

Opvallende zaken uit Case 1. De Cursus Administratie van het KC

PI test:	1
Pi test rapportage	3
Over de persistence.....	3
Persistence bij springboot :	4
Verschillen database test/deployment	5
CodeCoverage:.....	6
Opvallende zaken in code.....	7

PI test:

Pi/mutation testing. Doel, een goed beeld krijgen in hoeverre je tests ook werkelijk de code dekken. Aanvullend op Code Coverage.

In (parent) pom:

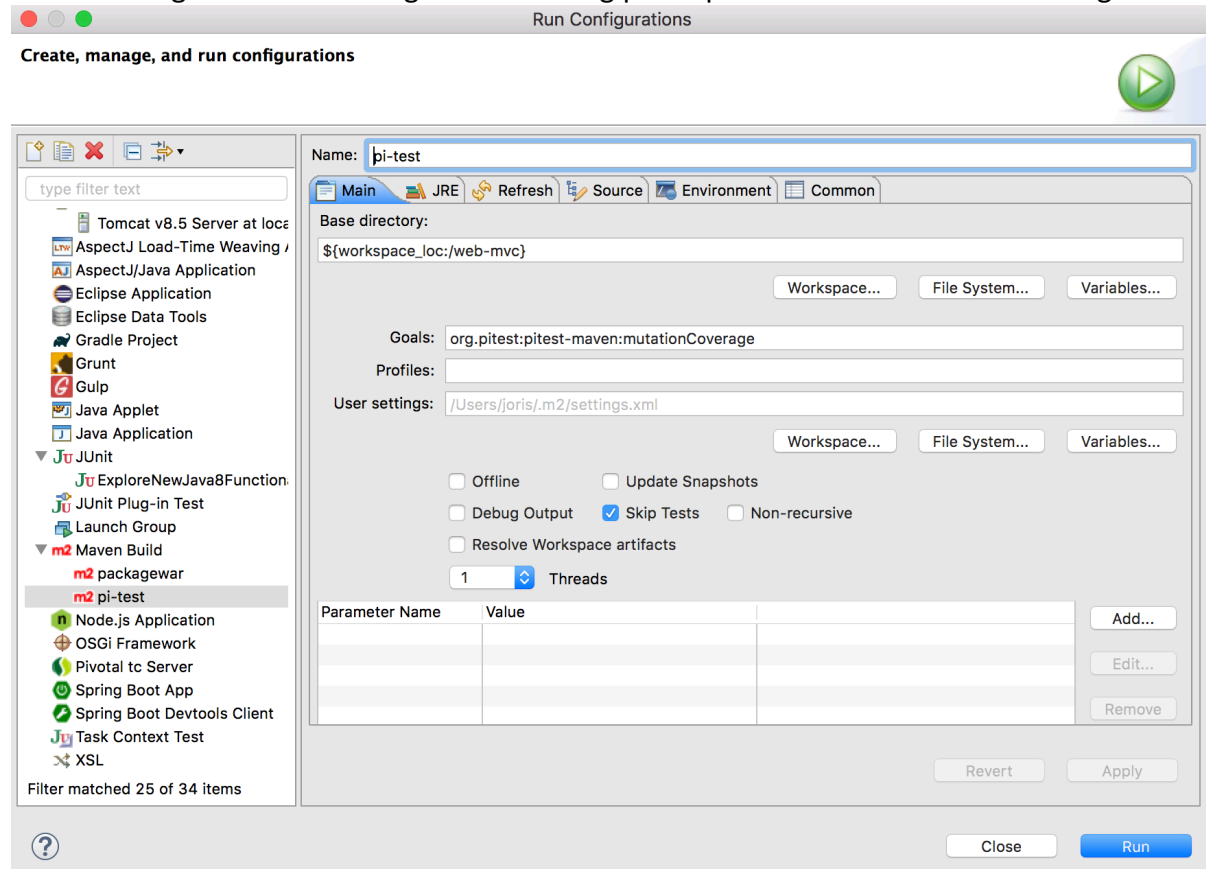
```
<build>
  <plugins>
    <plugin>
      <groupId>org.pitest</groupId>
      <artifactId>pitest-maven</artifactId>
      <version>1.2.3</version>
      <configuration>
        <targetClasses>
          <!-- Location of code to mutate- ->
          <param>web.controllers*</param>
          <param>web.services*</param>
        </targetClasses>
        <targetTests>
          <!-- Location of the test to run- ->
          <param>controller*</param>
          <param>services*</param>
        </targetTests>
      </configuration>
    </plugin>
  </plugins>
</build>
<reporting>
  <plugins>
    <plugin>
      <groupId>org.pitest</groupId>
      <artifactId>pitest-maven</artifactId>
```

```

<version>LATEST</version>
<reportSets>
  <reportSet>
    <reports>
      <report>report</report>
    </reports>
  </reportSet>
</reportSets>
</plugin>
</plugins>
</reporting>

```

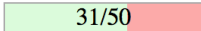
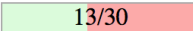
Maak vervolgens een runconfiguratie met: org.pitest:pitest-maven:mutationCoverage



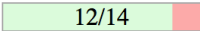
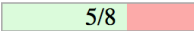
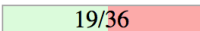
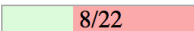
Configuratie in eclipse.

Pit Test Coverage Report

Project Summary

Number of Classes	Line Coverage	Mutation Coverage
3	62% 	43% 

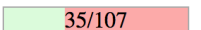
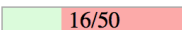
Breakdown by Package

Name	Number of Classes	Line Coverage	Mutation Coverage
com.bobsoppe.cursus.dao.impl	1	86% 	63% 
com.bobsoppe.cursus.domain	2	53% 	36% 

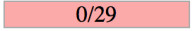
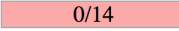
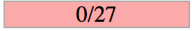
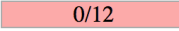
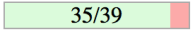
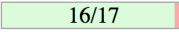
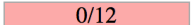
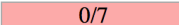
Report generated by [PIT](#) 1.1.11

Pit Test Coverage Report

Project Summary

Number of Classes	Line Coverage	Mutation Coverage
6	33% 	32% 

Breakdown by Package

Name	Number of Classes	Line Coverage	Mutation Coverage
com.bobsoppe.cursus.web.configuration	1	0% 	0% 
com.bobsoppe.cursus.web.controllers	3	0% 	0% 
com.bobsoppe.cursus.web.helpers.impl	1	90% 	94% 
com.bobsoppe.cursus.web.services.impl	1	0% 	0% 

Report generated by [PIT](#) 1.1.11

Over de persistence

In de persistence.xml van productie:

```
<property name="hibernate.hbm2ddl.auto" value="create-drop" />
```

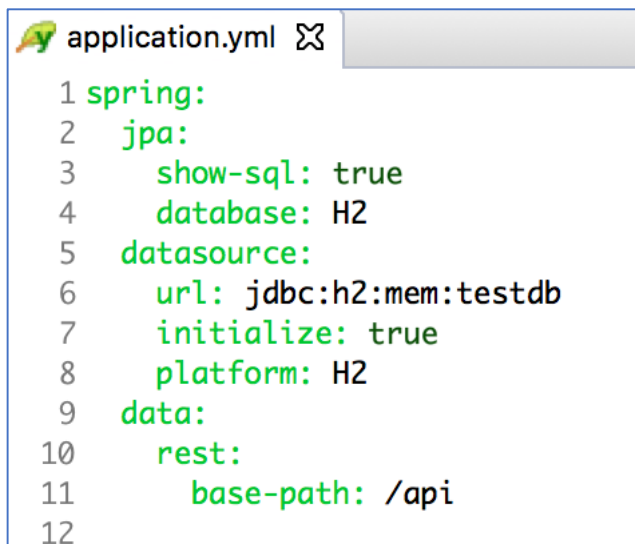
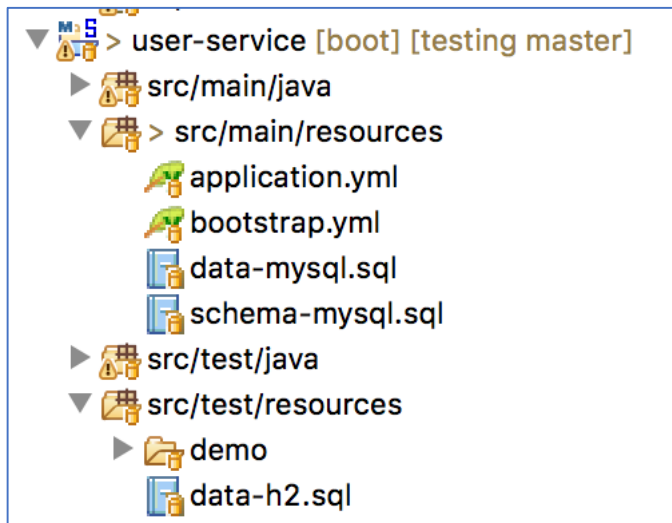
Gebruik de optie validate

```
<property name="hibernate.hbm2ddl.auto" value="validate" />
```

In de applicatie wordt een datasource geconfigureerd met usn/pwd echter usn/pwd komen nog steeds voor in persistence.xml (worden weliswaar genegeerd)-> verwarrend

Zorg voor scripts die de database opbouwen.

Persistence bij springboot :



```
spring.jpa.database-platform : String
spring.jpa.generate-ddl : boolean
spring.jpa.hibernate.ddl-auto : String
spring.jpa.hibernate.naming.implicit-strategy : String
spring.jpa.hibernate.naming.physical-strategy : String
spring.jpa.hibernate.naming.strategy : String
spring.jpa.hibernate.use-new-id-generator-mappings : boolean
spring.jpa.open-in-view : boolean
spring.jpa.properties : Map<String, String>
spring.jpa.database : org.springframework.orm.jpa.vendor.Database[DEFAULT, DB2, DERBY,
spring.jpa.show-sql : boolean
```

`spring.jpa.hibernate.ddl-auto`
[java.lang.String](#)

DDL mode. This is actually a shortcut for the "hibernate.hbm2ddl.auto" property. Default to "create-drop" when using an embedded database, "none" otherwise.

Verschillen database test/deployment

Kijk uit voor Case-(IN)-sensitive database character sets (collation.)

```
assertThat(countRowsInTable("cursist"), is(4)); ->
```

moet dan zijn Cursist

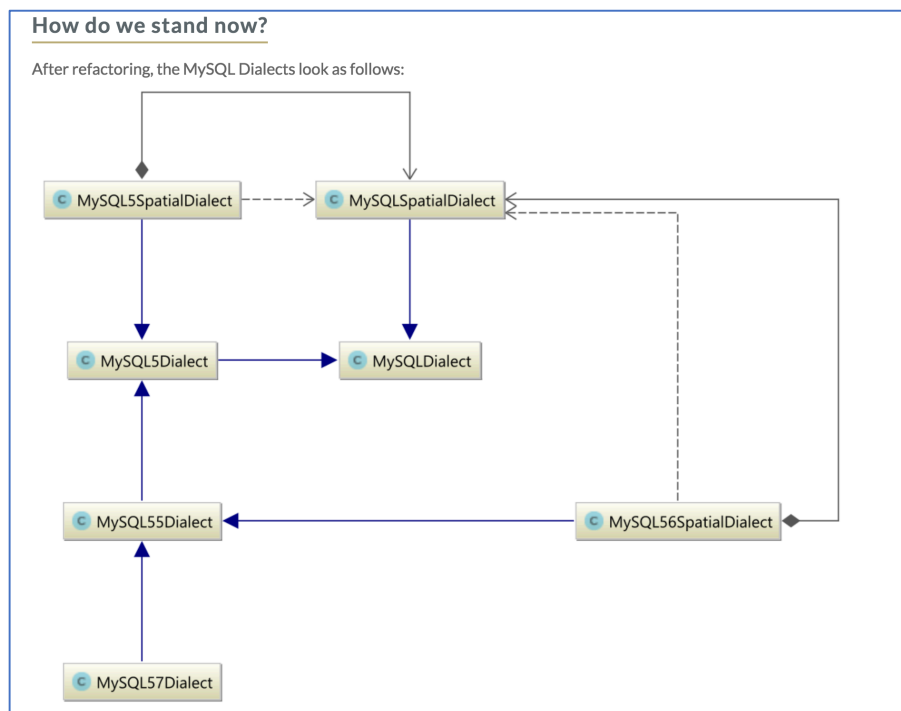
```
assertThat(countRowsInTable("Cursist"), is(4));
```

```
<property name="hibernate.dialect"
          value="org.hibernate.dialect.MySQL5Dialect" />
```

moet zijn:

```
<property name="hibernate.dialect"
          value="org.hibernate.dialect.MySQL5XDialect" />
```

Met X een van de waarden uit onderstaande table.



```
jdbcTemplate.update("ALTER TABLE cursus AUTO_INCREMENT = 1");
```

moet zijn:

```
jdbcTemplate.update("ALTER TABLE Cursus AUTO_INCREMENT = 1");
```

CodeCoverage:

parent_case (Oct 10, 2017 3:51:43 PM)				
Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
▼ datastorage	59.2 %	158	109	267
▼ src/main/java	59.2 %	158	109	267
▼ jpa.domain	49.3 %	106	109	215
▶ Cursus.java	52.6 %	70	63	133
▶ Cursist.java	43.9 %	36	46	82
▼ jpa.dao.impl	100.0 %	52	0	52
▶ CursistDaoImpl.java	100.0 %	26	0	26
▶ CursusDaoImpl.java	100.0 %	26	0	26

web (Oct 10, 2017 3:54:06 PM)				
Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
▼ web	50.6 %	306	299	605
▼ src/main/java	50.6 %	306	299	605
▼ web.controller	7.6 %	18	218	236
▶ CursusController.java	6.1 %	11	170	181
▶ CursistController.java	12.7 %	7	48	55
▼ web.business.impl	62.8 %	137	81	218
▶ CursusServiceImpl.java	58.9 %	116	81	197
▶ CursistServiceImpl.java	100.0 %	21	0	21
▼ web.configuration	100.0 %	151	0	151
▶ SpringWebConfig.java	100.0 %	151	0	151
▼ datastorage	33.7 %	90	177	267
▼ src/main/java	33.7 %	90	177	267
▼ jpa.domain	39.1 %	84	131	215
▼ jpa.dao.impl	11.5 %	6	46	52
▼ CursistDaoImpl.java	11.5 %	3	23	26
▶ CursistDaoImpl	11.5 %	3	23	26
▼ CursusDaoImpl.java	11.5 %	3	23	26
▶ CursusDaoImpl	11.5 %	3	23	26

Als je vanuit het web maar heel beperkt de Dao's aanspreekt, heb je dan niet veel te veel gebouwd?

Opvallende zaken in code

Over : catch blok

```
public int saveCursussenFromFile(MultipartFile file) {  
    List<Cursus> cursuses = null;  
  
    try {  
        cursussen = this.processCursusFile(convert(file));  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
}
```

Wil je een exception opeten? Je hoeft het nog niet te implementeren hier, laat het er daarom wel inzetten zodat op het moment dat het relevant wordt dat het gedrag niet verstopt is.

Mooie toepassing flashAttribute:

```
public String create(@ModelAttribute Cursist cursist, RedirectAttributes  
redirectAttributes) {  
    this.cursistService.saveCursist(cursist);  
  
    redirectAttributes.addFlashAttribute("message_added_cursist",  
        "De cursist: " + cursist.getFirstName() + " " +  
        cursist.getLastName() + " is succesvol toegevoegd.");  
  
    return "redirect:/cursussen";  
}
```

Namen methoden/classes/packages scheppen een verwachting :

Naam methode -> saveCourseInstanceIfUnique() -> je verwacht in code dan 2 delen

checkIfCourseInstance is unique

als code uniek is save CourseInstance

echter -> 1 blob code

Naam -> `private List<Course> checkWhichCourseInstancesAreWithinDates(List<Course> courses, LocalDate from, LocalDate to)`

- ➔ Check suggereert een Boolean maar je geeft een List terug.
- ➔ getCouseInstancesWithinPeriod
- ➔ de naam heeft het over CourseInstances, hij return een lijst met Course's

Zeg wat je doet, en doe wat je zegt.

Over naamgeving:

```
public int parseAndSaveCursussen(File fileIn){..}
```

We zien direct de delen -> waar is het overkoepelende geheel?

Maak veel mooie kleine methoden op het zelfde abstractie niveau.

De webconfiguratie:

In de pom.xml

```
<build>
  <finalName>CAS_NK</finalName>
</build>
```

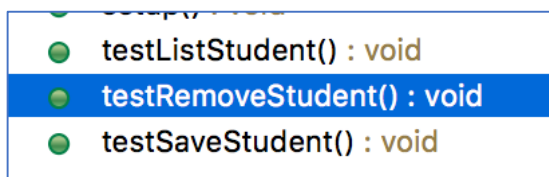
In de web.xml

```
<servlet-mapping>
  <servlet-name>spring</servlet-name>
  <url-pattern>/CASservice/*</url-pattern>
</servlet-mapping>
```

Gebruik van de Lombok library.

Lombok -> voegt veel overzicht toe -> afgestemmen met belanghebbende partijen.

Denk om naamgeving testmethodes:



●	testListStudent() : void
●	testRemoveStudent() : void
●	testSaveStudent() : void

tests -> when – scenario –outcome -> wat verwacht je?

```

private void setWeeksAndYears(int weekId, int year){
    if (weekId > 51) {
        nextWeek = 1;
        nextYear = year + 1;
        previousWeek = weekId - 1;
        previousYear = year;
    }
    else if (weekId == 1) {
        previousWeek = 52;
        previousYear = year - 1;
        nextWeek = weekId + 1;
        nextYear = year;
    }else{
        previousWeek = weekId - 1;
        previousYear = year;
        nextWeek = weekId + 1;
        nextYear = year;
    }
}

```

Goed om hier over na te denken. (hierboven)
Merk op er zijn ook Jaren met 53 weken.

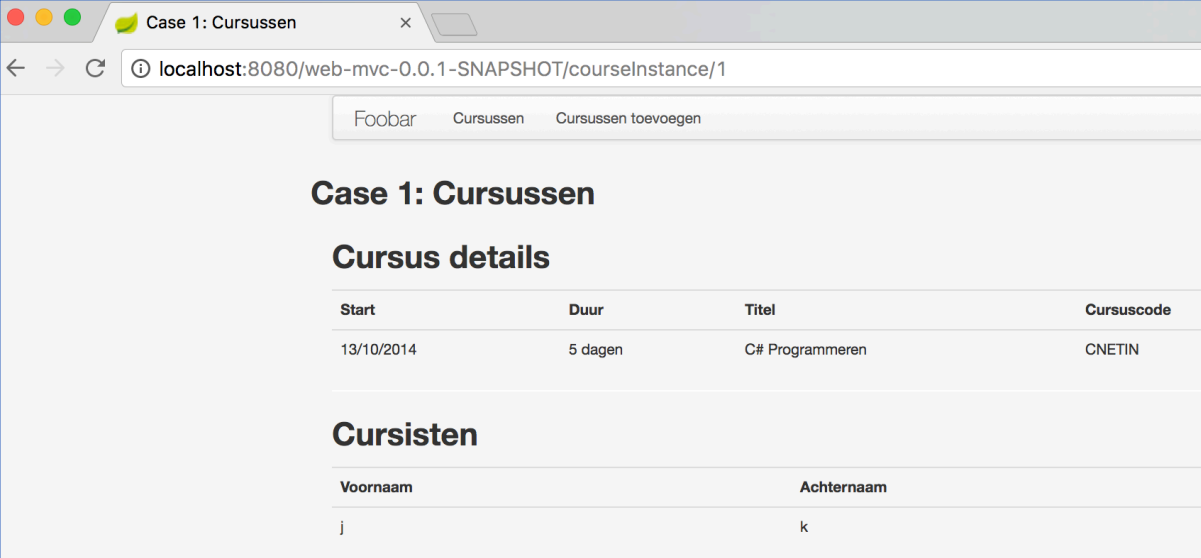
Er staat code uit gecomment in de projecten: wat is de betekenis daarvan?
Weghalen! Lever alles schoon op.

Over de implementatie van de PlanningParser

- Is naar buiten toe de public parse(file) methode niet voldoende
- De andere methodes beschrijven fasen van de verwerking van het parse proces en zouden dus private moeten zijn.
- Er bestaat geen behoefte om de onderdelen ook naar buiten aan te bieden.

```
@Test
public void importCorrectCoursesTestFileShouldPutFiveInstancesInDatabase() {
    courseService.importCoursesFromFile(getTestFile());
    Query query = entityManager.createQuery("SELECT c,i FROM Cursus as c, CursusInstance as i");
    int amountOfCourseInstancesInDb = query.getResultList().size();
    assertThat(amountOfCourseInstancesInDb, is(5));
}
```

5 Instances van Wat? Courses en Cursus is dat synonym?



The screenshot shows a web browser window with the title 'Case 1: Cursussen'. The address bar shows 'localhost:8080/web-mvc-0.0.1-SNAPSHOT/courseInstance/1'. The page has a navigation bar with 'Foobar', 'Cursussen', and 'Cursussen toevoegen'. The main content area is titled 'Case 1: Cursussen' and contains two sections: 'Cursus details' and 'Cursisten'.

Cursus details

Start	Duur	Titel	Cursuscode
13/10/2014	5 dagen	C# Programmeren	CNETIN

Cursisten

Voornaam	Achternaam
j	k

In je URL's geef je veel informative over de interne structuur weer.

Die is voor de buitenwereld zichtbaar! -> hackers

Maar ook de gebruikers -> de gebruikers zien hele andere concepten, waarom niet eenduidig?

```
<td><a th:href="@{course/{courseId}(courseId=${course.id})}">
```

Het plakken met paden is een lastige aangelegenheid omdat je dan meer kennis in je navigatie moet stoppen.

Gebruik daarom relatieve paden + Thymeleaf voor de absolute naam webapp -> waarop je geen greep hebt.

Course List

Start	Duur	Titel	Studenten
13-10-2014	5	C# Programmeren	0 Studenten
20-10-2014	5	C# Programmeren	0 Studenten
20-10-2014	2	Advanced C#	0 Studenten
13-10-2014	2	Advanced C#	0 Studenten
13-10-2014	5	C# Programmeren	0 Studenten

→ ↻ ⓘ localhost:8080/CAS_TVO/CASservice/course/fileupload

apps /manager k Kindle Dordt1Jen Dordt2Jen cnj MacGi

File to upload: Choose file No file chosen

Upload

Cursussen Studenten Upload cursus Voeg student toe

Voornaam:

Achternaam:

Toevoegen

Niet test vriendelijk!

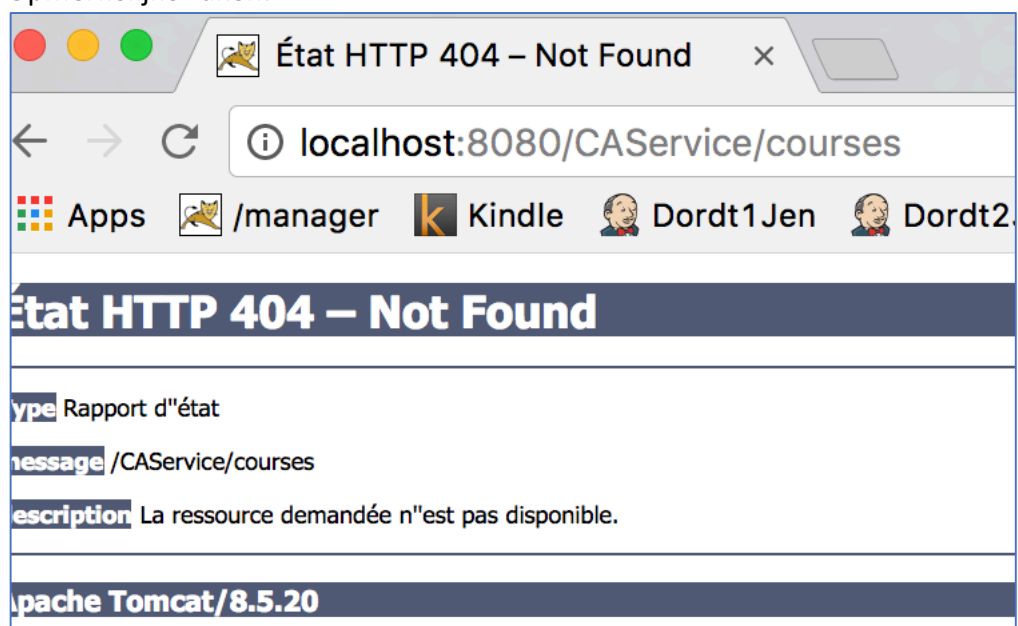
```
@Test
public void InputStreamWithOneInstanceShouldCreateOneInstance() throws Exception {

    FileInputStream fileInputStream = new FileInputStream("C:\\dev\\Intellij\\jpawebparent\\jpa21\\src\\te

    MultiPartFileParser parser = new MultiPartFileParser();
}
```

1 Controller -> beter weinig en dan ok.

Opmerkelijke zaken:



```
@RequestMapping(value = "/courses")
public String viewCourseList(Model model) {
    model.addAttribute("courses", courseRepository.findAll());

    return "course";
}
```

Opvallende naam: courses gaan er een, courses gaan op het attribuut en je verwijst door naar course ?

```

@Controller
@RequestMapping("/course")
public class CourseController {

    @PersistenceContext
    EntityManager em;

    @Autowired
    private CourseInstanceRepository courseInstanceRepository;
    @Autowired
    private CourseRepository courseRepository;

    @Autowired
    private MultiPartFileParser parser;

    @RequestMapping(value = "/index")
    public String index(Model model) {

```

elke requestmapping in deze controller wordt nu ge pre mapped met course -> dus index wordt /course/index

```

<title>title</title>
<link rel="stylesheet" type="text/css" href="/css/bootstrap.min.css" media="screen"
      th:href="@{../../css/bootstrap.css}"/>
</head>

```

@ -> geeft je de naam van de webapplicatie, -> daar plak je dan relatief pad aan.

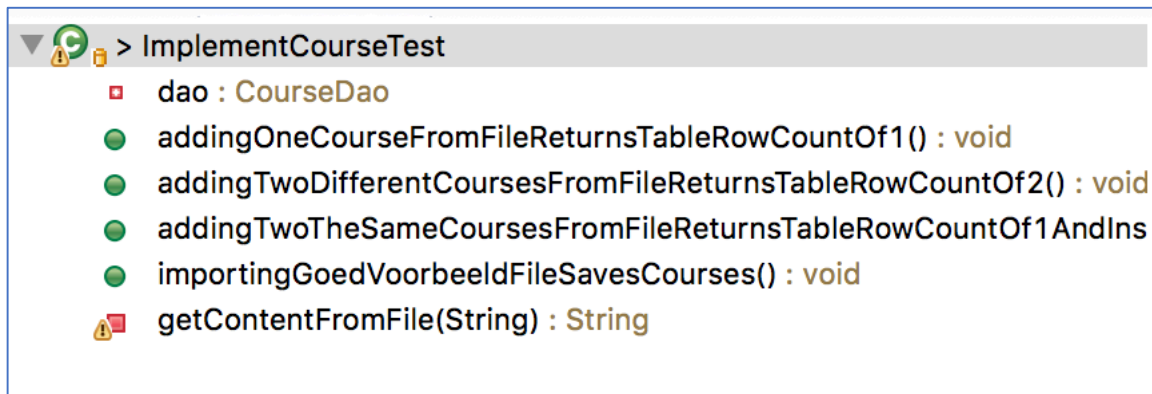
```

<div th:insert=~{fragments/footer :: footer}></div>
<script src="/js/jquery-3.2.1.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.11.0/umd/popper.min.js"
      integrity="sha384-b/U6ypiBEHp0f/4+1nzFpr53nxSS+GLCkfwBdFNTxtclqgenISfwAzpKaMNFNmj4"
      crossorigin="anonymous"></script>
<script src="/js/bootstrap.js"></script>

```

in elke thymeleaf file neem je dit weer op. -> gebruik gemeenschappelijke header.

course -> Course en het loopt weer.
Wat voor setting heb jij gebruikt?



Mooie verzameling testen: Naam test en de testmethode namen lijken niet helemaal op elkaar afgestemd.

Dit lijkt meer een sort integratie test terwijl de naam iets met Courses suggereert.

Notes

Tijdens het ontwerpen van de applicatie liep ik herhaaldelijk tegen problemen aan tijdens het testen. Laat tijdens de ontwikkelfase kreeg ik het eindelijk werkend, maar helaas was dit toen al aan de late kant en had ik nog maar weinig tijd om mijn klassen te testen. In principe liggen alle bouwstenen klaar om alles goed te testen, maar de tijd om deze goed uit werken ontbrak.

Je levert je helemaal over aan de andere kant -> Hoe ga je met zo'n situatie om?

```
<build>
  <finalName>CAS_TVDS</finalName>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
      <version>1.5.7.RELEASE</version>
    </plugin>
  </plugins>
</build>
```

Springboot?

```

<dependency>
  <groupId>org.projectlombok</groupId>
  <artifactId>lombok</artifactId>
  <version>1.16.18</version>
</dependency>
<dependency>
  <groupId>commons-io</groupId>
  <artifactId>commons-io</artifactId>
  <version>2.4</version>
</dependency>
<dependency>
  <groupId>commons-fileupload</groupId>
  <artifactId>commons-fileupload</artifactId>
  <version>RELEASE</version>
</dependency>
<dependency>
  <groupId>org.thymeleaf</groupId>
  <artifactId>thymeleaf-spring4</artifactId>
  <version>RELEASE</version>
</dependency>
<dependency>
  <groupId>org.mockito</groupId>
  <artifactId>mockito-core</artifactId>
  <version>RELEASE</version>
</dependency>

```

Veel dependencies worden helemaal niet gebruikt. Waarom dan opnemen?


```

@Test
public void testFindCursus() {
    cursusDAO.saveCursus(new Cursus("IIUIA", "Karten voor beginners 2", 12));
    Cursus result = cursusDAO.findCursus("IIUIA");

    assertThat(result.getName(), is("Karten voor beginners 2"));
}

@Test
public void testListCursisten() {
    List<Cursist> results = cursistDAO.findAll();

    //assertThat(results.size(), is(3));
}

@Test
public void testRemoveCursist() {
    //cursistDAO.removeCursist(cursistDAO.findAll().get(0).getId());
    //entityManager.flush();

    //assertThat(countRowsInTable("cursist"), is(2));
}

```

De test slagen maar ...

```

Cursus currentCursus = null;
CursusInstantie currentCursusInstantie = null;
int succesFullCursusImports = 0;
int succesFullCursusInstantieImports = 0;
BufferedReader br;
try {
    br = new BufferedReader(new InputStreamReader(file.getInputStream()));

    String currentLine = "";
    int currentLineNumber = 0;
    while ((currentLine = br.readLine()) != null) {
        currentLineNumber++;
        if (currentLine.contains("Titel: ")) {
            currentCursus = new Cursus();
            currentCursusInstantie = new CursusInstantie();

            String title = currentLine.substring(currentLine.indexOf("Titel: ").replace(" ", ""));
            currentCursus.setName(title);
        } else if (currentLine.contains("Cursuscode: ")) {
            String code = currentLine.substring(currentLine.indexOf("Cursuscode: ").replace(" ", ""));
            currentCursus.setCursusCode(code);
        } else if (currentLine.contains("Duur: ")) {
            int duration = Integer.parseInt(currentLine.replaceAll("\\D+", ""));
            currentCursus.setLength(duration);
        } else if (currentLine.contains("Startdatum: ")) {
            String startDateString = currentLine.substring(currentLine.indexOf("Startdatum: ").replace(" ", ""));
            SimpleDateFormat formatter = new SimpleDateFormat("dd/MM/yyyy");
            try {
                currentCursusInstantie.setDate(formatter.parse(startDateString));
            } catch (ParseException e) {
                model.addAttribute("message", "ERROR: Datum op regel " + currentLineNumber + " is ongeldig");
            }
        }

        if (cursusService.getCursusByCursusCode(currentCursus.getCursusCode()) == null) {
            cursusService.saveCursus(currentCursus);
            succesFullCursusImports++;
        }

        currentCursusInstantie.setCursus(cursusService.getCursusByCursusCode(currentCursus.getCursusCode()));

        if (cursusInstantieService.savePlannedCursus(currentCursusInstantie) != null) {
            succesFullCursusInstantieImports++;
        } else {
            model.addAttribute("message", "ERROR: Cursusinstantie is niet geïmporteerd omdat regel " + currentLineNumber + " ongeldig is");
        }
    }

    /*for (CursusInstantie tempPlannedCursus : cursusInstantieService.findPlannedCursusByCursussenCode(currentCursusInstantie.getCursus().getCursusCode())) {
        */
}

```

Wat zou je met zo'n brok aan code willen doen?