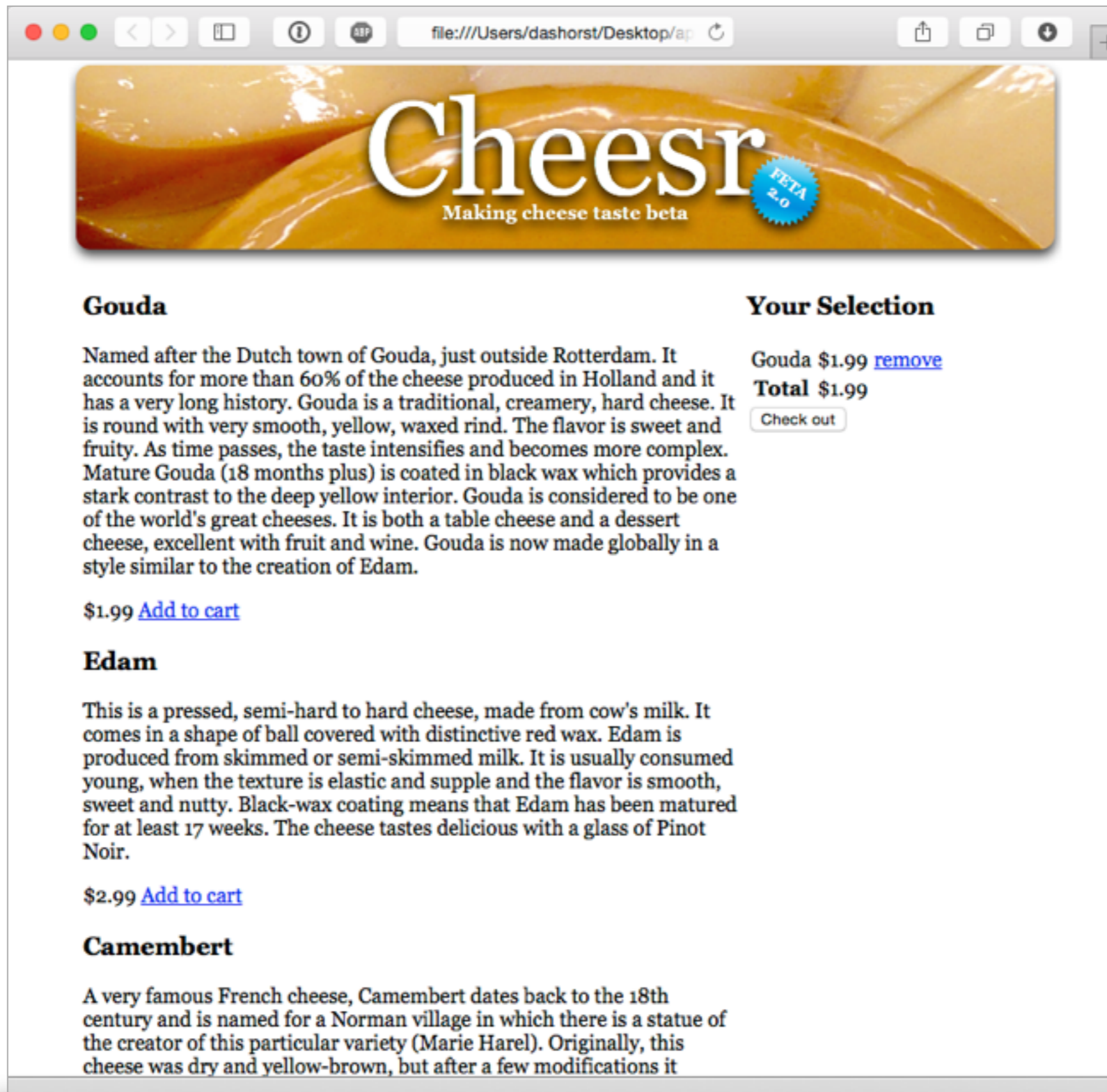
```
public class WicketApplication extends WebApplication
{
    @Override
    public Class< ? extends WebPage> getHomePage()
    {
        return Index.class;
    }

    @Override
    public void init()
    {
        super.init();
    }
}
```

```
public class WicketApplication extends WebApplication
{
    @Override
    public Class< ? extends WebPage> getHomePage()
    {
        return Index.class;
    }

    @Override
    public void init()
    {
        super.init();

        getComponentInstantiationListeners()
            .add(new JavaEEComponentInjector(this));
    }
}
```



Cheesr

Making cheese taste beta



Gouda

Named after the Dutch town of Gouda, just outside Rotterdam. It accounts for more than 60% of the cheese produced in Holland and it has a very long history. Gouda is a traditional, creamery, hard cheese. It is round with very smooth, yellow, waxed rind. The flavor is sweet and fruity. As time passes, the taste intensifies and becomes more complex. Mature Gouda (18 months plus) is coated in black wax which provides a stark contrast to the deep yellow interior. Gouda is considered to be one of the world's great cheeses. It is both a table cheese and a dessert cheese, excellent with fruit and wine. Gouda is now made globally in a style similar to the creation of Edam.

\$1.99 [Add to cart](#)

Edam

This is a pressed, semi-hard to hard cheese, made from cow's milk. It comes in a shape of ball covered with distinctive red wax. Edam is produced from skimmed or semi-skimmed milk. It is usually consumed young, when the texture is elastic and supple and the flavor is smooth, sweet and nutty. Black-wax coating means that Edam has been matured for at least 17 weeks. The cheese tastes delicious with a glass of Pinot Noir.

\$2.99 [Add to cart](#)

Camembert

A very famous French cheese, Camembert dates back to the 18th century and is named for a Norman village in which there is a statue of the creator of this particular variety (Marie Harel). Originally, this cheese was dry and yellow-brown, but after a few modifications it

Your Selection

Gouda \$1.99 [remove](#)

Total \$1.99

[Check out](#)

- add **EJB** dependency and **wicketstuff-javaee-inject** dependency
- make DAOs EJBs
- configure component instantiation listener
- now you can use **@EJB** injection

Using CDI for injection



```
<dependencies>
  <!-- CDI DEPENDENCIES -->
  <dependency>
    <groupId>javax.enterprise</groupId>
    <artifactId>cdi-api</artifactId>
  </dependency>
  <dependency>
    <groupId>org.apache.wicket</groupId>
    <artifactId>wicket-cdi-1.1</artifactId>
    <version>6.18.0</version>
  </dependency>
</dependencies>
```

```
public class WicketApplication extends WebApplication
{
    @Override
    public void init()
    {
        super.init();
    }
}
```

```
public class WicketApplication extends WebApplication
{
    @Override
    public void init()
    {
        super.init();

        CdiConfiguration cdiConfiguration =
            new CdiConfiguration();
        cdiConfiguration.configure(this);
    }
}
```

```
@ApplicationScoped
public class MessageOfTheDay
{
    public String getMessage()
    {
        return "What happens to the hole when the cheese is go
    }
}
```

```
public class Index extends WebPage {
    @EJB
    private CheeseDao cheeseDao;

    public Index()
    {
        List<Cheese> cheeses = cheeseDao.getCheeses();

        add(new ListView<Cheese>("cheese", cheeses) {...});
        add(new ListView<Cheese>("item", cart.getItems()) {...})
    }
}
```

```
public class Index extends WebPage {
    @EJB
    private CheeseDao cheeseDao;

    @Inject
    private MessageOfTheDay message;

    public Index() {
        List<Cheese> cheeses = cheeseDao.getCheeses();

        add(new ListView<Cheese>("cheese", cheeses) {...});
        add(new ListView<Cheese>("item", cart.getItems()) {...})
    }
}
```

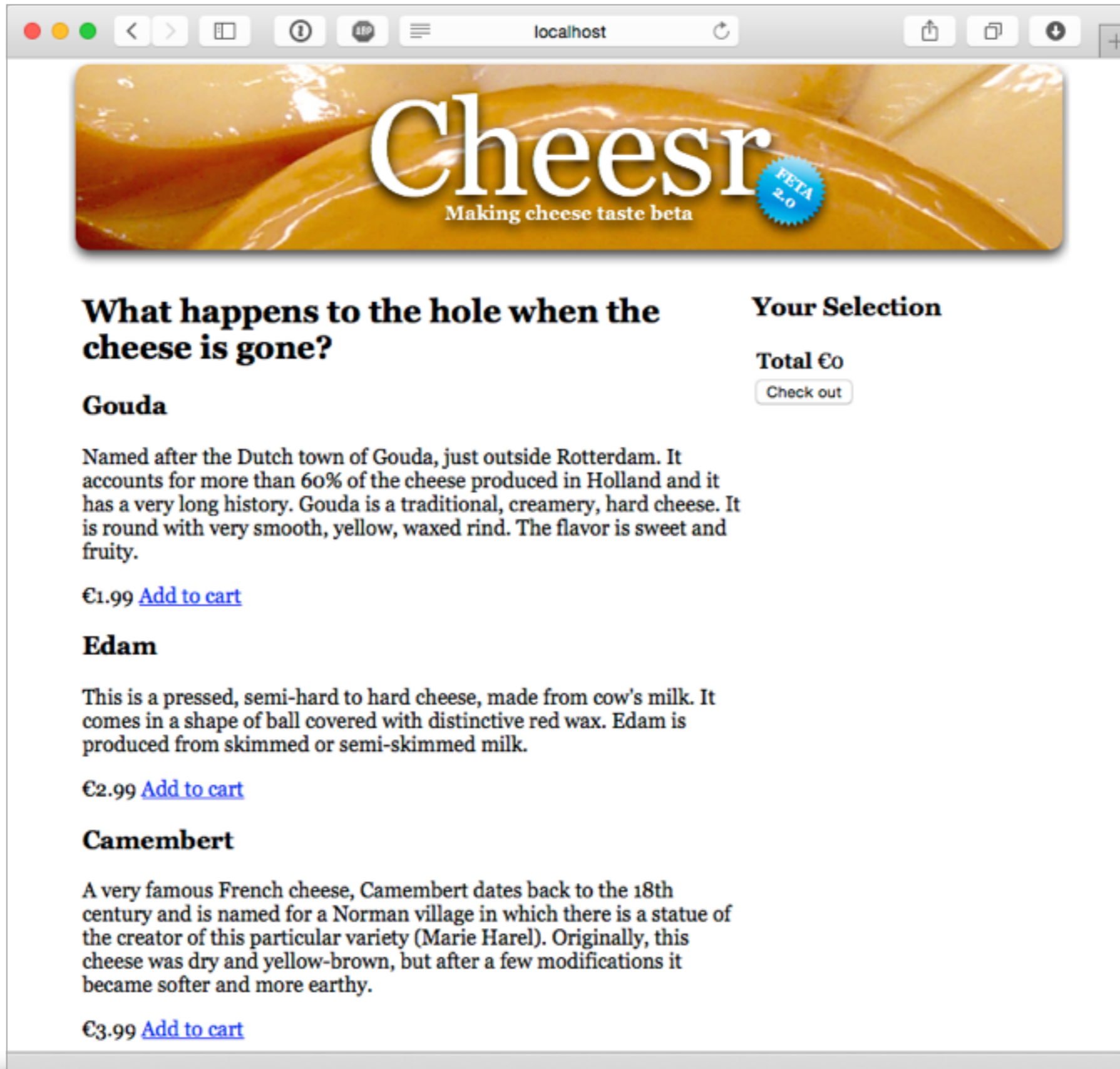
```
public class Index extends WebPage {
    @EJB
    private CheeseDao cheeseDao;

    @Inject
    private MessageOfTheDay message;

    public Index() {
        add(new Label("message", message.getMessage()));

        List<Cheese> cheeses = cheeseDao.getCheeses();

        add(new ListView<Cheese>("cheese", cheeses) {...});
        add(new ListView<Cheese>("item", cart.getItems()) {...})
    }
}
```

Cheesr

Making cheese taste beta

FETA 2.0

What happens to the hole when the cheese is gone?

Gouda

Named after the Dutch town of Gouda, just outside Rotterdam. It accounts for more than 60% of the cheese produced in Holland and it has a very long history. Gouda is a traditional, creamery, hard cheese. It is round with very smooth, yellow, waxed rind. The flavor is sweet and fruity.

€1.99 [Add to cart](#)

Edam

This is a pressed, semi-hard to hard cheese, made from cow's milk. It comes in a shape of ball covered with distinctive red wax. Edam is produced from skimmed or semi-skimmed milk.

€2.99 [Add to cart](#)

Camembert

A very famous French cheese, Camembert dates back to the 18th century and is named for a Norman village in which there is a statue of the creator of this particular variety (Marie Harel). Originally, this cheese was dry and yellow-brown, but after a few modifications it became softer and more earthy.

€3.99 [Add to cart](#)

Your Selection

Total €0

[Check out](#)

Injecting in non-managed objects



```
public class MessageModel
    extends LoadableDetachableModel<String> {
    @Inject
    private MessageOfTheDay messages;

    @Override
    protected String load() {
        return messages.getMessage();
    }
}
```

Caused by: [java.lang.NullPointerException](#)

at com.cheesr.web.MessageModel.load([MessageModel.java:15](#)) [:]

at com.cheesr.web.MessageModel.load([MessageModel.java:1](#)) [:]

at org.apache.wicket.model.LoadableDetachableModel.getObject(

at org.apache.wicket.Component.getDefaultModelObject([Componen](#)

... 55 more

```
public class MessageModel
    extends LoadableDetachableModel<String> {
@Inject
private MessageOfTheDay messages;

@Override
protected String load() {
    return messages.getMessage();
}
}
```

```
public class MessageModel
    extends LoadableDetachableModel<String> {
    @Inject
    private MessageOfTheDay messages;

    public MessageModel() {
        NonContextual.of(MessageModel.class).inject(this);
    }

    @Override
    protected String load() {
        return messages.getMessage();
    }
}
```

- Add **wicket-cdi-1.1** and **javax.cdi-api**
- Configure Wicket's **CdiConfiguration**
- Use **@Inject** injection
- Use **NonContextual** in non-managed objects

Injection (EJB, CDI)

Conversations (CDI)

JPA

Bean validation

Testing with Arquillian



Next steps

4. Navigate to Checkout

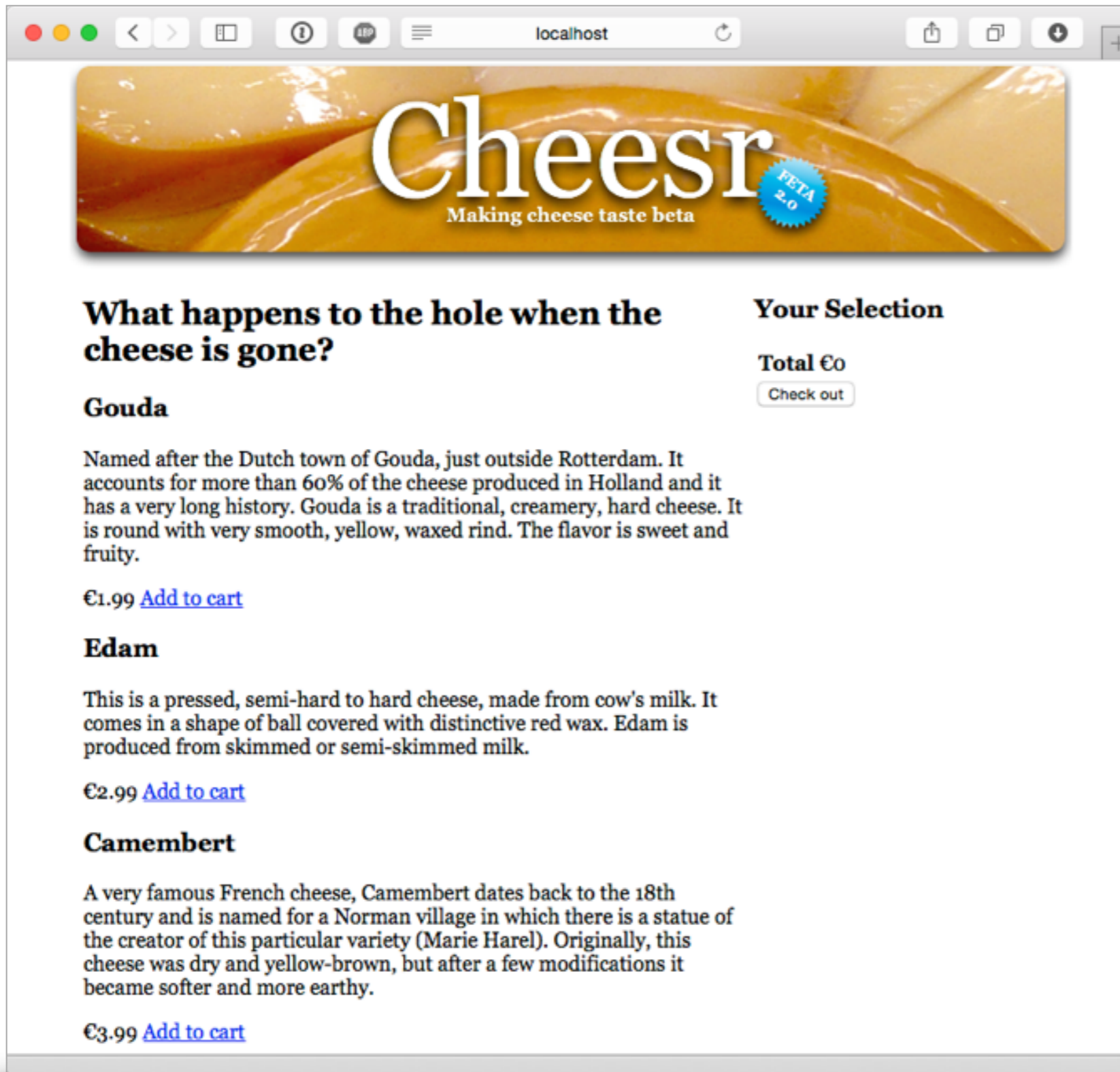
5. Implement Checkout

6. Make Cart Conversation Scoped



4. navigate to checkout





Cheesr

Making cheese taste beta

FETA
2.0

What happens to the hole when the cheese is gone?

Gouda

Named after the Dutch town of Gouda, just outside Rotterdam. It accounts for more than 60% of the cheese produced in Holland and it has a very long history. Gouda is a traditional, creamery, hard cheese. It is round with very smooth, yellow, waxed rind. The flavor is sweet and fruity.

€1.99 [Add to cart](#)

Edam

This is a pressed, semi-hard to hard cheese, made from cow's milk. It comes in a shape of ball covered with distinctive red wax. Edam is produced from skimmed or semi-skimmed milk.

€2.99 [Add to cart](#)

Camembert

A very famous French cheese, Camembert dates back to the 18th century and is named for a Norman village in which there is a statue of the creator of this particular variety (Marie Harel). Originally, this cheese was dry and yellow-brown, but after a few modifications it became softer and more earthy.


€3.99 [Add to cart](#)

Your Selection

Total €0

[Check out](#)

localhost



Cheesr

Making cheese taste beta

What happens to the hole when the cheese is gone?

Gouda

Named after the Dutch town of Gouda, just outside Rotterdam. It accounts for more than 60% of the cheese produced in Holland and it has a very long history. Gouda is a traditional, creamery, hard cheese. It is round with very smooth, yellow, waxed rind. The flavor is sweet and fruity.

€1.99 [Add to cart](#)

Edam

This is a pressed, semi-hard to hard cheese, made from cow's milk. It comes in a shape of ball covered with distinctive red wax. Edam is produced from skimmed or semi-skimmed milk.

€2.99 [Add to cart](#)

Camembert

A very famous French cheese, Camembert dates back to the 18th century and is named for a Norman village in which there is a statue of the creator of this particular variety (Marie Harel). Originally, this cheese was dry and yellow-brown, but after a few modifications it became softer and more earthy.


€3.99 [Add to cart](#)

Your Selection

Total €0

[Check out](#)

localhost



Cheesr

Making cheese taste beta

- Name is required

Check out

Please enter your billing address.

Name

Street

Zipcode

City

Country

Your Selection

Gouda \$1.99 [remove](#)

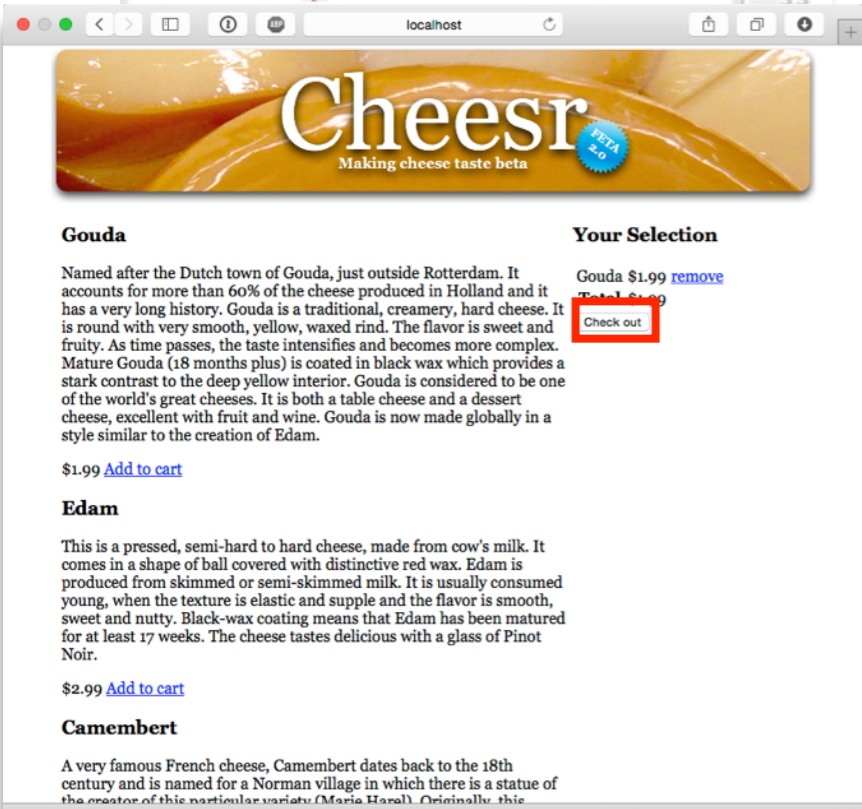
Total \$1.99



Package Explorer | Index.html | Index.java

```
Package Explorer
└─ cheesr
   └─ src/main/java
      └─ com.cheesr
         └─ dao
```

```
27      Edam.</p>
28      <p>
29          €<span wicket:id="price">1.99</span>
30          <a wicket:id="add" href="#">Add to cart</a>
31      </p>
32  </div>
33  </div>
</div>
<div id="cart">
  <div>
    <h3>Your Selection</h3>
    <table>
      <tbody>
        <tr wicket:id="item">
          <td wicket:id="name">Gouda</td>
          <td wicket:id="price">$1.99</td>
          <td><a wicket:id="remove" href="#">remove</a></td>
        </tr>
      </tbody>
    </table>
    <tfoot>
      <tr class="total">
        <th>Total</th>
        <td>$1.99</td>
        <td>&nbsp;</td>
      </tr>
    </tfoot>
  </table>
  <div>
    <input type="button" value="Check out" />
  </div>
</div>
</div>
</body>
</html>
60
```

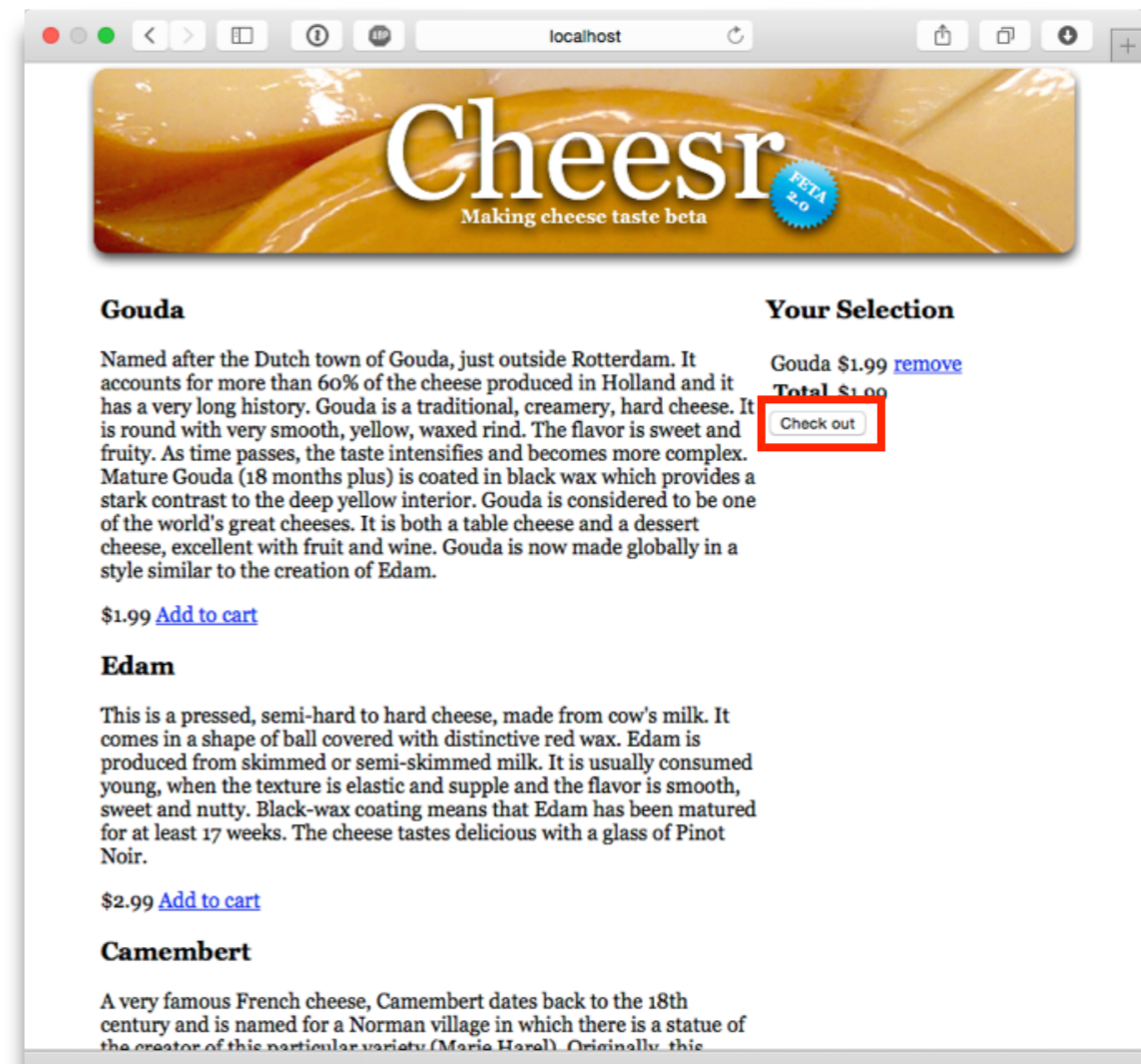


```
Package Explorer
└─ WEB-INF
   └─ web.xml
└─ test
└─ target
   └─ pom.xml
```

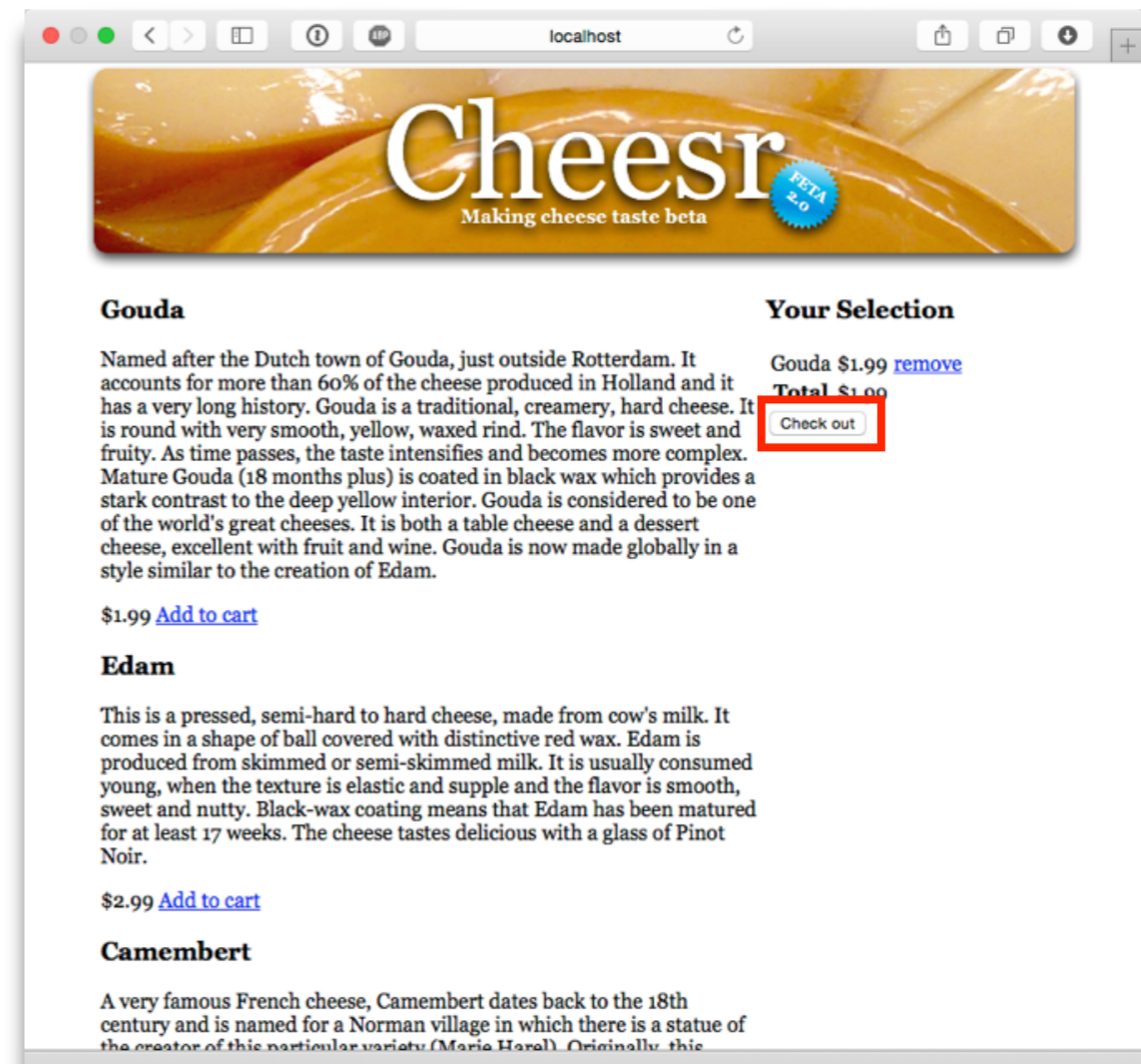
Problems | Javadoc | Declaration | Servers | Console

```
WildFly 8.x [Debugging, Republish]
└─ cheesr [Started, Republish]
   └─ XML Configuration
   └─ Server Details
   └─ Filesets
   └─ JMX[Disconnected]
```

```
<div id="cart">
  <h3>Your Selection</h3>
  <table>
    <tbody>...</tbody>
    <tfoot>...</tfoot>
  </table>
  <input type="button" value="Check out" />
</div>
```



```
<div id="cart">
  <h3>Your Selection</h3>
  <table>
    <tbody>...</tbody>
    <tfoot>...</tfoot>
  </table>
  <input wicket:id="checkout" type="button" value="Check ou
</div>
```



```
public class Index extends WebPage {
    private CheeseDao cheeseDao = new CheeseDao();
    private Cart cart = new Cart();

    public Index() {
        List<Cheese> cheeses = cheeseDao.getCheeses();

        add(new ListView<Cheese>("cheese", cheeses) {...});
        add(new ListView<Cheese>("item", cart.getItems()) {...})
    }
}
```



```
public class Index extends WebPage {
    private CheeseDao cheeseDao = new CheeseDao();
    private Cart cart = new Cart();

    public Index() {
        List<Cheese> cheeses = cheeseDao.getCheeses();

        add(new ListView<Cheese>("cheese", cheeses) {...});
        add(new ListView<Cheese>("item", cart.getItems()) {...})
        add(new Link<Void>("checkout") {});
    }
}
```

```
public class Index extends WebPage {
    private CheeseDao cheeseDao = new CheeseDao();
    private Cart cart = new Cart();

    public Index() {
        List<Cheese> cheeses = cheeseDao.getCheeses();

        add(new ListView<Cheese>("cheese", cheeses) {...});
        add(new ListView<Cheese>("item", cart.getItems()) {...})
        add(new Link<Void>("checkout") {
            @Override
            public void onClick() {
            }
        });
    }
}
```

```
public class Index extends WebPage {
    private CheeseDao cheeseDao = new CheeseDao();
    private Cart cart = new Cart();

    public Index() {
        List<Cheese> cheeses = cheeseDao.getCheeses();

        add(new ListView<Cheese>("cheese", cheeses) {...});
        add(new ListView<Cheese>("item", cart.getItems()) {...})
        add(new Link<Void>("checkout") {
            @Override
            public void onClick() {
                setResponsePage(new CheckoutPage(cart));
            }
        });
    }
}
```

```
public class CheckoutPage extends WebPage {  
    private Cart cart = new Cart();  
  
    public CheckoutPage(Cart cart) {  
        this.cart = cart;  
    }  
}
```

5. implement checkout



Package Explorer

- cheesr
 - src/main/java
 - com.cheesr
 - dao

CheckoutPage.java

```
6 </head>
7 <body>
8 <div id="container">
9 <div id="header"><h1>Cheesr - checkout</h1></div>
10 <div id="content">
11 <div id="main">
12 <div ><ul><li>Name is required</li></ul></div>
13 <form >
14 <h3>Check out</h3>
15 <p>Please enter your billing address.</p>
16 <table>
17 <tr>
18 <th>Name</th>
19 <td><input type="text" /></td>
20 </tr>
21 <tr>
22 <th>Street</th>
23 <td><input type="text" /></td>
24 </tr>
25 <tr>
26 <th>Zipcode</th>
27 <td><input type="text" /></td>
28 </tr>
29 <tr>
30 <th>City</th>
31 <td><input type="text" /></td>
32 </tr>
33 <tr>
34 <th>Country</th>
35 <td><input type="text" /></td>
36 </tr>
37 <tr>
38 <th>&nbsp;</th>
39 <td>
40 <input type="button" value="Cancel" />
41 <input type="submit" value="Order!" />
42 </td>
43 </tr>
44 </table>
45 </form>
46 </div>
47 <div id="cart">
48 <div>
49 <h3>Your Selection</h3>
50 <table>
51 <tbody>
```

CheckoutPage.html

localhost

• Name is required

Check out

Please enter your billing address.

Name

Street

Zipcode

City

Country

Your Selection

Gouda \$1.99 [remove](#)

Total \$1.99

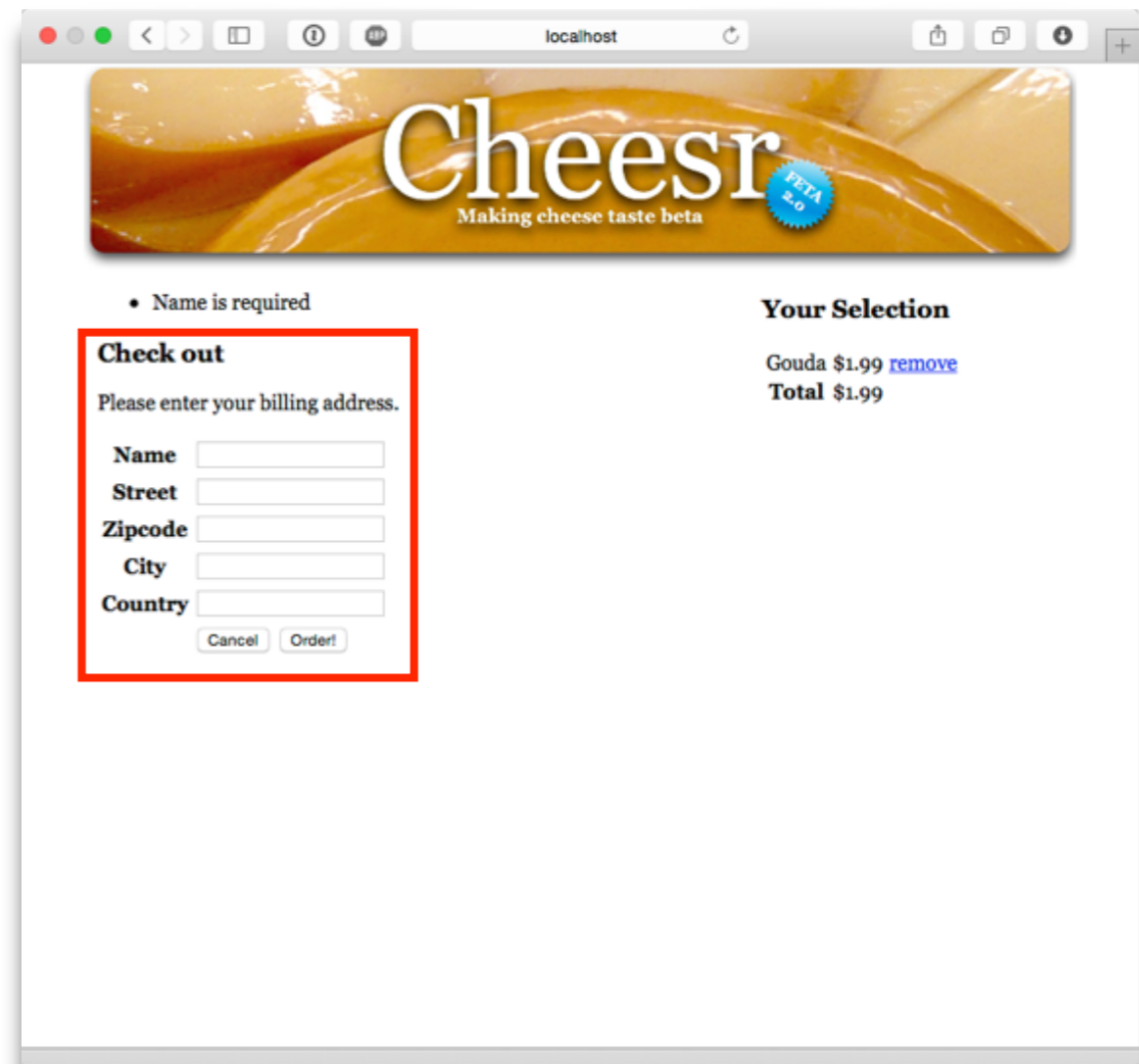
src

- main
- test
- target
 - generated-sources
 - m2e-wtp
- pom.xml

```

<form >
  <h3>Check out</h3>
  <p>Please enter your billing address.</p>
  <table>
    <tr>
      <th>Name</th>
      <td><input type="text" /></td>
    </tr>
    <tr>
      <th>Street</th>
      <td><input type="text" />
    </tr>
    <tr>
      <th>Zipcode</th>
      <td><input type="text" />
    </tr>
    <tr>
      <th>City</th>
      <td><input type="text" />
    </tr>

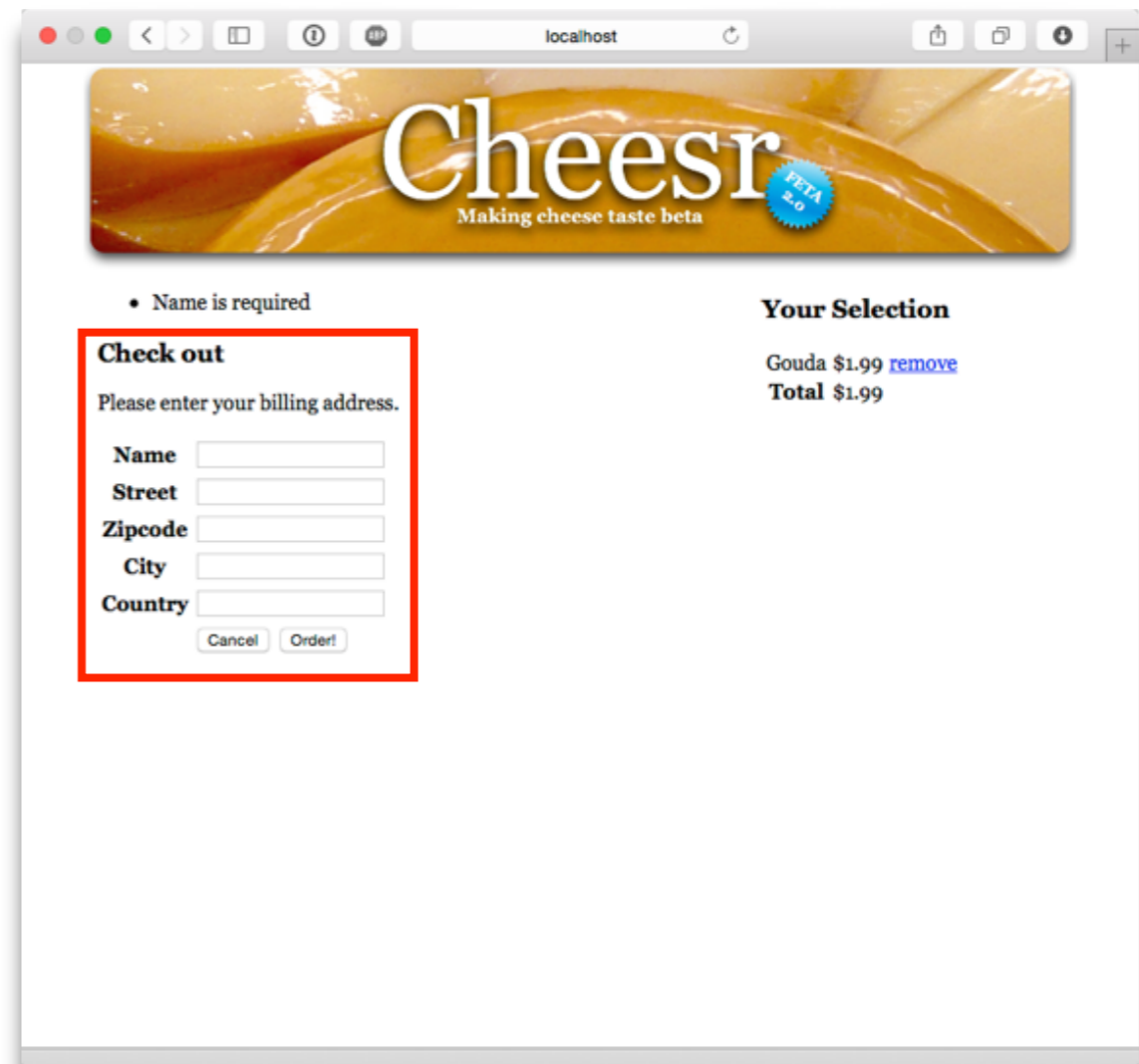
```



```

<form wicket:id="form">
  <h3>Check out</h3>
  <p>Please enter your billing address.</p>
  <table>
    <tr>
      <th>Name</th>
      <td><input type="text" /></td>
    </tr>
    <tr>
      <th>Street</th>
      <td><input type="text" /></td>
    </tr>
    <tr>
      <th>Zipcode</th>
      <td><input type="text" /></td>
    </tr>
    <tr>
      <th>City</th>
      <td><input type="text" /></td>
    </tr>
  </table>
  <input type="button" value="Cancel" />
  <input type="button" value="Order!" />
</form>

```




```
public class CheckoutPage extends WebPage {  
    private Cart cart;  
  
    public CheckoutPage(Cart cart) {  
        this.cart = cart;  
    }  
}
```

```
public class CheckoutPage extends WebPage {  
    private Cart cart;  
  
    public CheckoutPage(Cart cart) {  
        this.cart = cart;  
  
        Form<Void> form = new Form<Void>("form") {};  
        add(form);  
    }  
}
```

```
public class CheckoutPage extends WebPage {
    private Cart cart;

    public CheckoutPage(Cart cart) {
        this.cart = cart;

        Form<Void> form = new Form<Void>("form") {
            @Override
            protected void onSubmit() {
            }
        };
        add(form);
    }
}
```

```
public class CheckoutPage extends WebPage {  
    private Cart cart;  
  
    private Order order = new Order();  
  
    public CheckoutPage(Cart cart) {  
        this.cart = cart;  
  
        Form<Void> form = new Form<Void>("form") {  
            @Override  
            protected void onSubmit() {  
            }  
        };  
        add(form);  
    }  
}
```

```
public class CheckoutPage extends WebPage {
    private Cart cart;

    private Order order = new Order();

    @EJB
    private OrderDao orders;

    public CheckoutPage(Cart cart) {
        this.cart = cart;

        Form<Void> form = new Form<Void>("form") {
            @Override
            protected void onSubmit() {
            }
        };
        add(form);
    }
}
```

```
public class CheckoutPage extends WebPage {  
    ...  
    public CheckoutPage(Cart cart) {  
        this.cart = cart;  
        Form<Void> form = new Form<Void>("form") {  
            @Override  
            protected void onSubmit() {  
            }  
        };  
        add(form);  
    }  
}
```

```
public class CheckoutPage extends WebPage {
    ...
    public CheckoutPage(Cart cart) {
        this.cart = cart;

        Form<Void> form = new Form<Void>("form") {
            @Override
            protected void onSubmit() {
                for (Cheese cheese : cart.getItems()) {
                }
            }
        };
        add(form);
    }
}
```

```
public class CheckoutPage extends WebPage {
    ...
    public CheckoutPage(Cart cart) {
        this.cart = cart;

        Form<Void> form = new Form<Void>("form") {
            @Override
            protected void onSubmit() {
                for (Cheese cheese : cart.getItems()) {
                    OrderItem item = new OrderItem();
                    item.setOrder(order);
                    item.setCheese(cheese);
                    order.getItems().add(item);
                }
            }
        };
        add(form);
    }
}
```



```
public class CheckoutPage extends WebPage {
    ...
    public CheckoutPage(Cart cart) {
        this.cart = cart;

        Form<Void> form = new Form<Void>("form") {
            @Override
            protected void onSubmit() {
                for (Cheese cheese : cart.getItems()) {
                    OrderItem item = new OrderItem();
                    item.setOrder(order);
                    item.setCheese(cheese);
                    order.getItems().add(item);
                }
                orders.save(order);
            }
        };
        add(form);
    }
}
```

6. make Cart Conversation Scoped



What does the navigation between
shopping and **checkout** look like?

```
public class Index extends WebPage {
    public Index() {
        ...
        add(new Link<Void>("checkout") {
            @Override
            public void onClick() {
                setResponsePage(new CheckoutPage(cart));
            }
        });
    }
}
```

```
public class CheckoutPage extends WebPage {
    private Cart cart;

    public CheckoutPage(Cart cart) {
        this.cart = cart;
        ...
    }
}
```

```
public class Index extends WebPage {
    public Index() {
        ...
        add(new Link<Void>("checkout") {
            @Override
            public void onClick() {
                setResponsePage(new CheckoutPage(cart));
            }
        });
    }
}
```

```
public class CheckoutPage extends WebPage {
    private Cart cart;

    public CheckoutPage(Cart cart) {
        this.cart = cart;
        ...
    }
}
```

```
public class Index extends WebPage {
    public Index() {
        ...
        add(new Link<Void>("checkout") {
            @Override
            public void onClick() {
                setResponsePage(new CheckoutPage(cart));
            }
        });
    }
}
```

Cart should be

conversation scoped

```
private Cart cart;

public CheckoutPage(Cart cart) {
    this.cart = cart;
    ...
}
}
```

```
public class Cart implements Serializable {
    public Cart() {
    }

    public void setItems(List<Cheese> items) {
        this.items = items;
    }

    public List<Cheese> getItems() {
        return items;
    }

    public double getTotal() {
        return items.stream().mapToDouble(Cheese::getPrice).su
    }
}
```

```
@ConversationScoped
public class Cart implements Serializable {
    private List<Cheese> items = new ArrayList<>();

    public Cart() {
    }

    public void setItems(List<Cheese> items) {
        this.items = items;
    }

    public List<Cheese> getItems() {
        return items;
    }

    public double getTotal() {
        return items.stream().mapToDouble(Cheese::getPrice).sum()
    }
}
```


Let's get this conversation started

```
public class Index extends WebPage {
    ...
    public Index() {
        ...
        add(new Link<Void>("checkout") {
            @Override
            public void onClick() {
                setResponsePage(new CheckoutPage(cart));
            }
        });
    }
}
```

```
public class Index extends WebPage {
    ...
    @Inject
    private Conversation shopping;

    public Index() {
        ...
        add(new Link<Void>("checkout") {
            @Override
            public void onClick() {
                setResponsePage(new CheckoutPage(cart));
            }
        });
    }
}
```

```
public class Index extends WebPage {
    ...
    @Inject
    private Conversation shopping;

    public Index() {
        ...
        shopping.begin();
        ...
        add(new Link<Void>("checkout") {
            @Override
            public void onClick() {
                setResponsePage(new CheckoutPage(cart));
            }
        });
    }
}
```

```
public class Index extends WebPage {  
    ...  
    @Inject  
    private Conversation shopping;  
  
    @Inject  
    private Cart cart = new Cart();  
  
    public Index() {  
        ...  
        shopping.begin();  
        ...  
        add(new Link<Void>("checkout") {  
            @Override  
            public void onClick() {  
                setResponsePage(new CheckoutPage(cart));  
            }  
        });  
    }  
}
```

```
public class Index extends WebPage {
    ...
    @Inject
    private Conversation shopping;

    @Inject
    private Cart cart;

    public Index() {
        ...
        shopping.begin();
        ...
        add(new Link<Void>("checkout") {
            @Override
            public void onClick() {
                setResponsePage(new CheckoutPage(cart));
            }
        });
    }
}
```

```
public class Index extends WebPage {
    ...
    @Inject
    private Conversation shopping;

    @Inject
    private Cart cart;

    public Index() {
        ...
        shopping.begin();
        ...
        add(new Link<Void>("checkout") {
            @Override
            public void onClick() {
                setResponsePage(new CheckoutPage());
            }
        });
    }
}
```

```
public class CheckoutPage extends WebPage {
    @EJB
    private OrderDao orders;

    private Order order = new Order();

    private Cart cart;

    public CheckoutPage(Cart cart) {
        this.cart = cart;

        Form<Void> form = new Form<Void>("form") {...};
        add(form);
    }
}
```



```
public class CheckoutPage extends WebPage {
    @EJB
    private OrderDao orders;

    private Order order = new Order();

    private Cart cart;

    public CheckoutPage() {
        this.cart = cart;

        Form<Void> form = new Form<Void>("form") {...};
        add(form);
    }
}
```

```
public class CheckoutPage extends WebPage {
    @EJB
    private OrderDao orders;

    private Order order = new Order();

    private Cart cart;

    public CheckoutPage() {
        Form<Void> form = new Form<Void>("form") {...};
        add(form);
    }
}
```

```
public class CheckoutPage extends WebPage {
    @EJB
    private OrderDao orders;

    private Order order = new Order();


    @Inject
    private Cart cart;

    public CheckoutPage() {
        Form<Void> form = new Form<Void>("form") {...};
        add(form);
    }
}
```

Let's end this conversation

```
public class CheckoutPage extends WebPage {  
    ...  
    @Inject  
    private Cart cart;  
  
    public CheckoutPage() {  
        Form<Void> form = new Form<Void>("form") {...};  
        add(form);  
    }  
}
```

```
public class CheckoutPage extends WebPage {  
    ...  
    @Inject  
    private Cart cart;  
  
    public CheckoutPage() {  
        Form<Void> form = new Form<Void>("form") {  
            @Override  
            protected void onSubmit() {  
                for (Cheese cheese : cart.getItems()) {...}  
                orders.save(order);  
            }  
        };  
        add(form);  
    }  
}
```



end conversation here...

```
public class CheckoutPage extends WebPage {
    ...
    @Inject
    private Conversation shopping;

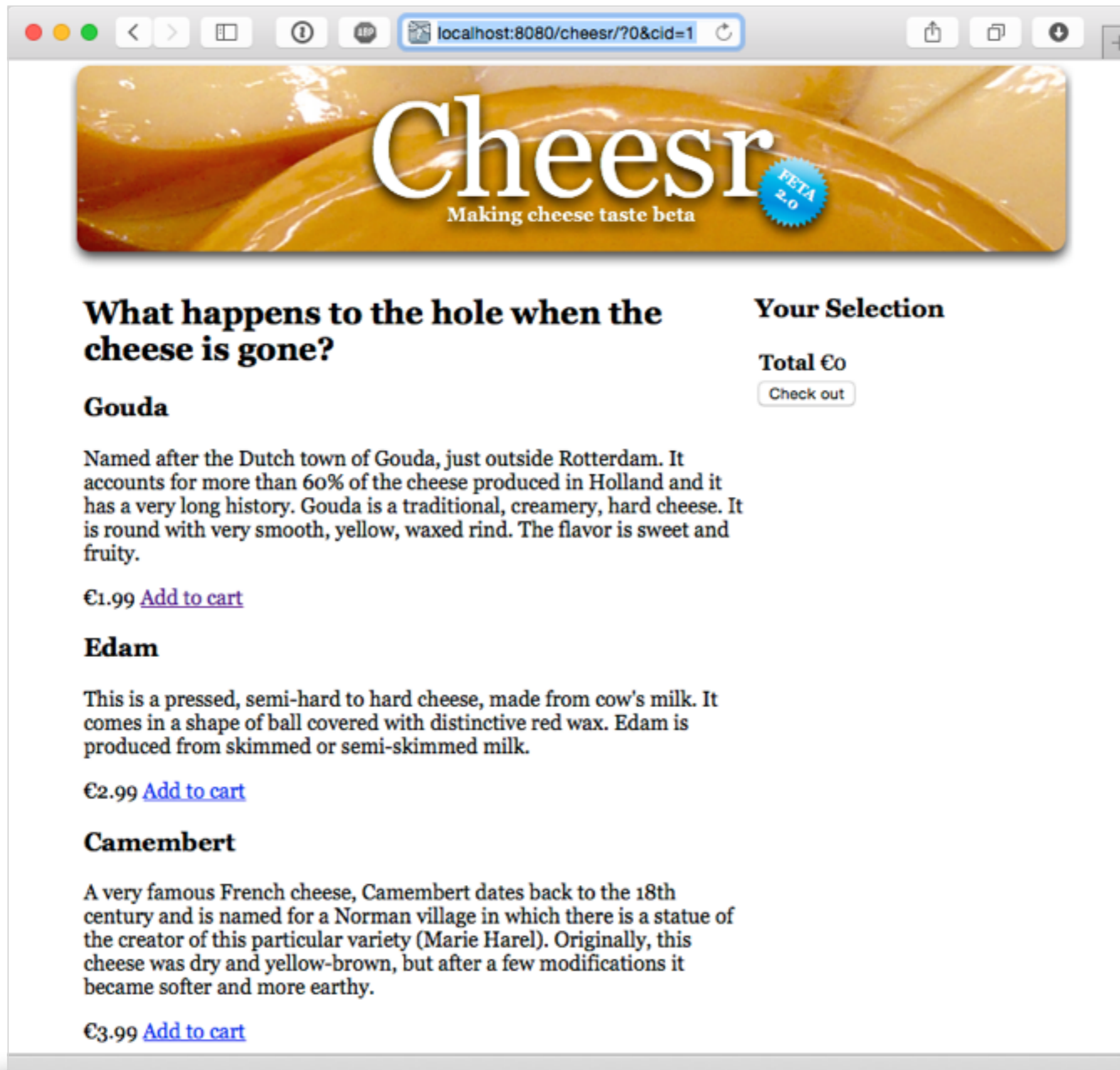
    @Inject
    private Cart cart;

    public CheckoutPage() {
        Form<Void> form = new Form<Void>("form") {
            @Override
            protected void onSubmit() {
                for (Cheese cheese : cart.getItems()) {}
                orders.save(order);
            }
        };
        add(form);
    }
}
```

```
public class CheckoutPage extends WebPage {
    ...
    @Inject
    private Conversation shopping;

    @Inject
    private Cart cart;

    public CheckoutPage() {
        Form<Void> form = new Form<Void>("form") {
            @Override
            protected void onSubmit() {
                for (Cheese cheese : cart.getItems()) {}
                orders.save(order);
                shopping.end();
            }
        };
        add(form);
    }
}
```

Cheesr

Making cheese taste beta

FETA
2.0

What happens to the hole when the cheese is gone?

Gouda

Named after the Dutch town of Gouda, just outside Rotterdam. It accounts for more than 60% of the cheese produced in Holland and it has a very long history. Gouda is a traditional, creamery, hard cheese. It is round with very smooth, yellow, waxed rind. The flavor is sweet and fruity.

€1.99 [Add to cart](#)

Edam

This is a pressed, semi-hard to hard cheese, made from cow's milk. It comes in a shape of ball covered with distinctive red wax. Edam is produced from skimmed or semi-skimmed milk.

€2.99 [Add to cart](#)

Camembert

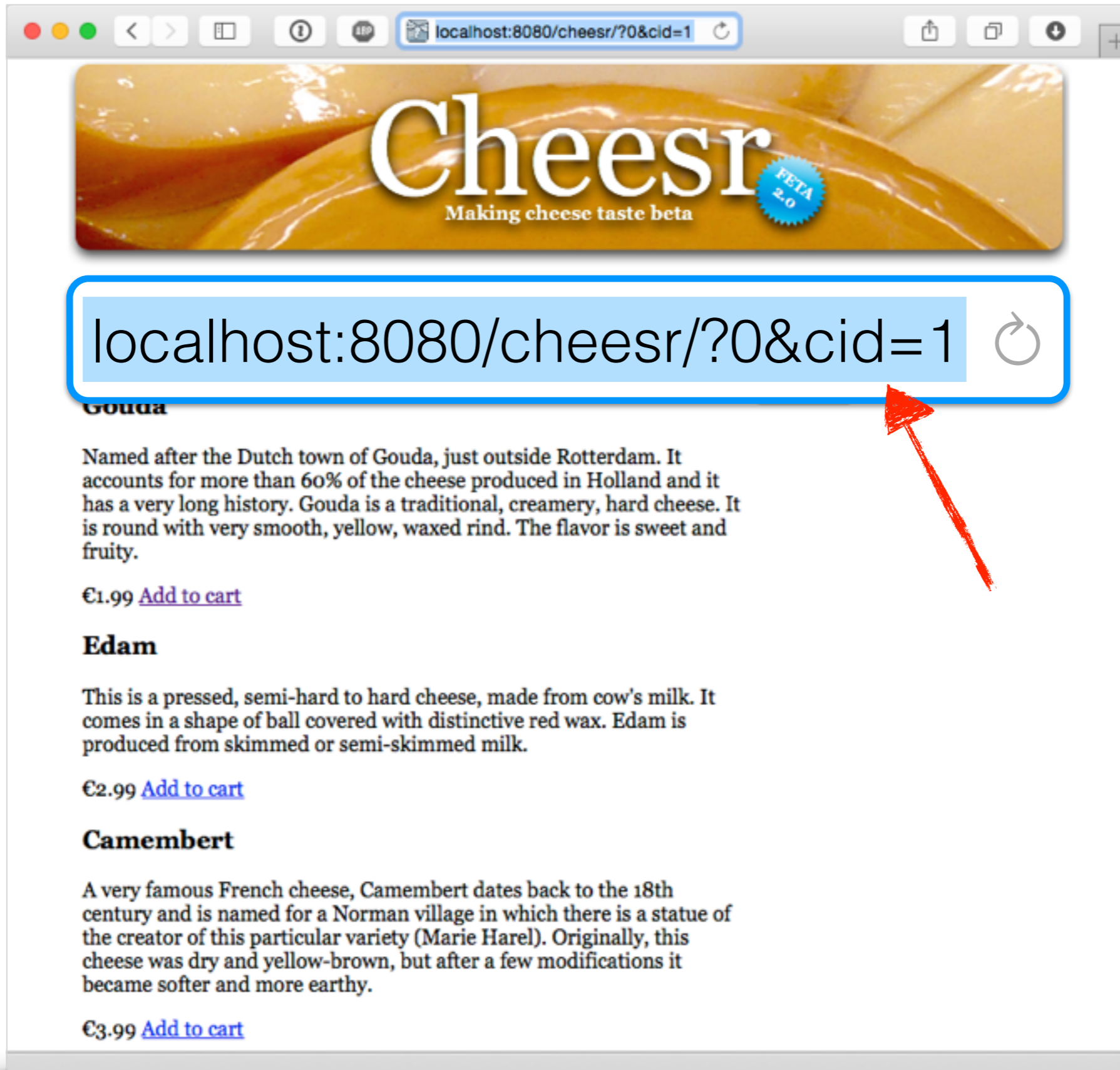
A very famous French cheese, Camembert dates back to the 18th century and is named for a Norman village in which there is a statue of the creator of this particular variety (Marie Harel). Originally, this cheese was dry and yellow-brown, but after a few modifications it became softer and more earthy.

€3.99 [Add to cart](#)

Your Selection

Total €0

[Check out](#)



localhost:8080/cheesr/?0&cid=1

Gouda

Named after the Dutch town of Gouda, just outside Rotterdam. It accounts for more than 60% of the cheese produced in Holland and it has a very long history. Gouda is a traditional, creamery, hard cheese. It is round with very smooth, yellow, waxed rind. The flavor is sweet and fruity.

€1.99 [Add to cart](#)

Edam

This is a pressed, semi-hard to hard cheese, made from cow's milk. It comes in a shape of ball covered with distinctive red wax. Edam is produced from skimmed or semi-skimmed milk.

€2.99 [Add to cart](#)

Camembert

A very famous French cheese, Camembert dates back to the 18th century and is named for a Norman village in which there is a statue of the creator of this particular variety (Marie Harel). Originally, this cheese was dry and yellow-brown, but after a few modifications it became softer and more earthy.

€3.99 [Add to cart](#)


```
public class CheckoutPage extends WebPage {
    ...
    @Inject
    private Conversation shopping;

    @Inject
    private Cart cart;

    public CheckoutPage() {
        Form<Void> form = new Form<Void>("form") {
            @Override
            protected void onSubmit() {
                for (Cheese cheese : cart.getItems()) {}
                orders.save(order);
                shopping.end();
            }
        };
        add(form);
    }
}
```

```
public class CheckoutPage extends WebPage {
    ...
    @Inject
    private Conversation shopping;

    @Inject
    private Cart cart;

    public CheckoutPage() {
        Form<Void> form = new Form<Void>("form") {
            @Override
            protected void onSubmit() {
                for (Cheese cheese : cart.getItems()) {}
                orders.save(order);
                shopping.end();
            }
        };
        add(form);
    }
}
```

```
public class CheckoutPage extends WebPage {  
    ...  
    @Inject  
    private Conversation shopping;  
  
    @Inject  
    private Cart cart;  
  
    public CheckoutPage() {  
        Form<Void> form = new Form<Void>("form") {  
            @Override  
            protected void onSubmit() {  
                for (Cheese cheese : cart.getItems()) {}  
                orders.save(order);  
                shopping.end();  
            }  
        };  
        add(form);  
    }  
}
```

```
public class CheckoutPage extends WebPage {
    ...
    @Inject
    private Conversation shopping;

    @Inject
    private Cart cart;

    public CheckoutPage() {
        Form<Void> form = new Form<Void>("form") {
            @Override
            protected void onSubmit() {
                for (Cheese cheese : cart.getItems()) {}
                orders.save(order);
                shopping.end();
                setResponsePage(Index.class);
            }
        };
        add(form);
    }
}
```

- Add wicket-cdi-1.1 and javax.cdi-api
- Configure Wicket's CdiConfiguration
- Use @Inject injection **and**
@ConversationScoped
- Use NonContextual in non-managed objects

Injection (EJB, CDI)

Conversations (CDI)

JPA

Bean validation

Testing with Arquillian



- Cheese, Order, OrderItem now **@Entity**
- Order → OrderItem
@OneToMany(fetchType=LAZY)
- OrderItem → Cheese, OrderItem → Order
@ManyToOne(optional=false)
- CheeseDao, OrderDao now
@Stateless
@TransactionAttribute(REQUIRED)

A refreshing model

A model that reloads data from DB
for every render



```
public class OrdersModel {  
}
```

```
public class OrdersModel
    extends LoadableDetachableModel<List<Order>> {
}
```

```
public class OrdersModel
    extends LoadableDetachableModel<List<Order>> {
    @Override
    protected List<Order> load() {
    }
}
```

```
public class OrdersModel
    extends LoadableDetachableModel<List<Order>> {
    @Override
    protected List<Order> load() {
        return orders.list();
    }
}
```

```
public class OrdersModel
    extends LoadableDetachableModel<List<Order>> {
    @Inject
    private OrderDao orders;

    @Override
    protected List<Order> load() {
        return orders.list();
    }
}
```



```
public class OrdersModel
    extends LoadableDetachableModel<List<Order>> {
    @Inject
    private OrderDao orders;

    public OrdersModel() {
    }

    @Override
    protected List<Order> load() {
        return orders.list();
    }
}
```

```
public class OrdersModel
    extends LoadableDetachableModel<List<Order>> {
    @Inject
    private OrderDao orders;

    public OrdersModel() {
        NonContextual.of(OrdersModel.class).inject(this);
    }

    @Override
    protected List<Order> load() {
        return orders.list();
    }
}
```

```
public class OrdersPage extends WebPage {
    public OrdersPage() {
        add(new ListView<Order>("items", new OrdersModel()) {
            @Override
            protected void populateItem(ListItem<Order> item) {
                Order o = item.getModelObject();
                item.add(new Label("id", o.getId()));
                item.add(new Label("name", o.getName()));
                item.add(new Label("total", o.getTotal()));
            }
        });
    }
}
```

```
public class OrdersPage extends WebPage {  
    public OrdersPage() {  
    }  
}
```

```
public class OrdersPage extends WebPage {  
    public OrdersPage() {  
        add(new ListView<Order>("items", new OrdersModel()) {  
        });  
    }  
}
```

```
public class OrdersPage extends WebPage {
    public OrdersPage() {
        add(new ListView<Order>("items", new OrdersModel()) {
            @Override
            protected void populateItem(ListItem<Order> item) {
            }
        });
    }
}
```

```
public class OrdersPage extends WebPage {
    public OrdersPage() {
        add(new ListView<Order>("items", new OrdersModel()) {
            @Override
            protected void populateItem(ListItem<Order> item) {
                Order o = item.getModelObject();
                item.add(new Label("id", o.getId()));
                item.add(new Label("name", o.getName()));
                item.add(new Label("total", o.getTotal()));
            }
        });
    }
}
```

localhost

Unexpected RuntimeException

Last cause: failed to lazily initialize a collection of role: com.cheesr.entities.Order.items, could not initialize proxy - WicketMessage: Error attaching this container for rendering: [Page class = com.cheesr.web.OrdersPage, id = 6, r...

Stacktrace

Root cause:

```
org.hibernate.LazyInitializationException: failed to lazily initialize a collection of role: com.cheesr.entities.Order.items
at org.hibernate.collection.internal.AbstractPersistentCollection.throwLazyInitializationException(AbstractPersistentCollection.java:554)
at org.hibernate.collection.internal.AbstractPersistentCollection.withTemporarySessionIfNeeded(AbstractPersistentCollection.java:142)
at org.hibernate.collection.internal.AbstractPersistentCollection.initialize(AbstractPersistentCollection.java:142)
at org.hibernate.collection.internal.PersistentBag.iterator(PersistentBag.java:294)
at java.util.Spliterators$IteratorSpliterator.estimateSize(Spliterators.java:1821)
at java.util.Spliterator.getExactSizeIfKnown(Spliterator.java:408)
at java.util.stream.AbstractPipeline.copyInto(AbstractPipeline.java:511)
at java.util.stream.AbstractPipeline.wrapAndCopyInto(AbstractPipeline.java:502)
at java.util.stream.ReduceOps$ReduceOp.evaluateSequential(ReduceOps.java:708)
at java.util.stream.AbstractPipeline.evaluate(AbstractPipeline.java:234)
at java.util.stream.DoublePipeline.collect(DoublePipeline.java:476)
at java.util.stream.DoublePipeline.sum(DoublePipeline.java:388)
at com.cheesr.entities.Order.getTotal(Order.java:94)
at com.cheesr.web.OrdersPage$1.populateItem(OrdersPage.java:22)
at org.apache.wicket.markup.html.list.ListView.onPopulate(ListView.java:523)
at org.apache.wicket.markup.repeater.AbstractRepeater.onBeforeRender(AbstractRepeater.java:123)
at org.apache.wicket.Component.internalBeforeRender(Component.java:949)
at org.apache.wicket.Component.beforeRender(Component.java:1017)
at org.apache.wicket.MarkupContainer.onBeforeRenderChildren(MarkupContainer.java:1684)
at org.apache.wicket.Component.onBeforeRender(Component.java:3826)
at org.apache.wicket.Page.onBeforeRender(Page.java:809)
at org.apache.wicket.Component.internalBeforeRender(Component.java:949)
at org.apache.wicket.Component.beforeRender(Component.java:1017)
at org.apache.wicket.Component.internalPrepareForRender(Component.java:2194)
at org.apache.wicket.Page.internalPrepareForRender(Page.java:240)
at org.apache.wicket.Component.render(Component.java:2283)
```


Caused by: [org.hibernate.LazyInitializationException](#): failed to lazily initialize a collection of role: com.cheesr.entities.Order.items, could not initialize proxy - no Session

```
at org.hibernate.collection.internal.AbstractPersistentCollection.throw
at org.hibernate.collection.internal.AbstractPersistentCollection.withT
at org.hibernate.collection.internal.AbstractPersistentCollection.initi
at org.hibernate.collection.internal.AbstractPersistentCollection.read(
at org.hibernate.collection.internal.PersistentBag.iterator(PersistentB
at java.util.Spliterators$IteratorSpliterator.estimateSize(Spliterators
at java.util.Spliterator.getExactSizeIfKnown(Spliterator.java:408)
at java.util.stream.AbstractPipeline.copyInto(AbstractPipeline.java:511
at java.util.stream.AbstractPipeline.wrapAndCopyInto(AbstractPipeline.j
at java.util.stream.ReduceOps$ReduceOp.evaluateSequential(ReduceOps.jav
at java.util.stream.AbstractPipeline.evaluate(AbstractPipeline.java:234
at java.util.stream.DoublePipeline.collect(DoublePipeline.java:476)
at java.util.stream.DoublePipeline.sum(DoublePipeline.java:388)
at com.cheesr.entities.Order.getTotal(Order.java:94)
at com.cheesr.web.OrdersPage$1.populateItem(OrdersPage.java:22)
at org.apache.wicket.markup.html.list.ListView.onPopulate(ListView.java
at org.apache.wicket.markup.repeater.AbstractRepeater.onBeforeRender(Ab
at org.apache.wicket.Component.internalBeforeRender(Component.java:949)
at org.apache.wicket.Component.beforeRender(Component.java:1017)
at org.apache.wicket.MarkupContainer.onBeforeRenderChildren(MarkupConta
... 50 more
```

LazyInitializationException

failed to lazily initialize a collection of role:

com.cheesr.entities.Order.items, could not initialize proxy

- no Session

LazyInitException: AARGH

Why and how to get rid of them



```
@Stateless
@Transactional(TransactionalType.REQUIRED)
public class OrderDao implements Serializable {

    public void save(Order order) {...}

    public List<Order> list() {...}
}
```

```
@Stateless
```

```
@TransactionAttribute(TransactionAttributeType.REQUIRED)
```

```
public class OrderDao implements Serializable {
```

```
    public void save(Order order) {...}
```

```
    public List<Order> list() {...}
```

```
}
```

After dao.list() JPA session ends

```
@Entity
@Table(name = "Orders")
public class Order implements Serializable {
    @Id
    @GeneratedValue
    private Long id;
    private String name;
    private String street;
    private String zipcode;
    private String city;
    private String country;

    @OneToMany(fetch = FetchType.LAZY,
        mappedBy = "order",
        cascade = CascadeType.ALL,
        orphanRemoval = true)
    private List<OrderItem> items = new ArrayList<>();

    public Long getId() {
        return id;
    }
}
```

```
@Entity
@Table(name = "Orders")
public class Order implements Serializable {
    @Id
    @GeneratedValue
    private Long id;
    private String name;
    private String street;
    private String zipcode;
    private String city;
    private String country;

    @OneToMany(fetch = FetchType.LAZY,
        mappedBy = "order",
        cascade = CascadeType.ALL,
        orphanRemoval = true)
    private List<OrderItem> items = new ArrayList<>();

    public Long getId() {
        return id;
    }
}
```


Generates lazy proxies, resolved
upon request

```
public class OrdersPage extends WebPage {
    public OrdersPage() {
        add(new ListView<Order>("items", new OrdersModel()) {
            @Override
            protected void populateItem(ListItem<Order> item) {
                Order o = item.getModelObject();
                item.add(new Label("id", o.getId()));
                item.add(new Label("name", o.getName()));
                item.add(new Label("total", o.getTotal()));
            }
        });
    }
}
```

```
public class OrdersPage extends WebPage {
    public OrdersPage() {
        add(new ListView<Order>("items", new OrdersModel()) {
            @Override
            protected void populateItem(ListItem<Order> item) {
                Order o = item.getModelObject();
                item.add(new Label("id", o.getId()));
                item.add(new Label("name", o.getName()));
                item.add(new Label("total", o.getTotal()));
            }
        });
    }
}
```

getTotal() requests upon the
proxy: LazyInitException

Fix LazyInitExceptions

Extend transaction for whole request
(using a Servlet 3.0 @WebFilter)



```
@WebFilter(filterName = "Cheesr",
    value = "/*",
    initParams = {
        @WebInitParam(
            name = "applicationClassName",
            value = "com.cheesr.web.WicketApplication"),
        @WebInitParam(
            name = "filterMappingUrlPattern",
            value = "/*")
    })
public class CheesrFilter extends WicketFilter {
    @Override
    public void doFilter(ServletRequest req, ServletResponse
        super.doFilter(request, response, chain);
    }
}
```

```
@WebFilter(filterName = "Cheesr",
    value = "/*",
    initParams = {
        @WebInitParam(
            name = "applicationClassName",
            value = "com.cheesr.web.WicketApplication"),
        @WebInitParam(
            name = "filterMappingUrlPattern",
            value = "/*")
    })
public class CheesrFilter extends WicketFilter {
    @Transactional
    @Override
    public void doFilter(ServletRequest req, ServletResponse
        super.doFilter(request, response, chain);
    }
}
```



Orders

1 Martijn Dashorst €5.42

Injection (EJB, CDI)

Conversations (CDI)

JPA

Bean validation

Testing with Arquillian



Bean Validation

Single Point of Definition



Steps

Add dependencies

Add validation constraints

Configure Wicket

Add Validators to components



```
<!-- BEAN VALIDATION DEPENDENCIES -->
<dependency>
  <groupId>javax.validation</groupId>
  <artifactId>validation-api</artifactId>
  <scope>provided</scope>
</dependency>
<dependency>
  <groupId>org.apache.wicket</groupId>
  <artifactId>wicket-bean-validation</artifactId>
</dependency>
```

```
@Entity
@Table(name = "Orders")
public class Order implements Serializable {
    @Id
    @GeneratedValue
    private Long id;

    private String name;

    private String street;

    private String zipcode;

    private String city;

    private String country;

    @OneToMany(fetch = FetchType.LAZY, mappedBy = "order", ca
    private List<OrderItem> items = new ArrayList<>();
```

```
@Entity
@Table(name = "Orders")
public class Order implements Serializable {
    @Id
    @GeneratedValue
    private Long id;

    @NotNull
    private String name;

    private String street;

    private String zipcode;

    private String city;

    private String country;

    @OneToMany(fetch = FetchType.LAZY, mappedBy = "order", ca
    private List<OrderItem> items = new ArrayList<>();
```

```
@Entity
@Table(name = "Orders")
public class Order implements Serializable {
    @Id
    @GeneratedValue
    private Long id;

    @NotNull
    private String name;

    @Size(min = 3, max = 20)
    @NotNull
    private String street;

    private String zipcode;

    private String city;

    private String country;
```

```
@Entity
@Table(name = "Orders")
public class Order implements Serializable {
    @Id
    @GeneratedValue
    private Long id;

    @NotNull
    private String name;

    @Size(min = 3, max = 20)
    @NotNull
    private String street;

    @Pattern(regexp = "\\d{4} [A-Z]{2}")
    @NotNull
    private String zipcode;

    private String city;
```



```
public void init() {  
    super.init();
```

```
    BeanValidationConfiguration beanValidation =  
        new BeanValidationConfiguration();  
    beanValidation.configure(this);
```

```
    CdiConfiguration cdiConfiguration =  
        new CdiConfiguration();  
    cdiConfiguration  
        .setPropagation(ConversationPropagation.ALL);  
    cdiConfiguration.configure(this);
```

```
    getComponentInstantiationListeners().add(  
        new JavaEEComponentInjector(this,  
            new WildflyWicketJndiNamingStrategy()));
```

```
    mountPage("/checkout", CheckoutPage.class);  
    mountPage("/orders", OrdersPage.class);
```

```
}
```

```
TextField<String> nameField =  
    new TextField<>("name", PropertyModel.of(this, "order.name"));  
form.add(nameField);
```

```
TextField<String> streetField =  
    new TextField<>("street", PropertyModel.of(this, "order.street"));  
form.add(streetField);
```

```
TextField<String> nameField =  
    new TextField<>("name", PropertyModel.of(this, "order.name"));  
nameField.add(new PropertyValidator<>());  
form.add(nameField);
```

```
TextField<String> streetField =  
    new TextField<>("street", PropertyModel.of(this, "order.street"));  
form.add(streetField);
```

```
TextField<String> nameField =  
    new TextField<>("name", PropertyModel.of(this, "order.name"));  
nameField.add(new PropertyValidator<>());  
form.add(nameField);
```

```
TextField<String> streetField =  
    new TextField<>("street", PropertyModel.of(this, "order.street"));  
streetField.add(new PropertyValidator<>());  
form.add(streetField);
```

add **wicket-bean-validation** and
javax.validation:validation-api to project

configure **BeanValidationConfiguration** in **init()**

add **PropertyValidator** to fields

Injection (EJB, CDI)

Conversations (CDI)

JPA

Bean validation

Testing with Arquillian



Testing with WicketTester

```
public class TestIndex {
    private WicketTester tester;

    @Before
    public void setUp() {
        WicketApplication app = new WicketApplication();
        tester = new WicketTester(app);
    }

    @Test
    public void homepageRendersSuccessfully() {
        tester.startPage(Index.class);

        tester.assertRenderedPage(Index.class);
    }
}
```


- No bean manager
- No data source
- No persistence context
- No transactions
- No EJBs

**NO
TESTS**

Arquillian

No more mocks. No more container lifecycle and deployment hassles. Just real tests!



Steps

Add Arquillian dependencies to project

Build deployable archive with all necessary resources

Create a test case

Execute test case with Arquillian

```
<!-- ARQUILLIAN DEPENDENCY FOR TESTING -->
<dependency>
  <groupId>org.jboss.arquillian.junit</groupId>
  <artifactId>arquillian-junit-container</artifactId>
  <scope>test</scope>
</dependency>
```

```
<!-- ARQUILLIAN DEPENDENCY FOR TESTING -->
<dependency>
  <groupId>org.jboss.arquillian.junit</groupId>
  <artifactId>arquillian-junit-container</artifactId>
  <scope>test</scope>
</dependency>
<dependency>
  <groupId>org.jboss.shrinkwrap.descriptors</groupId>
  <artifactId>shrinkwrap-descriptors-impl-javaee</artifact
  <scope>test</scope>
</dependency>
```

```
<!-- ARQUILLIAN DEPENDENCY FOR TESTING -->
<dependency>
  <groupId>org.jboss.arquillian.junit</groupId>
  <artifactId>arquillian-junit-container</artifactId>
  <scope>test</scope>
</dependency>
<dependency>
  <groupId>org.jboss.shrinkwrap.descriptors</groupId>
  <artifactId>shrinkwrap-descriptors-impl-javaee</artifact
  <scope>test</scope>
</dependency>
<dependency>
  <groupId>org.jboss.shrinkwrap.resolver</groupId>
  <artifactId>shrinkwrap-resolver-impl-maven</artifactId>
  <scope>test</scope>
</dependency>
```

Steps

Add Arquillian dependencies to project

Build deployable archive with all necessary resources

Create a test case

Execute test case with Arquillian




```
public class TestIndex {
    private WicketTester tester;

    @Before
    public void setUp() {
        WicketApplication app = new WicketApplication();
        tester = new WicketTester(app);
    }

    @Test
    public void homepageRendersSuccessfully() {
        tester.startPage(Index.class);

        tester.assertRenderedPage(Index.class);
    }
}
```

```
@RunWith(Arquillian.class)
```

```
public class TestIndex {  
    private WicketTester tester;
```

```
@Before
```

```
public void setUp() {  
    WicketApplication app = new WicketApplication();  
    tester = new WicketTester(app);  
}
```

```
@Test
```

```
public void homepageRendersSuccessfully() {  
    tester.startPage(Index.class);  
  
    tester.assertRenderedPage(Index.class);  
}
```

```
}
```

```
@RunWith(Arquillian.class)
public class TestIndex {
    @Deployment
    public static WebArchive deployment() {...}

    private WicketTester tester;

    @Before
    public void setUp() {
        WicketApplication app = new WicketApplication();
        tester = new WicketTester(app);
    }

    @Test
    public void homepageRendersSuccessfully() {
        tester.startPage(Index.class);

        tester.assertRenderedPage(Index.class);
    }
}
```

```
@Deployment
public static WebArchive deployment() {
    File[] dependencies =
        Maven.configureResolver()
            .workOffline()
            .loadPomFromFile("pom.xml")
            .importCompileAndRuntimeDependencies()
            .resolve()
            .withTransitivity()
            .as(File.class);
}
```

```
@Deployment
public static WebArchive deployment() {
    File[] dependencies =
        Maven.configureResolver()
            .workOffline()
            .loadPomFromFile("pom.xml")
            .importCompileAndRuntimeDependencies()
            .resolve()
            .withTransitivity()
            .as(File.class);
    WebArchive war = ShrinkWrap
        .create(WebArchive.class)
        .addAsResource(new File("target/classes"), "")
        .addAsLibraries(dependencies)
        .addAsWebInfResource(
            new File("src/main/webapp/WEB-INF/beans.xml"))
        .addAsWebInfResource(
            new File("src/main/webapp/WEB-INF/cheersr-ds.xml"));
    return war;
}
```

```
6ea1e1a7-7f07-4318-8f70-ec7b8d5edbcc.war:
/WEB-INF/
/WEB-INF/lib/
/WEB-INF/lib/annotations-3.0.0.jar
/WEB-INF/lib/wicketstuff-javaee-inject-6.17.0.jar
/WEB-INF/lib/javax.inject-1.jar
/WEB-INF/lib/wicket-cdi-1.1-6.18.0.jar
/WEB-INF/lib/cglib-2.2.2.jar
/WEB-INF/lib/fest-util-1.1.6.jar
/WEB-INF/lib/wicket-ioc-6.18.0.jar
/WEB-INF/lib/wicket-bean-validation-6.18.0.jar
/WEB-INF/lib/wicket-request-6.18.0.jar
/WEB-INF/lib/log4j-1.2.17.jar
/WEB-INF/lib/asm-3.3.1.jar
/WEB-INF/lib/fest-assert-1.4.jar
/WEB-INF/lib/validation-api-1.1.0.Final.jar
/WEB-INF/lib/slf4j-log4j12-1.7.7.jar
/WEB-INF/lib/slf4j-api-1.7.7.jar
/WEB-INF/lib/jcl-over-slf4j-1.7.7.jar
/WEB-INF/lib/wicket-core-6.18.0.jar
/WEB-INF/lib/wicket-util-6.18.0.jar
/WEB-INF/cheesr-ds.xml
/WEB-INF/classes/
/WEB-INF/classes/wildfly-doesnt-need-log4j.properties
/WEB-INF/classes/META-INF/
/WEB-INF/classes/META-INF/persistence.xml
/WEB-INF/classes/com/
/WEB-INF/classes/com/cheesr/
/WEB-INF/classes/com/cheesr/domain/
/WEB-INF/classes/com/cheesr/domain/Cart.class
/WEB-INF/classes/com/cheesr/domain/MessageOfTheDay.class
/WEB-INF/classes/com/cheesr/web/
/WEB-INF/classes/com/cheesr/web/CheckoutPage$2.class
/WEB-INF/classes/com/cheesr/web/Index$2$1.class
/WEB-INF/classes/com/cheesr/web/OrdersPage.class
/WEB-INF/classes/com/cheesr/web/WicketApplication.class
/WEB-INF/classes/com/cheesr/web/Index$1$1.class
/WEB-INF/classes/com/cheesr/web/Index.html
/WEB-INF/classes/com/cheesr/web/WicketApplication$WildflyWicketJndiNamingStrategy.class
/WEB-INF/classes/com/cheesr/web/Index$2.class
/WEB-INF/classes/com/cheesr/web/CheckoutPage.class
/WEB-INF/classes/com/cheesr/web/CheckoutPage.html
/WEB-INF/classes/com/cheesr/web/Index.class
/WEB-INF/classes/com/cheesr/web/CheesrFilter.class
/WEB-INF/classes/com/cheesr/web/MessageModel.class
/WEB-INF/classes/com/cheesr/web/Index$3.class
/WEB-INF/classes/com/cheesr/web/OrdersModel.class
```

Steps

Add Arquillian dependencies
to project

Build deployable archive
with all necessary resources

Create a test case

Execute test case with Arquillian

The image shows an IDE window with the following components:

- Package Explorer:** Shows the project structure with 'JUnit' and 'TestIndex' visible. A green bar indicates the test passed.
- Code Editor:** Displays the source code for `TestIndex.java`. The code is as follows:

```
1 package com.cheesr.web;
2
3 import java.io.File;
4
14
15 @RunWith(Arquillian.class)
16 public class TestIndex
17 {
18     @Deployment
19     public static WebArchive deployment()
20     {
21         File[] dependencies =
22             Maven.configureResolver().workOffline().loadPomFromFile("pom.xml")
23                 .importCompileAndRuntimeDependencies()
24                 // .addDependencies()
25                 .resolve().withTransitivity().as(File.class);
26
27         WebArchive war =
28             ShrinkWrap.create(WebArchive.class).addAsResource(new File("target/classes"), "")
29                 .addAsLibraries(dependencies)
30                 .addAsWebInfResource(new File("src/main/webapp/WEB-INF/beans.xml"))
31                 .addAsWebInfResource(new File("src/main/webapp/WEB-INF/cheesr-ds.xml"));
32         System.out.println(war.toString(true));
33         return war;
34     }
35
36     private WicketTester tester;
37
38     @Before
39     public void setUp()
40     {
41         WicketApplication app = new WicketApplication();
42         tester = new WicketTester(app);
43     }
44 }
```
- Console:** Shows the output of the test execution:

```
<terminated> TestIndex [JUnit] /Library/Java/JavaVirtualMachines/jdk1.8.0.jdk/Contents/Home/bin/java (Nov 16, 2014, 2:50:56 AM)
/WEB-INF/classes/com/cheesr/entities/OrderItem.class
/WEB-INF/classes/import.sql
/WEB-INF/classes/rebel.xml
/WEB-INF/beans.xml
INFO - xnio - XNIO version 3.2.3.Final
INFO - nio - XNIO NIO Implementation Version 3.2.3.Final
INFO - remoting - JBoss Remoting version 4.0.5.Final
```


Steps

Add Arquillian dependencies to project

Build deployable archive with all necessary resources

Create a test case

Execute test case with Arquillian

```
@Inject  
private Conversation conversation;
```

```
@EJB  
private CheeseDao cheeses;
```

```
@Inject  
private Cart cart;
```

```
@Test  
public void addCheeseToCart() {  
}
```

```
public void addCheeseToCart() {  
}
```

```
public void addCheeseToCart() {  
    Cheese edam = cheeses.byCode("edam");  
}
```

```
public void addCheeseToCart() {  
    Cheese edam = cheeses.byCode("edam");  
  
    tester.startPage(Index.class);  
    tester.assertRenderedPage(Index.class);  
}
```

```
public void addCheeseToCart() {  
    Cheese edam = cheeses.byCode("edam");  
  
    tester.startPage(Index.class);  
    tester.assertRenderedPage(Index.class);  
    tester.assertModelValue("cheese:1", edam);  
}
```

```
public void addCheeseToCart() {  
    Cheese edam = cheeses.byCode("edam");  
  
    tester.startPage(Index.class);  
    tester.assertRenderedPage(Index.class);  
    tester.assertModelValue("cheese:1", edam);  
  
    tester.clickLink("cheese:1:add");  
    assertThat(cart.getItems().size(), is(1));  
    tester.assertModelValue("item:0", edam);  
}
```

```
public void addCheeseToCart() {  
    Cheese edam = cheeses.byCode("edam");  
  
    tester.startPage(Index.class);  
    tester.assertRenderedPage(Index.class);  
    tester.assertModelValue("cheese:1", edam);  
  
    tester.clickLink("cheese:1:add");  
    assertThat(cart.getItems().size(), is(1));  
    tester.assertModelValue("item:0", edam);  
  
    tester.clickLink("cheese:1:add");  
    assertThat(cart.getItems().size(), is(2));  
    assertThat(cart.getItems().get(0), is(edam));  
    assertThat(cart.getItems().get(1), is(edam));  
}
```



```
public void addCheeseToCart() {
    Cheese edam = cheeses.byCode("edam");

    tester.startPage(Index.class);
    tester.assertRenderedPage(Index.class);
    tester.assertModelValue("cheese:1", edam);

    tester.clickLink("cheese:1:add");
    assertThat(cart.getItems().size(), is(1));
    tester.assertModelValue("item:0", edam);

    tester.clickLink("cheese:1:add");
    assertThat(cart.getItems().size(), is(2));
    assertThat(cart.getItems().get(0), is(edam));
    assertThat(cart.getItems().get(1), is(edam));

    tester.clickLink("checkout");
    tester.assertRenderedPage(CheckoutPage.class);
}
```

Package Explore JUnit

Finished after 4.388 seconds

Runs: 2/2 Errors: 0 Failures: 0

com.cheesr.web.TestIndex [Runner: JUn

```

65     private Conversation conversation;
66
67     @EJB
68     private CheeseDao cheeses;
69
70     @Inject
71     private Cart cart;
72
73     @Test
74     public void addCheeseToCart()
75     {
76         Cheese edam = cheeses.byCode("edam");
77
78         assertThat(cart.getItems().isEmpty(), is(true));
79
80         tester.startPage(Index.class);
81         tester.assertRenderedPage(Index.class);
82
83         tester.assertModelValue("cheese:1", edam);
84         tester.clickLink("cheese:1:add");
85
86         assertThat(cart.getItems().size(), is(1));
87
88         tester.assertModelValue("item:0", edam);
89
90         tester.clickLink("cheese:1:add");
91
92         assertThat(cart.getItems().size(), is(2));
93         assertThat(cart.getItems().get(0), is(edam));
94         assertThat(cart.getItems().get(1), is(edam));
95
96         tester.clickLink("checkout");
97         tester.assertRenderedPage(CheckoutPage.class);
98     }

```

Failure Trace

Problems Javadoc Declaration Servers Console

```

<terminated> TestIndex [JUnit] /Library/Java/JavaVirtualMachines/jdk1.8.0.jdk/Contents/Home/bin/java (Nov 16, 2014, 3:05:01 AM)
/WEB-INF/classes/com/cheesr/entities/OrderItem.class
/WEB-INF/classes/import.sql
/WEB-INF/classes/rebel.xml
/WEB-INF/beans.xml
INFO - xnio - XNIO version 3.2.3.Final
INFO - nio - XNIO NIO Implementation Version 3.2.3.Final
INFO - remoting - JBoss Remoting version 4.0.5.Final

```

Injection (EJB, CDI)

Conversations (CDI)

JPA

Bean validation

Arquillian



Wicket and Java EE?





Questions?





Apache Wicket and Java EE sitting in a tree

Martijn Dashorst
topicus onderwijs



@dashorst
dashorst@apache.org