



samen sterk voor werk

JAVA

JSP / Servlets Takenbundel



Inhoudsopgave

1	TAKEN	5
1.1	Frituur Frida	5
1.2	Sluitingsdagen	5
1.3	Sluitingsdagen 2	5
1.4	Sluitingsdagen 3	5
1.5	Sluitingsdagen 4	5
1.6	Adres	5
1.7	Sauzen	6
1.8	Initialisatieparameters	6
1.9	Servlet context Initialisatieparameters	6
1.10	MVC	6
1.11	Ingrediënten	6
1.12	Sauzen verwijderen	7
1.13	Meisjes jongens	7
1.13.1	Tip	7
1.14	Zoek de friet	7
1.14.1	Tip	8
1.15	Saus raden	8
1.16	Statistiek	9
1.17	Statistiek 2	9
1.18	Database	9
1.19	Vertalen	9
1.20	Menu	9
1.21	Gastenboek	10
1.22	Gastenboekbeheer	10
2	VOORBEELDOPLOSSINGEN	11
2.1	Frituur Frida	11
2.1.1	index.html	11
2.1.2	default.css	11

2.2	Sluitingsdagen	11
2.2.1	index.html	11
2.2.2	IndexServlet	11
2.3	Sluitingsdagen 2	12
2.3.1	IndexServlet	12
2.3.2	index.jsp	12
2.4	Sluitingsdagen 3	12
2.4.1	IndexServlet	12
2.4.2	WEB-INF/JSP/index.jsp	12
2.5	Sluitingsdagen 4	13
2.5.1	IndexServlet: ongewijzigd	13
2.5.2	index.jsp	13
2.6	Adres	13
2.6.1	Gemeente	13
2.6.2	Adres	13
2.6.3	IndexServlet: extra opdracht in de method doGet	13
2.6.4	index.jsp	13
2.7	Sauzen	13
2.7.1	Head.jsp: zelfde als in theorie	13
2.7.2	Saus	13
2.7.3	SauzenServlet	14
2.7.4	/WEB-INF/JSP/sauzen.jsp	14
2.8	Initialisatieparameters	14
2.8.1	IndexServlet	14
2.8.2	web.xml	15
2.8.3	index.jsp	15
2.9	Servlet context initialisatieparameters	15
2.9.1	web.xml	15
2.9.2	IndexServlet: gewijzigde opdracht in de method doGet	15
2.10	MVC	15
2.10.1	SausRepository	15
2.10.2	SauzenServlet	16
2.11	Ingrediënten	16
2.11.1	SausRepository: extra code	16
2.11.2	IngredientenServlet	16
2.11.3	/WEB-INF/JSP/ingredienten.jsp	17
2.11.4	default.css: uitbreiding	17
2.12	Sauzen	18
2.12.1	/WEB-INF/JSP/sauzen.jsp	18
2.12.2	SausRepository: extra code	18
2.12.3	SauzenVerwijderenServlet	18
2.13	Meisjes jongens	19
2.13.1	MeisjesJongensServlet	19

2.13.2	WEB-INF/JSP/meisjesjongens.jsp	19
2.13.3	default.css: uitbreiding	19
2.14	Zoek de friet	20
2.14.1	Deur	20
2.14.2	ZoekDeFrietSpel	20
2.14.3	ZoekDeFrietServlet	20
2.14.4	WEB-INF/JSP/zoekdefriet.jsp	21
2.15	Saus raden	22
2.15.1	SausRadenSpel	22
2.15.2	SausRadenServlet	23
2.15.3	WEB-INF/JSP/sausraden.jsp	23
2.16	Statistiek	24
2.16.1	RequestStatistiekListener	24
2.16.2	StatistiekServlet	25
2.16.3	WEB-INF/JSP/statistiek.jsp	25
2.17	Statistiek 2	26
2.17.1	RequestStatistiekFilter	26
2.18	Database	27
2.18.1	Databasestructuur	27
2.18.2	META-INF/context.xml	27
2.18.3	RepositoryException	27
2.18.4	AbstractRepository	27
2.18.5	SausRepository	27
2.18.6	Extra code in Servlets die een variabele sausRepository hebben	29
2.19	Vertalen	29
2.19.1	resourceBundles/teksten.properties	29
2.19.2	resourceBundles/teksten_en.properties	29
2.19.3	index.jsp	29
2.20	Menu	30
2.20.1	/WEB-INF/tags/vdab/menu.tag	30
2.20.2	/WEB-INF/tlds/vdab.tld	30
2.20.3	default.css: uitbreiding	30
2.20.4	JSP's	30
2.21	Gastenboek	30
2.21.1	Nieuwe table in de database	30
2.21.2	GastenBoekEntry	31
2.21.3	GastenBoekServlet	31
2.21.4	GastenboekRepository	32
2.21.5	menu.tag: extra regel	33
2.21.6	default.css: aangepaste regel	33
2.21.7	gastenboek.jsp	33
2.22	Gastenboekbeheer	34
2.22.1	web.xml: extra regels	34

2.22.2	InloggenServlet	34
2.22.3	Inloggen.jsp	34
2.22.4	gastenboek.jsp	35
2.22.5	GastenboekServlet	35
2.22.6	GastenboekRepository	35

3	COLOFON	37
----------	----------------	-----------

1 TAKEN

1.1 Frituur Frida

Je maakt een project voor een website met één statische HTML welkompagina.

- Deze pagina bevat een <h1> element met de tekst Frituur Frida
- Je geeft deze tekst een kleur met een CSS bestand
- Deze pagina gebruikt favicon.ico uit het takenmateriaal

1.2 Sluitingsdagen

Frituur Frida is gesloten op maandag en donderdag.

Je voegt een servlet toe aan de website en je stelt deze in als welkompagina.

Als je naar de website surft op maandag of donderdag, toon je de tekst Vandaag zijn we gesloten.

Op andere dagen toon je de tekst Vandaag zijn we open.

1.3 Sluitingsdagen 2

Frituur Frida is gesloten op maandag en donderdag.

Je voegt een JSP toe aan de website en je stelt deze in als welkompagina.

Als je naar de website surft op maandag of donderdag, toon je de tekst Vandaag zijn we gesloten.

Op andere dagen toon je de tekst Vandaag zijn we open.

1.4 Sluitingsdagen 3

Frituur Frida is gesloten op maandag en donderdag.

Je verwerkt een request naar de welkompagina eerst in een servlet.

Deze servlet maakt een request attribuut met de naam openGesloten. De inhoud van dit request attribuut is gesloten (op maandag en donderdag) of open (op andere dagen).

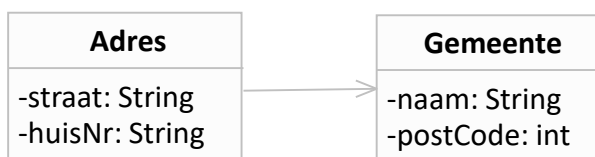
De servlet geeft daarna de request door aan een JSP. Deze JSP bevindt zich in WEB-INF/JSP. Deze JSP gebruikt EL om met het request attribuut openGesloten de tekst Vandaag zijn we gesloten of Vandaag zijn we open te tonen.

1.5 Sluitingsdagen 4

- Op de dagen dat frituur Frida gesloten is, toon je onder de tekst Vandaag zijn we gesloten de afbeelding gesloten.png (uit het materiaal bij deze cursus.)
- Op andere dagen toon je onder de tekst Vandaag zijn we open de afbeelding open.png uit het materiaal bij deze cursus.

1.6 Adres

Je voegt aan het project voor Frituur Frida volgende classes toe



In de servlet die hoort bij de welkompagina

- Maak je een Adres object en een bijbehorend Gemeente object
- Vul je de attributen van deze objecten met zelf verzonden adresgegevens van frituur Frida
- Maak je van dit adres object een request attribuut

In de JSP die hoort bij de welkompagina toon je het volledige adres van frituur Frida aan de hand van dit request attribuut.

1.7 Sauzen

Je maakt een include file `head.jsp` (zoals in de theorie) en past die toe.

Je voegt aan het project voor Frituur Frida een class `Saus` toe.

Saus
-nummer: long
-naam: String
-ingredienten: String[*]

Je voegt aan het project een nieuwe servlet toe, geassocieerd met de URL `/sauzen`

Bij een GET request naar deze servlet, doe je het volgende:

- Je maakt een `List<Saus>` en je vult die met sauzen met volgende namen: cocktail, mayonaise, mosterd, tartare, vinaigrette.
Je verzint zelf ids en ingrediënten van deze sauzen.
- Je geeft deze verzameling sauzen door aan een JSP.
- Je toont in deze JSP de naam en de ingrediënten van deze sauzen.
Als een saus meerdere ingrediënten heeft, scheid je deze ingrediënten met een komma.
- Je toont bij iedere saus ook een afbeelding van die saus.
Je vindt deze afbeeldingen in het materiaal bij deze cursus.

Tip:

vergeet niet de jar bestanden die bij JSTL horen toe te voegen aan je project.

1.8 Initialisatieparameters

Je maakt een initialisatieparameter met het telefoonnummer van de helpdesk in `web.xml`.

Je toont dit nummer in de welkompagina.

1.9 Servlet context Initialisatieparameters

Je maakt van het telefoonnummer van de helpdesk een servlet context Initialisatieparameter.

1.10 MVC

Je maakt een package `be.vdab.repositories`. Je maakt daarin een class `SausRepository`.

Deze class bevat een static `private Map<Long,Saus>`.

- De key van iedere entry is het id van een saus.
- De value van iedere entry is een `Saus` object.

Je vult deze map met de sauzen die je tot nu in de `SauzenServlet` maakte.

Je voegt aan deze class een method `findAll()` toe. Deze method geeft een `List<Saus>` terug, met alle `Saus` objecten uit de static `private Map`.

Je roept vanuit de `SauzenServlet` deze method op en geeft de sauzen door aan de JSP die bij deze servlet hoort. Je beeldt de sauzen af in deze JSP.

1.11 Ingrediënten

Je toont de gebruiker een form waar hij een ingrediënt tikt.

Wanneer de gebruiker de form submit, doe je het volgende:

- Als de gebruiker het vak niet invulde, toon je de foutmelding verplicht
- Anders toon je de namen van de sauzen die het ingetikte ingrediënt bevatten

1.12 Sauzen verwijderen

Je breidt de pagina uit waarin je alle sauzen toont:

- Je toont voor de naam van iedere saus een vinkje.
- Je toont onder alle sauzen een knop met de tekst Aangevinkte sauzen verwijderen

Als de gebruiker op deze knop klikt, verwijder je de aangevinkte sauzen.

Je toont daarna deze pagina opnieuw.

Als de gebruiker dan een 'refresh' doet, krijgt hij geen waarschuwingen van de browser.

1.13 Meisjes jongens

Je maakt in de website Frituur Frida een nieuwe pagina

- Als de gebruiker de knop meisjes kiest, onthoud je deze keuze in een cookie en toon je dezelfde pagina gedurende 30 minuten met een roze achtergrond en rode tekstkleur.
- Als de gebruiker de knop jongens kiest, onthoud je deze keuze in een cookie en toon je dezelfde pagina gedurende 30 minuten met een lichtblauwe achtergrond en blauwe tekstkleur.



1.13.1Tip

Je kan de twee knoppen uitschrijven in één HTML form

```
<form method='post'>
  <input type='submit' name='meisjesjongens' value='meisjes'>
  <input type='submit' name='meisjesjongens' value='jongens'>
</form>
```

Als de gebruiker één van deze knoppen aanklikt, submit hij de form.

In de doPost method, van de servlet die deze request verwerkt, kan je de request parameter meisjesjongens opvragen: request.getParameter("meisjesjongens");

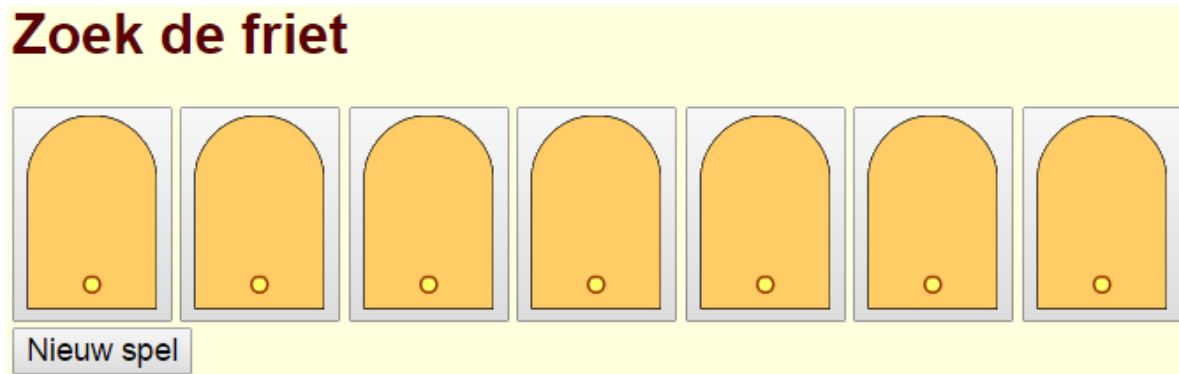
Als het resultaat gelijk is aan "meisjes" heeft de gebruiker de knop meisjes aangeklikt.

Als het resultaat gelijk is aan "jongens" heeft de gebruiker de knop jongens aangeklikt.

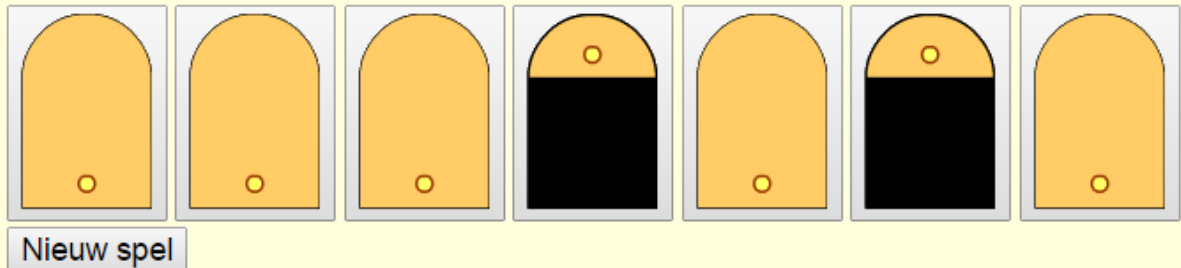
1.14 Zoek de friet

De gebruiker ziet in een pagina zeven gesloten deuren.

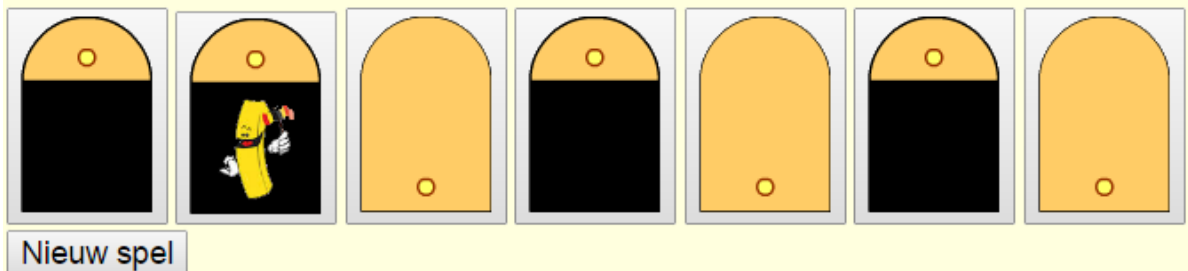
Achter één van de deuren bevindt zich een friet. De gebruiker zoekt deze deur. De gebruiker kan de deuren één per één openen door op een deur te klikken. Eenmaal geopend blijft de deur open.



Zoek de friet



Zoek de friet



Als de gebruiker `Nieuw spel` kiest, maak je een nieuw spel met zeven gesloten deuren.

Je gebruikt de afbeeldingen `deuropen.png`, `deurtoe.png` en `gevonden.png` bij deze cursus.

1.14.1 Tip

Je maakt met volgende HTML code een afbeelding die een request parameter met de naam `volgnummer` en de waarde 3 doorgeeft bij het submitten van de form die de afbeelding bevat

```
<button type='submit' name='volgnummer' value='3'>
  <img src='<c:url value="/images/deurtoe.png"/>' alt='deur toe'>
</button>
```

1.15 Saus raden

Je maakt een nieuwe pagina waarin de gebruiker de naam van één van de sauzen van frituur Frida raadt.

- Je toont naast `Te raden saus:` evenveel puntjes als de saus die de gebruiker moet raden. Je kiest de saus willekeurig uit de in-memory sauzen database.
- De gebruiker kan naast `letter:` een letter intikken en op `Raden` klikken. De gebruiker ziet daarna dezelfde pagina.
 - Als de ingetikte letter voorkomt in de te raden saus, toon je op de plaats van die letter geen puntje, maar de letter. Als de gebruiker bijvoorbeeld de letter *t* tikte en de te raden saus is *cocktail*, toon je `Te raden saus:t...`

- Als de ingetikte letter niet voorkomt in de te raden saus, verhoog je een teller met het aantal verkeerde pogingen. Je toont deze teller aan de gebruiker via één van de afbeeldingen 0.png ... 9.png bij deze cursus. Als de gebruiker drie verkeerde pogingen deed, toon je bijvoorbeeld 3.png:

- Als de gebruiker 10 verkeerde pogingen deed, verliest hij het spel en toon je volgende tekst: U bent verloren, de saus was cocktail
- Als de gebruiker de saus helemaal raadt en nog geen 10 verkeerde pogingen deed, wint hij het spel en toon je volgende tekst: U bent gewonnen, de saus was cocktail
- De gebruiker kan het spel herbeginnen met de knop Nieuw spel
Jij kiest dan terug een willekeurige saus voor dit nieuwe spel.

1.16 Statistiek

Je houdt met een listener bij hoeveel requests elke URL van de website Frituur Frida verwerkt. Je houdt geen statistieken bij voor URL's die verwijzen naar afbeeldingen, CSS bestanden of JavaScript bestanden.

Je maakt een extra pagina waarmee de gebruiker deze statistieken kan raadplegen.

1.17 Statistiek 2

Je vervangt de listener uit de taak Statistiek door een filter.

1.18 Database

Je maakt een database voor frituur Frida, met één of meerdere tables naar eigen smaak.

Je voegt in die table(s) ook enkele voorbeeldrecords toe.

Je gebruikt deze database in de repository class van frituur Frida.

Je gebruikt een DataSource.

1.19 Vertalen

Als één van de items van de request header Accept-Language als taal en bevat, toon je de beginpagina van Frituur Frida in het Engels.

Je gebruikt dan de afbeelding gesloten_en.png of open_en.png

Anders toon je de beginpagina in het Nederlands.

Je gebruikt dan de afbeelding gesloten.png of open.png

1.20 Menu

Je maakt een custom tag die de menu voorstelt van de website Frituur Frida.

Je neemt deze custom tag op in elke pagina van de website.

Je hoeft geen head tag te maken zoals in de theorie.

1.21 Gastenboek

Je voegt aan de website voor frituur Frida een gastenboek toe.

Als de gebruiker de pagina opent, ziet hij de berichten die gebruikers al aan het gastenboek toevoegden, in omgekeerde chronologische volgorde.



Gastenboek

[Toevoegen](#)

28-9-15 Jeanine
Frida heeft de lekkerste frikandellen

28-9-15 Jean
Honger, op naar Frida, smullen maar !

Je voegt zelf een table toe aan de database om de gastenboekdata bij te houden.

Als de gebruiker [Toevoegen](#) kiest, ziet hij de pagina opnieuw, maar kan hij een eigen bericht toevoegen. Hierbij zijn beide vakken verplicht in te tikken.



Gastenboek

Naam:

Bericht:

28-9-15 Jeanine
Frida heeft de lekkerste frikandellen

28-9-15 Jean
Honger, op naar Frida, smullen maar !

1.22 Gastenboekbeheer

Als de beheerder van de website surft naar de URL `inloggen.htm` binnen de website, vraag je zijn paswoord. Als hij dit correct kan intikken toon je de pagina met het gastenboek.

Je bewaart het paswoord van de beheerder in `web.xml`.

Je voegt aan `menu.tag` geen item toe voor de inlogpagina:
enkel de beheerder kent de URL `inloggen.htm`

Een ingelogde beheerder ziet in het gastenboek een knop `Uitloggen`.
Hij blijft ingelogd tot hij deze knop aanklikt.

Een ingelogde beheerder ziet in het gastenboek bij ieder bericht een checkbox.

Een ingelogde beheerder ziet in het gastenboek een knop `Verwijderen`.

Als hij deze knop aanklikt, verwijder jij alle aangevinkte berichten.

2 VOORBEELDOPLOSSINGEN

2.1 Frituur Frida

2.1.1 index.html

```
<!doctype html>
<html lang='nl'>
  <head>
    <meta charset='UTF-8'>
    <title>Frituur Frida</title>
    <link rel='icon' href='images/favicon.ico'>
    <meta name='viewport' content='width=device-width,initial-scale=1'>
    <link rel='stylesheet' href='styles/default.css'>
  </head>
  <body>
    <h1>Frituur Frida</h1>
  </body>
</html>
```

2.1.2 default.css

```
body {
  background-color: #C0BD92;
  font-family: sans-serif;
}
h1 {
  color: red;
}
```

2.2 Sluitingsdagen

2.2.1 index.html

Verdwijnt

2.2.2 IndexServlet

```
package be.vdab.servlets;
// enkele imports ...
@WebServlet("/index.htm")
public class IndexServlet extends HttpServlet {
  private static final long serialVersionUID = 1L;
  @Override
  protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    PrintWriter out = response.getWriter();
    out.println("<!doctype html>");
    out.println("<html lang='nl'><head>");
    out.println("<title>Frituur Frida</title></head>");
    out.println("<body><h1>Vandaag zijn we ");
    LocalDate vandaag = LocalDate.now();
    DayOfWeek weekdag = vandaag.getDayOfWeek();
    out.print(weekdag == DayOfWeek.MONDAY || weekdag == DayOfWeek.THURSDAY ?
      "gesloten" : "open");
    out.println("</h1></body></html>");
  }
}
```

2.3 Sluitingsdagen 2

2.3.1 IndexServlet

```
@WebServlet("oud.htm")
```

2.3.2 index.jsp

```
<%@page contentType='text/html' pageEncoding='UTF-8' session='false'%>
<%@page import='java.time.DayOfWeek'%>
<%@page import='java.time.LocalDate'%>
<!doctype html>
<html lang='nl'>
<head><title>Frituur Frida</title>
<link rel='icon' href='images/favicon.ico'>
<meta name='viewport' content='width=device-width,initial-scale=1'>
<link rel="stylesheet" href="styles/default.css"></head>
<body>
<h1>Vandaag zijn we
<%
LocalDate vandaag = LocalDate.now();
DayOfWeek weekdag = vandaag.getDayOfWeek();
out.print(weekdag == DayOfWeek.MONDAY || weekdag == DayOfWeek.THURSDAY ?
    "gesloten" : "open");
%>
</h1>
</body>
</html>
```

2.4 Sluitingsdagen 3

2.4.1 IndexServlet

```
package be.vdab.servlets;
// enkele imports ...
@WebServlet("/index.htm")
public class IndexServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    private static final String VIEW = "/WEB-INF/JSP/index.jsp";
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        LocalDate vandaag = LocalDate.now();
        DayOfWeek weekdag = vandaag.getDayOfWeek();
        request.setAttribute("openGesloten",
            weekdag == DayOfWeek.MONDAY || weekdag == DayOfWeek.THURSDAY ?
                "gesloten" :
                "open");
        RequestDispatcher dispatcher = request.getRequestDispatcher(VIEW);
        dispatcher.forward(request, response);
    }
}
```

2.4.2 WEB-INF/JSP/index.jsp

```
<%@page contentType='text/html' pageEncoding='UTF-8' session='false'%>
<!doctype html>
<html lang="nl">
    <head><title>Frituur Frida</title>
    <link rel='icon' href='images/favicon.ico'>
    <meta name='viewport' content='width=device-width,initial-scale=1'>
    <link rel='stylesheet' href='styles/default.css'></head>
    <body><h1>Vandaag zijn we ${openGesloten}</h1></body>
</html>
```

2.5 Sluitingsdagen 4

2.5.1 IndexServlet: ongewijzigd

2.5.2 index.jsp

Nieuwe regel voor </body>

```

```

2.6 Adres

2.6.1 Gemeente

```
package be.vdab.entities;
public class Gemeente {
    private String naam;
    private int postcode;
    // Je maakt zelf getters en setters voor deze private variabelen met je IDE
    // Je maakt zelf een default en een geparametriseerde constructor met je IDE
}
```

2.6.2 Adres

```
package be.vdab.entities;
public class Adres {
    private String straat;
    private String huisNr;
    private Gemeente gemeente;
    // Je maakt zelf getters en setters voor deze private variabelen met je IDE
    // Je maakt zelf een default en een geparametriseerde constructor met je IDE
}
```

2.6.3 IndexServlet: extra opdracht in de method doGet

```
request.setAttribute("adres", new Adres("Grote markt", "7",
    new Gemeente(1000, "Brussel")));
```

2.6.4 index.jsp

```
<h2>Adres</h2>
${adres.straat} ${adres.huisNr}<br>
${adres.gemeente.postcode} ${adres.gemeente.naam}
```

2.7 Sauzen

2.7.1 Head.jsp: zelfde als in theorie

2.7.2 Saus

```
package be.vdab.entities;
// enkele imports ...
public class Saus {
    private long id;
    private String naam;
    private List<String> ingredienten = new ArrayList<>();
    public Saus(long id, String naam, Iterable<String> ingredienten) {
        // elke array, List en Set implementeert de interface Iterable
        this.id = id;
        this.naam = naam;
        this.ingredienten.addAll(ingredienten);
    }
}

public long getId() {
    return id;
}

public String getNaam() {
    return naam;
}
```

```

    public List<String> getIngredienten() {
        return ingredienten;
    }
    public void addIngredient(String ingredient) {
        ingredienten.add(ingredient);
    }
}

```

2.7.3 SauzenServlet

```

package be.vdab.servlets;
// enkele imports ...
@WebServlet("/sauzen.htm")
public class SauzenServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    private static final String VIEW = "/WEB-INF/JSP/sauzen.jsp";
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        List<Saus> sauzen = Arrays.asList(
            new Saus(3L, "cocktail", Arrays.asList("mayonaise", "ketchup", "cognac")),
            new Saus(6L, "mayonaise", Arrays.asList("ei", "mosterd")),
            new Saus(7L, "mosterd", Arrays.asList("mosterd", "azijn", "witte wijn")),
            new Saus(12L, "tartare", Arrays.asList("mayonaise", "augurk", "tabasco")),
            new Saus(44L, "vinaigrette", Arrays.asList("olijfolie", "mosterd", "azijn")));
        request.setAttribute("sauzen", sauzen);
        request.getRequestDispatcher(VIEW).forward(request, response);
    }
}

```

2.7.4 /WEB-INF/JSP/sauzen.jsp

```

<%@page contentType='text/html' pageEncoding='UTF-8' session='false'%>
<%@taglib prefix='c' uri='http://java.sun.com/jsp/jstl/core'%>
<!doctype html>
<html lang='nl'>
    <head>
        <c:import url='head.jsp'><c:param name='title' value='Sauzen'/></c:import>
    </head>
    <body>
        <h1>Sauzen</h1>
        <c:forEach var='saus' items='${sauzen}'>
            <h2>${saus.naam}</h2>
            <img src='images/${saus.naam}.png' alt='${saus.naam}'> ingrediënten:
            <c:forEach var='ingredient' items='${saus.ingredienten}'
                varStatus='status'>
                ${ingredient}<c:if test='${not status.last}'>, </c:if>
            </c:forEach>
        </c:forEach>
    </body>
</html>

```

2.8 Initialisatieparameters

2.8.1 IndexServlet

```
@WebServlet(urlPatterns = "/index.htm", name = "indexservlet")
```

Extra code in de method `doGet`

```

request.setAttribute("telefoonNummerHelpDesk",
    this.getInitParameter("telefoonNummerHelpDesk"));

```

2.8.2 web.xml

```
<servlet>
  <servlet-name>indexServlet</servlet-name>
  <servlet-class>be.vdab.servlets.IndexServlet</servlet-class>
  <init-param>
    <param-name>telefoonNummerHelpDesk</param-name>
    <param-value>01/234567</param-value>
  </init-param>
</servlet>
```

2.8.3 index.jsp

```
<div>Telefoonnummer helpdesk:
<a
href="tel:>${telefoonNummerHelpDesk.replace('/', '')}">${telefoonNummerHelpDesk}
</a></div>
```

Als de gebruiker een hyperlink aanklikt, waarvan het href attribuut begint met tel; belt je tablet of smartphone het nummer vermeld in href.

Dit nummer moet beginnen met + en mag geen / tekens bevatten.

2.9 Servlet context initialisatieparameters

2.9.1 web.xml

```
<context-param>
  <param-name>telefoonNummerHelpDesk</param-name>
  <param-value>01/234567</param-value>
</context-param>
```

2.9.2 IndexServlet: gewijzigde opdracht in de method doGet

```
request.setAttribute("telefoonNummerHelpDesk",
  request.getServletContext().getInitParameter("telefoonNummerHelpDesk"));
```

2.10 MVC

2.10.1 SausRepository

```
package be.vdab.repositories;
// enkele imports ...
public class SausRepository {
  private static final Map<Long, Saus> SAUZEN = new ConcurrentHashMap<>();
  static {
    SAUZEN.put(3L, new Saus(3, "cocktail",
      Arrays.asList("mayonaise", "ketchup", "cognac"))));
    SAUZEN.put(6L, new Saus(6, "mayonaise",
      Arrays.asList("ei", "mosterd"))));
    SAUZEN.put(7L, new Saus(7, "mosterd",
      Arrays.asList("mosterd", "azijn", "witte wijn"))));
    SAUZEN.put(12L, new Saus(12, "tartare",
      Arrays.asList("mayonaise", "augurk", "tabasco"))));
    SAUZEN.put(44L, new Saus(44, "vinaigrette",
      Arrays.asList("olijfolie", "mosterd", "azijn"))));
  }
  public List<Saus> findAll() {
    return new ArrayList<>(SAUZEN.values());
  }
}
```


2.10.2 SauzenServlet

```
package be.vdab.servlets;
// enkele imports ...
@WebServlet("/sauzen.htm")
public class SauzenServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    private static final String VIEW = "/WEB-INF/JSP/sauzen.jsp";
    private final SausRepository sausRepository = new SausRepository();
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        request.setAttribute("sauzen", sausRepository.findAll());
        request.getRequestDispatcher(VIEW).forward(request, response);
    }
}
```

2.11 Ingrediënten

2.11.1 SausRepository: extra code

```
public List<Saus> findByIngredient(String ingredient) {
    return SAUZEN.values().stream()
        .filter(saus -> saus.getIngredienten().contains(ingredient))
        .collect(Collectors.toList());
}
```

2.11.2 IngredientenServlet

```
package be.vdab.servlets;
// enkele imports ...
@WebServlet("/ingredienten.htm")
public class IngredientenServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    private static final String VIEW = "/WEB-INF/JSP/ingredienten.jsp";
    private final SausRepository sausRepository = new SausRepository();
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        String ingredient = request.getParameter("ingredient");
        if (ingredient != null) {
            if (ingredient.trim().isEmpty()) {
                request.setAttribute("fouten",
                    Collections.singletonMap("ingredient", "verplicht"));
            } else {
                request.setAttribute(
                    "sauzen", sausRepository.findByIngredient(ingredient));
            }
        }
        request.getRequestDispatcher(VIEW).forward(request, response);
    }
}
```

2.11.3 /WEB-INF/JSP/ingredienten.jsp

```
<%@page contentType='text/html' pageEncoding='UTF-8' session='false'%>
<%@taglib prefix='c' uri='http://java.sun.com/jsp/jstl/core'%>
<!doctype html>
<html lang='nl'>
  <head>
    <c:import url='head.jsp'>
      <c:param name='title' value='Ingrediënten' />
    </c:import>
  </head>
  <body>
    <h1>Ingrediënten</h1>
    <form>
      <label>Ingrediënt:<span>${fouten.ingredient}</span>
      <input name='ingredient' value='${param.ingredient}' autofocus required
        type='search'>
      </label>
      <input type='submit' value='Zoeken'>
    </form>
    <c:if test='${not empty sauzen}'>
      <ul>
        <c:forEach var='saus' items='${sauzen}'>
          <li>${saus.naam}</li>
        </c:forEach>
      </ul>
    </c:if>
    <c:if test='${empty sauzen and empty fouten}'>
      <div>Geen sauzen gevonden</div>
    </c:if>
  </body>
</html>
```

2.11.4 default.css: uitbreiding

```
label span {
  background-color: red;
  border-bottom-right-radius: 0.5em;
  border-top-left-radius: 0.5em;
  border-top-right-radius: 0.5em;
  box-shadow: 5px 5px #777777;
  margin-left: 0.5em;
  padding: 0.5em;
  color: white;
}
label span:empty {
  display: none;
}
input {
  display: block;
  margin-top: 0.2em;
  margin-bottom: 1em;
  font-size: 1.1em;
}
input[type='checkbox'] {
  display: inline;
}
input:focus,textarea:focus,select:focus {
  background-color: #FFFD6
}
```

2.12 Sauzen

2.12.1 /WEB-INF/JSP/sauzen.jsp

```
<%@page contentType='text/html' pageEncoding='UTF-8' session='false'%>
<%@taglib prefix='c' uri='http://java.sun.com/jsp/jstl/core'%>
<!doctype html>
<html lang='nl'>
  <head>
    <c:import url='head.jsp'>
      <c:param name='title' value='Sauzen'/>
    </c:import>
  </head>
  <body>
    <h1>Sauzen</h1>
    <form method='post' action="<c:url value='/sauzen/verwijderen.htm'/>">
      <c:forEach var='saus' items='${sausen}'>
        <h2>
          <label>
            <input type='checkbox' name='id' value='${saus.id}'>
              ${saus.naam}
            </label>
          </h2>
          <img src='<c:url value="/images/${saus.naam}.png"/>' alt='${saus.naam}'>
          ingrediënten:
          <c:forEach var='ingredient' items='${saus.ingredienten}'
            varStatus='status'>
            ${ingredient}<c:if test='${not status.last}'>, </c:if>
          </c:forEach>
        </c:forEach>
        <div><input type='submit' value='Aangevinkte sauzen verwijderen'></div>
      </form>
    </body>
  </html>
```

2.12.2 SausRepository: extra code

```
public void delete(Set<Long> idStream) {
    idStream.forEach(id -> SAUZEN.remove(id));
}
```

2.12.3 SauzenVerwijderenServlet

```
package be.vdab.servlets;
// enkele imports ...
@WebServlet("/sauzen/verwijderen.htm")
public class SauzenVerwijderenServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    private static final String REDIRECT_URL = "/sauzen.htm";
    private final SausRepository sausRepository = new SausRepository();
    @Override
    protected void doPost(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {
        String[] idsAlsString = request.getParameterValues("id");
        if (idsAlsString != null) {
            sausRepository.delete(Arrays.stream(idsAlsString)
                .map(id -> Long.parseLong(id))
                .collect(Collectors.toSet()));
        }
        response.sendRedirect(request.getContextPath() + REDIRECT_URL);
    }
}
```

2.13 Meisjes jongens

2.13.1 MeisjesJongensServlet

```
package be.vdab.servlets;
// enkele imports ...
@WebServlet("/meisjesjongens.htm")
public class MeisjesJongensServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    private static final String VIEW = "/WEB-INF/JSP/meisjesjongens.jsp";
    private static final int COOKIE_MAXIMUM_LEEFTIJD =
        60 /* seconden */ * 30 /* minuten */;
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        request.getRequestDispatcher(VIEW).forward(request, response);
    }
    @Override
    protected void doPost(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {
        String meisjesJongens = request.getParameter("meisjesjongens");
        Cookie cookie = new Cookie("meisjesjongens", meisjesJongens);
        cookie.setMaxAge(COOKIE_MAXIMUM_LEEFTIJD);
        response.addCookie(cookie);
        response.sendRedirect(request.getRequestURI());
    }
}
```

2.13.2 WEB-INF/JSP/meisjesjongens.jsp

```
<%@page contentType='text/html' pageEncoding='UTF-8' session='false'%>
<%@taglib prefix='c' uri='http://java.sun.com/jsp/jstl/core'%>
<!doctype html>
<html lang='nl'>
    <head>
        <c:import url='head.jsp'>
            <c:param name='title' value='Meisjes jongens'/>
        </c:import>
    </head>
    <body class="${cookie.meisjesjongens.value}">
        <h1>Meisjes jongens</h1>
        <form method='post'>
            <input type='submit' name='meisjesjongens' value='meisjes'>
            <input type='submit' name='meisjesjongens' value='jongens'>
        </form>
    </body>
</html>
```

2.13.3 default.css: uitbreiding

```
.meisjes {
    background-color: #FFCCFF;
    color: #FF0066;
}
.jongens {
    background-color: #CCFFFF;
    color: #3300FF;
}
```

2.14 Zoek de friet

2.14.1 Deur

```
package be.vdab.entities;
import java.io.Serializable;
public class Deur implements Serializable {
    private static final long serialVersionUID = 1L;
    private boolean open;
    private boolean metFriet;
    public Deur(boolean metFriet) {
        this.metFriet = metFriet;
    }
    public void open() {
        open = true;
    }
    public boolean isOpen() {
        return open;
    }
    public boolean isMetFriet() {
        return metFriet;
    }
}
```

2.14.2 ZoekDeFrietSpel

```
package be.vdab.entities;
import java.io.Serializable;
import java.util.Random;
public class ZoekDeFrietSpel implements Serializable {
    private static final long serialVersionUID = 1L;
    private static final int AANTAL_DEUREN = 7;
    private final Deur[] deuren = new Deur[AANTAL_DEUREN];
    public ZoekDeFrietSpel() {
        int indexMetFriet = new Random().nextInt(AANTAL_DEUREN);
        for (int index = 0; index != AANTAL_DEUREN; index++) {
            deuren[index] = new Deur(index == indexMetFriet);
        }
    }
    public void openDeur(int index) {
        deuren[index].open();
    }
    public Deur[] getDeuren() {
        return deuren;
    }
}
```

2.14.3 ZoekDeFrietServlet

```
package be.vdab.servlets;
// enkele imports ...
@WebServlet("/zoekdefriet.htm")
public class ZoekDeFrietServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    private static final String VIEW = "/WEB-INF/JSP/zoekdefriet.jsp";
    private static final String SPEL = "zoekDeFrietSpel";
```

```

@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    HttpSession session = request.getSession();
    if (session.getAttribute(SPEL) == null) {
        session.setAttribute(SPEL, new ZoekDeFrietSpel());
    }
    request.getRequestDispatcher(VIEW).forward(request, response);
}

@Override
protected void doPost(HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException {
    HttpSession session = request.getSession();
    if (request.getParameter("nieuwSpel") != null) {
        session.removeAttribute(SPEL);
    } else {
        int volgNummer = Integer.parseInt(request.getParameter("volgnummer"));
        ZoekDeFrietSpel spel = (ZoekDeFrietSpel) session.getAttribute(SPEL);
        if (spel != null) {
            spel.openDeur(volgNummer);
            session.setAttribute(SPEL, spel);
        }
    }
    response.sendRedirect(response.encodeRedirectURL(request.getRequestURI()));
}
}

```

2.14.4 WEB-INF/JSP/zoekdefriet.jsp

```

<%@page contentType='text/html' pageEncoding='UTF-8'%>
<!-- Op de vorige regel wordt session="false" niet vermeld -->
<!-- Zo komt session="true" (de default) -->
<%@taglib prefix='c' uri='http://java.sun.com/jsp/jstl/core'%>
<!doctype html>
<html lang='nl'>
<head>
<c:import url='head.jsp'>
    <c:param name='title' value='Zoek de friet'/>
</c:import>
</head>
<body>
<h1>Zoek de friet</h1>
<form method='post'>
    <c:forEach items='${zoekDeFrietSpel.deuren}' var='deur' varStatus='status'>
        <button type='submit' name='volgnummer' value='${status.index}'>
            <c:choose>
                <c:when test='${deur.open}'>
                    <c:choose>
                        <c:when test='${deur.metFriet}'>
                            <img src='<c:url value="/images/gevonden.png"/>' alt='gevonden'>
                        </c:when>
                        <c:otherwise>
                            <img src='<c:url value="/images/deuopen.png"/>' alt='deur open'>
                        </c:otherwise>
                    </c:choose>
                </c:when>
                <c:otherwise>
                    <img src='<c:url value="/images/deurtoe.png"/>' alt='deur toe'>
                </c:otherwise>
            </c:choose>
        </button>
    </c:forEach>

```

```

</form>
<c:url value='' var='nieuwSpelURL'>
  <c:param name='nieuwSpel' value='true'/>
</c:url>
<form method='post' action='${nieuwSpelURL}'>
  <input type='submit' value='Nieuw spel'>
</form>
</body>
</html>

```

2.15 Saus raden

2.15.1 SausRadenSpel

```

package be.vdab.entities;
// enkele imports ...
public class SausRadenSpel implements Serializable {
    private static final long serialVersionUID = 1L;
    private static final int MAX_BEURTEN = 10;
    private final String saus;
    private final StringBuilder sausMetPuntjes;
    private int verkeerdeBeurten;
    public SausRadenSpel(String saus) {
        this.saus = saus;
        sausMetPuntjes = new StringBuilder(saus.length());
        IntStream.rangeClosed(1, saus.length())
            .forEach(teller -> sausMetPuntjes.append('.'));
    }
    public void doeGok(char letter) {
        int letterIndex = saus.indexOf(letter);
        if (letterIndex == -1) {
            verkeerdeBeurten++;
        } else {
            do {
                sausMetPuntjes.setCharAt(letterIndex, letter);
                letterIndex = saus.indexOf(letter, letterIndex + 1);
            } while (letterIndex != -1);
        }
    }
    public String getSausMetPuntjes() {
        return sausMetPuntjes.toString();
    }
    public String getSaus() {
        return saus;
    }
    public int getVerkeerdeBeurten() {
        return verkeerdeBeurten;
    }
    public boolean isGewonnen() {
        return sausMetPuntjes.indexOf(".") == -1;
    }
    public boolean isVerloren() {
        return verkeerdeBeurten == MAX_BEURTEN;
    }
}

```

2.15.2 SausRadenServlet

```
package be.vdab.servlets;
// enkele imports ...
@WebServlet("/sausraden.htm")
public class SausRadenServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    private static final String VIEW = "/WEB-INF/JSP/sausraden.jsp";
    private static final String SPEL = "sausRadenSpel";
    private final SausRepository sausRepository = new SausRepository();
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        HttpSession session = request.getSession();
        if (session.getAttribute(SPEL) == null) {
            List<Saus> sauzen = new ArrayList<>(sausRepository.findAll());
            Saus randomSaus = sauzen.get(new Random().nextInt(sauzen.size()));
            SausRadenSpel spel = new SausRadenSpel(randomSaus.getNaam());
            session.setAttribute(SPEL, spel);
        }
        request.getRequestDispatcher(VIEW).forward(request, response);
    }
    @Override
    protected void doPost(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {
        HttpSession session = request.getSession();
        if (request.getParameter("nieuwSpel") != null) {
            session.removeAttribute(SPEL);
        } else {
            String letter = request.getParameter("letter");
            SausRadenSpel spel = (SausRadenSpel) session.getAttribute(SPEL);
            if (spel != null && letter != null && ! letter.isEmpty()) {
                spel.doeGok(letter.charAt(0));
                session.setAttribute(SPEL, spel);
            }
        }
        response.sendRedirect(response.encodeRedirectURL(request.getRequestURI()));
    }
}
```

2.15.3 WEB-INF/JSP/sausraden.jsp

```
<%@page contentType='text/html' pageEncoding='UTF-8'%>
<!-- Op de vorige regel wordt session="false" niet vermeld --%>
<!-- Zo komt session="true" (de default) --%>
<%@taglib prefix='c' uri='http://java.sun.com/jsp/jstl/core'%>
<!doctype html>
<html lang='nl'>
    <head>
        <c:import url='head.jsp'>
            <c:param name='title' value='Saus raden'/>
        </c:import>
    </head>
    <body>
        <h1>Saus raden</h1>
        <c:choose>
            <c:when test="${sausRadenSpel.verloren}">
                U bent verloren, de saus was ${sausRadenSpel.saus}.
            </c:when>
            <c:when test="${sausRadenSpel.gewonnen}">
                U bent gewonnen, de saus was ${sausRadenSpel.saus}.
            </c:when>
        </c:choose>
    </body>
</html>
```



```

<c:otherwise>
    Te raden saus: ${sausRadenSpel.sausMetPuntjes}
    <form method='post' id='radenform'>
        <label>letter:
            <input name='letter' size='1' maxlength='1' autofocus required>
        </label>
        <input type='submit' value='Raden' id='radenknop'>
    </form>
    <script>
        document.getElementById('radenform').onsubmit = function() {
            document.getElementById('radenknop').disabled = true;
        };
    </script>
</c:otherwise>
</c:choose>
<c:url value='' var='nieuwSpelURL'>
    <c:param name='nieuwSpel' value='true' />
</c:url>
<form method='post' action='${nieuwSpelURL}'>
    <input type='submit' value='Nieuw spel'>
</form>

</body>
</html>

```

2.16 Statistiek

2.16.1 RequestStatistiekListener

```

package be.vdab.listeners;
// enkele imports ...
@WebListener
public class RequestStatistiekListener implements ServletRequestListener,
    ServletContextListener {
    private final static String STATISTIEK = "statistiek";
    private final static Set<String> UITGESLOTEN_EXTENSIES = new
        CopyOnWriteArraySet<>(Arrays.asList("png", "gif", "jpg", "css", "js", "ico"));
    @Override
    public void contextInitialized(ServletContextEvent event) {
        event.getServletContext().setAttribute(STATISTIEK,
            new ConcurrentHashMap<String, AtomicInteger>());
    }
    @Override
    public void contextDestroyed(ServletContextEvent event) {
    }
    @Override
    public void requestInitialized(ServletRequestEvent event) {
        if (event.getServletRequest() instanceof HttpServletRequest) {
            HttpServletRequest request =
                (HttpServletRequest) event.getServletRequest();
            String url = request.getRequestURI();
            boolean verwerkRequest = true;
            int laatstePuntIndex = url.lastIndexOf('.');
            if (laatstePuntIndex != -1) {
                String extensie =
                    url.substring(laatstePuntIndex + 1).toLowerCase();
                if (UITGESLOTEN_EXTENSIES.contains(extensie)) {
                    verwerkRequest = false;
                }
            }
        }
    }
}

```

```

    if (verwerkRequest) {
        int index = url.indexOf(";jsessionid=");
        if (index != -1) {
            url = url.substring(0, index);
        }
        @SuppressWarnings("unchecked")
        ConcurrentHashMap<String, AtomicInteger> statistiek =
            (ConcurrentHashMap<String, AtomicInteger>) request.getServletContext()
                .getAttribute(STATISTIEK);
        AtomicInteger aantalReedsAanwezig =
            statistiek.putIfAbsent(url, new AtomicInteger(1));
        if (aantalReedsAanwezig != null) {
            aantalReedsAanwezig.incrementAndGet();
        }
    }
}
}
@Override
public void requestDestroyed(ServletRequestEvent event) {
}
}

```

2.16.2 StatistiekServlet

```

package be.vdab.servlets;
// enkele imports ...
@WebServlet("/statistiek.htm")
public class StatistiekServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    private static final String VIEW = "/WEB-INF/JSP/statistiek.jsp";
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        request.getRequestDispatcher(VIEW).forward(request, response);
    }
}

```

2.16.3 WEB-INF/JSP/statistiek.jsp

```

<%@page contentType='text/html' pageEncoding='UTF-8' session='false'%>
<%@taglib prefix='c' uri='http://java.sun.com/jsp/jstl/core'%>
<!doctype html>
<html lang='nl'>
    <head>
        <c:import url='head.jsp'>
            <c:param name='title' value='Statistiek'/>
        </c:import>
        <style>
            td:last-child {
                text-align:right;
            }
        </style>
    </head>
    <body>
        <h1>Statistiek</h1>
        <c:if test='${not empty statistiek}'>
            <table>
                <thead>
                    <tr>
                        <th>URL</th><th>aantal requests</th>
                    </tr>
                </thead>
            </table>
        </c:if>
    </body>
</html>

```

```

        <tbody>
        <c:forEach var='entry' items='${statistiek}'>
            <tr>
                <td>${entry.key}</td>
                <td>${entry.value}</td>
            </tr>
        </c:forEach>
        </tbody>
    </table>
</c:if>
</body>
</html>

```

2.17 Statistiek 2

2.17.1 RequestStatistiekFilter

```

package be.vdab.filters;
// enkele imports ...
@WebFilter("*.htm")
public class RequestStatistiekFilter implements Filter {
    private final static String STATISTIEK = "statistiek";

    @Override
    public void init(FilterConfig filterConfig) throws ServletException {
        filterConfig.getServletContext().setAttribute(STATISTIEK,
            new ConcurrentHashMap<String, AtomicInteger>());
    }

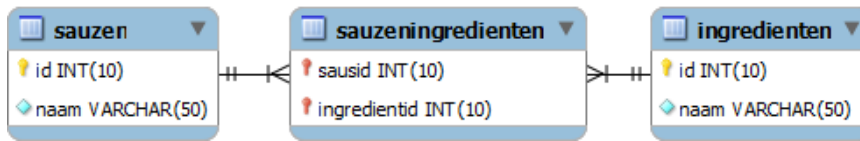
    @Override
    public void doFilter(ServletRequest request, ServletResponse response,
        FilterChain chain) throws IOException, ServletException {
        if (request instanceof HttpServletRequest) {
            String url = ((HttpServletRequest) request).getRequestURI();
            int index = url.indexOf(";jsessionid=");
            if (index != -1) {
                url = url.substring(0, index);
            }
            @SuppressWarnings("unchecked")
            ConcurrentHashMap<String, AtomicInteger> statistiek =
                (ConcurrentHashMap<String, AtomicInteger>) request.getServletContext()
                    .getAttribute(STATISTIEK);
            AtomicInteger aantalReedsAanwezig =
                statistiek.putIfAbsent(url, new AtomicInteger(1));
            if (aantalReedsAanwezig != null) {
                aantalReedsAanwezig.incrementAndGet();
            }
        }
        chain.doFilter(request, response);
    }

    @Override
    public void destroy() {
    }
}

```

2.18 Database

2.18.1 Databasestructuur



2.18.2 META-INF/context.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<Context path="/frituurfrida">
    <Resource name="jdbc/frituurfrida"
        type="javax.sql.DataSource"
        driverClassName="com.mysql.jdbc.Driver"
        url="jdbc:mysql://localhost/frituurfrida?useSSL=false"
        username="cursist" password="cursist"
        closeMethod="close"/>
</Context>
  
```

2.18.3 RepositoryException

```

package be.vdab.repositories;
public class RepositoryException extends RuntimeException {
    private static final long serialVersionUID = 1L;
    public RepositoryException(Throwable cause) {
        super(cause);
    }
}
  
```

2.18.4 AbstractRepository

```

package be.vdab.repositories;
import javax.sql.DataSource;
abstract class AbstractRepository {
    public final static String JNDI_NAME = "jdbc/frituurfrida";
    protected DataSource dataSource;
    public void setDataSource(DataSource dataSource) {
        this.dataSource = dataSource;
    }
}
  
```

2.18.5 SausRepository

```

package be.vdab.repositories;
// enkele imports ...
public class SausRepository extends AbstractRepository {
    private static final String FIND_ALL =
        "select sauzen.id, sauzen.naam as sausnaam," +
        "ingredienten.naam as ingredientnaam" +
        "from sauzen left join sauzeningredienten" +
        "on sauzen.id = sauzeningredienten.sausid" +
        "left join ingredienten " +
        "on sauzeningredienten.ingredientid = ingredienten.id" +
        "order by sauzen.naam";
    private static final String FIND_BY_INGREDIENT =
        "select sauzen.id, sauzen.naam as sausnaam" +
        "from sauzen inner join sauzeningredienten " +
        "on sauzen.id = sauzeningredienten.sausid" +
        "inner join ingredienten" +
        "on sauzeningredienten.ingredientid = ingredienten.id" +
        "where ingredienten.naam = ?" + " order by sauzen.naam";
    private static final String DELETE = "delete from sauzen where id in (";
  
```

```

public List<Saus> findAll() {
    try(Connection connection = dataSource.getConnection();
        Statement statement = connection.createStatement()) {
        connection.setTransactionIsolation(Connection.TRANSACTION_READ_COMMITTED);
        connection.setAutoCommit(false);
        List<Saus> sauzen = new ArrayList<>();
        try(ResultSet resultSet = statement.executeQuery(FIND_ALL)) {
            for (long vorigeId = 0; resultSet.next();) {
                long id = resultSet.getLong("id");
                if (id != vorigeId) {
                    sauzen.add(resultSetRijNaarSausZonderIngredienten(resultSet));
                    vorigeId = id;
                }
                sauzen.get(sauzen.size() - 1).addIngredient(
                    resultSet.getString("ingredientnaam"));
            }
        }
        connection.commit();
        return sauzen;
    } catch (SQLException ex) {
        throw new RepositoryException(ex);
    }
}

public List<Saus> findByIngredient(String ingredient) {
    try (Connection connection = dataSource.getConnection();
        PreparedStatement statement = connection.prepareStatement(
            FIND_BY_INGREDIENT)) {
        connection.setTransactionIsolation(Connection.TRANSACTION_READ_COMMITTED);
        connection.setAutoCommit(false);
        statement.setString(1, ingredient);
        List<Saus> sauzen = new ArrayList<>();
        try (ResultSet resultSet = statement.executeQuery()) {
            while (resultSet.next()) {
                sauzen.add(resultSetRijNaarSausZonderIngredienten(resultSet));
            }
            connection.commit();
            return sauzen;
        }
    } catch (SQLException ex) {
        throw new RepositoryException(ex);
    }
}

private Saus resultSetRijNaarSausZonderIngredienten(ResultSet resultSet)
    throws SQLException {
    return new Saus(resultSet.getLong("id"), resultSet.getString("sausnaam"),
        new ArrayList<String>());
}

public void delete(Set<Long> ids) {
    StringBuilder sql = new StringBuilder(DELETE);
    ids.forEach(id -> sql.append("?, "));
    sql.deleteCharAt(sql.length() - 1);
    sql.append(' ');
    try (Connection connection = dataSource.getConnection();
        PreparedStatement statement =
            connection.prepareStatement(sql.toString())) {
        connection.setTransactionIsolation(Connection.TRANSACTION_READ_COMMITTED);
        connection.setAutoCommit(false);
        int index = 1;
        for (long id : ids) {
            statement.setLong(index++, id);
        }
    }
}

```

```

        statement.executeUpdate();
        connection.commit();
    } catch (SQLException ex) {
        throw new RepositoryException(ex);
    }
}
}

```

2.18.6 Extra code in Servlets die een variabele sausRepository hebben

```

@Resource(name = SausRepository.JNDI_NAME)
public void setDataSource(DataSource dataSource) {
    sausRepository.setDataSource(dataSource);
} // en transient bij de declaratie van de variabele sausRepository

```

2.19 Vertalen

2.19.1 resourceBundles/teksten.properties

```

frituurFrida=Frituur Frida
vandaagZijnWeopen=Vandaag zijn we open
vandaagZijnWegesloten=Vandaag zijn we gesloten
afbeeldingopen=open
afbeeldinggesloten=gesloten
open=open
gesloten=gesloten
adres=adres

```

2.19.2 resourceBundles/teksten_en.properties

```

frituurFrida=Chips stand Frida
vandaagZijnWeopen=Today we're open
vandaagZijnWegesloten=Today we're closed
afbeeldingopen=open_en
afbeeldinggesloten=gesloten_en
open=open
gesloten=closed
adres=address

```

2.19.3 index.jsp

```

<%@page contentType='text/html' pageEncoding='UTF-8' session='false'%>
<%@taglib prefix='c' uri='http://java.sun.com/jsp/jstl/core'%>
<%@taglib prefix='fmt' uri='http://java.sun.com/jsp/jstl/fmt'%>
<fmt:setBundle basename='resourceBundles.teksten'/>
<!doctype html>
<html>
  <head>
    <fmt:message key='frituurFrida' var='title'/>
    <c:import url='head.jsp'><c:param name='title' value='${title}'/></c:import>
  </head>
  <body>
    <h1><fmt:message key='vandaagZijnWe${openGesloten}'/></h1>
    <fmt:message key='afbeelding${openGesloten}' var='afbeelding'/>
    <img src='<c:url value="/images/${afbeelding}.png"/>'
      alt="<fmt:message key='${openGesloten}'/>"
    <h2><fmt:message key='adres'/></h2>
    ${adres.straat} ${adres.huisNr}<br>
    ${adres.gemeente.postcode} ${adres.gemeente.naam}
  </body>
</html>

```

2.20 Menu

2.20.1 /WEB-INF/tags/vdab/menu.tag

```
<%@tag description='website menu' pageEncoding='UTF-8'%>
<%@taglib prefix='c' uri='http://java.sun.com/jsp/jstl/core'%>
<header>
  <nav>
    <ul>
      <li><a href="<c:url value='/'/'>">Welkom</a></li>
      <li><a href="<c:url value='sauzen.htm'/'>">Sauzen</a></li>
      <li><a href="<c:url value='ingredienten.htm'/'>">Ingrediënten</a></li>
      <li><a href="<c:url value='meisjesjongens.htm'/'>">Meisjes jongens</a></li>
      <li><a href="<c:url value='zoekdefriet.htm'/'>">Zoek de friet</a></li>
      <li><a href="<c:url value='sausraden.htm'/'>">Saus raden</a></li>
      <li><a href="<c:url value='statistiek.htm'/'>">Statistiek</a></li>
    </ul>
  </nav>
</header>
```

2.20.2 /WEB-INF/tlds/vdab.tld

```
<?xml version="1.0" encoding="UTF-8"?>
<taglib version="2.1" xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-jsptaglibrary_2_1.xsd">
<tlib-version>1.0</tlib-version>
<short-name>vdab</short-name>
<uri>http://vdab.be/tags</uri>
<tag-file>
<name>menu</name>
<path>/WEB-INF/tags/vdab/menu.tag</path>
</tag-file>
</taglib>
```

2.20.3 default.css: uitbreiding

```
nav ul {
  list-style:none;
  padding:0;
}
nav ul li {
  display:inline;
}
```

2.20.4 JSP's

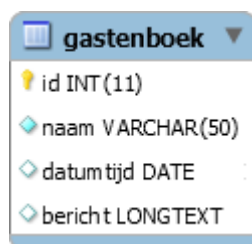
Je voegt boven in iedere JSP volgende regel toe:

```
<%@taglib prefix='vdab' uri='http://vdab.be/tags'%>
```

Je voegt in iedere JSP onder de regel <body> volgende regel toe: <vdab:menu/>

2.21 Gastenboek

2.21.1 Nieuwe table in de database



2.21.2 GastenBoekEntry

```

package be.vdab.entities;
import java.time.LocalDate;
public class GastenBoekEntry {
    private final long id;
    private final String naam;
    private final LocalDate datum;
    private final String bericht;
    public GastenBoekEntry(String naam, String bericht) {
        if (!isNaamValid(naam)) {
            throw new IllegalArgumentException();
        }
        if (!isBerichtValid(bericht)) {
            throw new IllegalArgumentException();
        }
        this.id = 0;
        this.naam = naam;
        this.datum = LocalDate.now();
        this.bericht = bericht;
    }
    public GastenBoekEntry(long id, String naam, LocalDate datum, String bericht){
        this.id = id;
        this.naam = naam;
        this.datum = datum;
        this.bericht = bericht;
    }
    public static boolean isNaamValid(String naam) {
        return naam != null && !naam.trim().isEmpty();
    }
    public static boolean isBerichtValid(String bericht) {
        return bericht != null && !bericht.trim().isEmpty();
    }
    // en getters voor alle private variabelen
}

```

2.21.3 GastenBoekServlet

```

package be.vdab.servlets;
// enkele imports ...
@WebServlet("/gastenboek.htm")
public class GastenBoekServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    private static final String VIEW = "/WEB-INF/JSP/gastenboek.jsp";
    private static final String REDIRECT_URL = "/gastenboek.htm";
    private final GastenboekRepository gastenBoekRepository =
        new GastenboekRepository();
    @Resource(name = GastenboekRepository.JNDI_NAME)
    public void setDataSource(DataSource dataSource) {
        gastenBoekRepository.setDataSource(dataSource);
    }
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        request.setAttribute("gastenboek", gastenBoekRepository.findAll());
        request.getRequestDispatcher(VIEW).forward(request, response);
    }
    @Override
    protected void doPost(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {

```



```

    if (request.getParameter("toevoegen") != null) {
        toevoegen(request, response);
    }
}

private void toevoegen(HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException {
    Map<String, String> fouten = new HashMap<>();
    String naam = request.getParameter("naam");
    if (!GastenBoekEntry.isNaamValid(naam)) {
        fouten.put("naam", "verplicht");
    }
    String bericht = request.getParameter("bericht");
    if (!GastenBoekEntry.isBerichtValid(bericht)) {
        fouten.put("bericht", "verplicht");
    }
    if (fouten.isEmpty()) {
        gastenBoekRepository.create(new GastenBoekEntry(naam, bericht));
        response.sendRedirect(response.encodeRedirectURL(
            request.getContextPath() + REDIRECT_URL));
    } else {
        request.setAttribute("fouten", fouten);
        request.getRequestDispatcher(VIEW).forward(request, response);
    }
}
}
}

```

2.21.4 GastenboekRepository

```

package be.vdab.repositories;
// enkele imports ...

public class GastenboekRepository extends AbstractRepository {
    private final static String FIND_ALL =
        "select id,naam,datumtijd,bericht from gastenboek order by id desc";
    private final static String CREATE =
        "insert into gastenboek(naam,datumtijd,bericht) values (?,?,?)";
    public List<GastenBoekEntry> findAll() {
        try (Connection connection = dataSource.getConnection();
            Statement statement = connection.createStatement()) {
            List<GastenBoekEntry> entries = new ArrayList<>();
            connection.setTransactionIsolation(Connection.TRANSACTION_READ_COMMITTED);
            connection.setAutoCommit(false);
            try (ResultSet resultSet = statement.executeQuery(FIND_ALL)) {
                while (resultSet.next()) {
                    entries.add(resultSetRijNaarGastenboekEntry(resultSet));
                }
            }
            connection.commit();
            return entries;
        } catch (SQLException ex) {
            throw new RepositoryException(ex);
        }
    }

    public void create(GastenBoekEntry entry) {
        try (Connection connection = dataSource.getConnection();
            PreparedStatement statement =
                connection.prepareStatement(CREATE)) {
            connection.setTransactionIsolation(Connection.TRANSACTION_READ_COMMITTED);
            connection.setAutoCommit(false);
            statement.setString(1, entry.getNaam());
            statement.setDate(2, java.sql.Date.valueOf(entry.getDatum()));
            statement.setString(3, entry.getBericht());
            statement.executeUpdate();
        }
    }
}

```

```

        connection.commit();
    } catch (SQLException ex) {
        throw new RepositoryException(ex);
    }
}

private GastenBoekEntry resultSetRijNaarGastenboekEntry(
    ResultSet resultSet) throws SQLException {
    return new GastenBoekEntry(resultSet.getLong("id"),
        resultSet.getString("naam"), resultSet.getDate("datum").toLocalDate(),
        resultSet.getString("bericht"));
}
}

```

2.21.5 menu.tag: extra regel

```
<li><a href="<c:url value= '/gastenboek.htm' />">Gastenboek</a></li>
```

2.21.6 default.css: aangepaste regel

```
input, textarea {
```

2.21.7 gastenboek.jsp

```

<%@ page contentType='text/html' pageEncoding='UTF-8' session='false'%>
<%@taglib prefix='c' uri='http://java.sun.com/jsp/jstl/core'%>
<%@taglib prefix='fmt' uri='http://java.sun.com/jsp/jstl/fmt'%>
<!doctype html>
<html lang='nl'>
<head>
<c:import url='head.jsp'>
    <c:param name='title' value='Gastenboek'/>
</c:import>
</head>
<body>
<h1>Gastenboek</h1>
<c:choose>
    <c:when test='${empty param.toevoegen}'>
        <a href='<c:url value= "/gastenboek.htm?toevoegen=true" />'>Toevoegen</a>
    </c:when>
    <c:otherwise>
        <form method='post'>
            <label>Naam:<span>${fouten.naam}</span><input name='naam' autofocus
                required value='${param.naam}'></label>
            <label>Bericht:<span>${fouten.bericht}</span><textarea name='bericht'
                required rows='5' cols='80'>${param.bericht}</textarea></label>
            <input type='submit' value='Toevoegen' name='toevoegen'>
        </form>
    </c:otherwise>
</c:choose>
<c:if test='${not empty gastenboek}'>
    <dl>
    <c:forEach var='entry' items='${gastenboek}'>
        <fmt:parseDate value='${entry.datum}' pattern="yyyy-MM-dd"
            var="datumAlsDate" type="date"/>
        <dt><fmt:formatDate value='${datumAlsDate}' type='both' dateStyle='short'
            timeStyle='short' />
            ${entry.naam}
        </dt>
        <dd>${entry.bericht}</dd>
    </c:forEach>
    </dl>
</c:if>
</body>
</html>

```

2.22 Gastenboekbeheer

2.22.1 web.xml: extra regels

```
<context-param>
  <param-name>paswoord</param-name>
  <param-value>bolle</param-value>
</context-param>
```

2.22.2 InloggenServlet

```
package be.vdab.servlets;
// enkele imports ...
@WebServlet("/inloggen.htm")
public class InloggenServlet extends HttpServlet {
  private static final long serialVersionUID = 1L;
  private static final String VIEW = "/WEB-INF/JSP/inloggen.jsp";
  private static final String REDIRECT_URL = "/gastenboek.htm";

  @Override
  protected void doGet(HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException {
    request.getRequestDispatcher(VIEW).forward(request, response);
  }

  @Override
  protected void doPost(HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException {
    if (!this.getServletContext().getInitParameter("paswoord")
      .equals(request.getParameter("paswoord"))) {
      request.setAttribute("fout", "verkeerd");
      request.getRequestDispatcher(VIEW).forward(request, response);
    } else {
      request.getSession().setAttribute("beheer", true);
      response.sendRedirect(response.encodeRedirectURL(
        request.getContextPath() + REDIRECT_URL));
    }
  }
}
```

2.22.3 Inloggen.jsp

```
<%@ page contentType='text/html' pageEncoding='UTF-8' session='false'%>
<%@taglib prefix='c' uri='http://java.sun.com/jsp/jstl/core'%>
<!doctype html>
<html lang='nl'>
<head>
<c:import url='head.jsp'>
  <c:param name='title' value='Inloggen'/>
</c:import>
</head>
<body>
  <h1>Inloggen</h1>
  <form method='post'>
    <label>Paswoord:<span>${fout}</span><input name='paswoord' autofocus
      required type='password'></label>
    <input type='submit' value='Inloggen'>
  </form>
</body>
</html>
```

2.22.4 gastenboek.jsp

extra code na </h1>

```
<c:if test='${not empty beheer}'>
  <form method='post'>
    <input type='submit' value='Uitloggen' name='uitloggen'>
  </form>
</c:if>
```

extra code voor <dl>

```
<c:if test='${not empty beheer}'>
  <form method='post'>
</c:if>
```

extra code voor </dt>

```
<c:if test='${not empty beheer}'>
  <input type='checkbox' name='id' value='${entry.id}'>
</c:if>
```

extra code voor </dl>

```
<c:if test='${not empty beheer}'>
  <input type='submit' value='Verwijderen' name='verwijderen'>
</form>
</c:if>
```

2.22.5 GastenboekServlet

extra code in de method doPost:

```
else if (request.getParameter("uitloggen") != null) {
    uitloggen(request, response);
} else if (request.getParameter("verwijderen") != null) {
    verwijderen(request, response);
}
```

extra method:

```
private void uitloggen(HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException {
    request.getSession().removeAttribute("beheer");
    response.sendRedirect(response.encodeRedirectURL(
        request.getContextPath() + REDIRECT_URL));
}
```

extra method:

```
private void verwijderen(HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException {
    String[] idsAlsString = request.getParameterValues("id");
    if (idsAlsString != null) {
        gastenBoekRepository.delete(
            Arrays.stream(idsAlsString)
                .map(id -> Long.parseLong(id))
                .collect(Collectors.toSet()));
    }
    response.sendRedirect(response.encodeRedirectURL(
        request.getContextPath() + REDIRECT_URL));
}
```

2.22.6 GastenboekRepository

extra constante

```
private final static String DELETE = "delete from gastenboek where id in (";
```

extra method

```
public void delete(Set<Long> ids) {
    StringBuilder sql = new StringBuilder(DELETE);
    ids.forEach(id -> sql.append("?, "));
    sql.setCharAt(sql.length() - 1, ')');
    try (Connection connection = dataSource.getConnection();
        PreparedStatement statement =
            connection.prepareStatement(sql.toString())) {
        connection.setTransactionIsolation(Connection.TRANSACTION_READ_COMMITTED);
        connection.setAutoCommit(false);
        int index = 1;
        for (long id : ids) {
            statement.setLong(index++, id);
        }
        statement.executeUpdate();
        connection.commit();
    } catch (SQLException ex) {
        throw new RepositoryException(ex);
    }
}
```

3 COLOFON

Domeinexpertisemanager:	Jean Smits
Moduleverantwoordelijke:	Hans Desmet
Medewerkers:	Hans Desmet
Versie:	16/8/2017