VDAB

samen sterk voor werk

Spring advanced takenbundel

Deze cursus is eigendom van de VDAB©

Inhoudsopgave

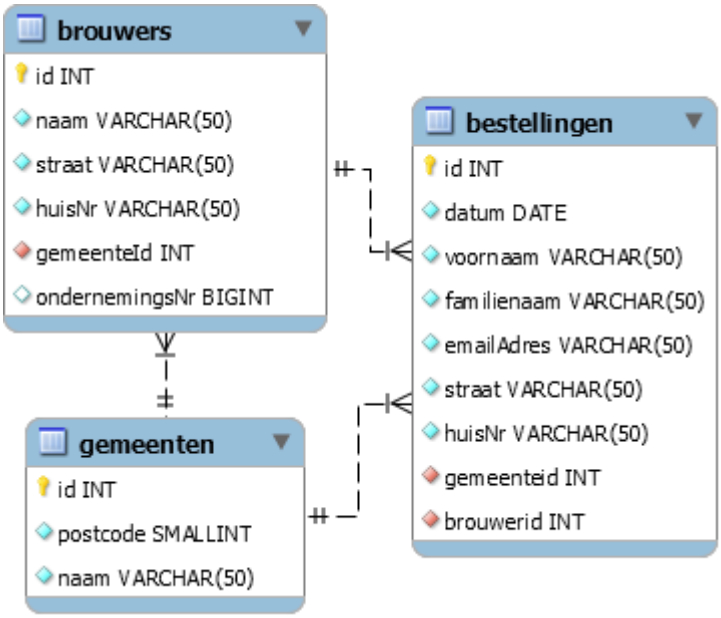
1 TAKEN

1.1 Proefpakket

Je maakt een website voor een firma die proefpakketten verdeelt.
Een proefpakket bevat de bieren van een Belgische brouwer.

Je maakt de bijbehorende database met het script `proefpakket.sql`.

De database bevat volgende tables:



Opmerking: op dit moment is de kolom `ondernemingsNr` bij geen enkele brouwer ingevuld.

Je voegt op http://start.spring.io volgende dependencies toe: `Web`, `DevTools`, `JPA`, `MySQL`, `Thymeleaf`, `Mail` en `JMS (ActiveMQ)`

Je voegt de instellingen voor de databaseverbinding toe aan `application.properties`.
Je logt in met de gebruikersnaam `cursist`, paswoord `cursist`.

Je maakt de welkompagina van de website. Deze pagina ziet er als volgt uit:

1.2 Brouwers

Je toont onder de foto in de welkompagina de letters van het alfabet. Iedere letter is een hyperlink. Als de gebruiker zo'n letter kiest, toon je onder het alfabet de namen van de brouwers wiens naam met die letter begint. Je sorteert de namen alfabetisch:

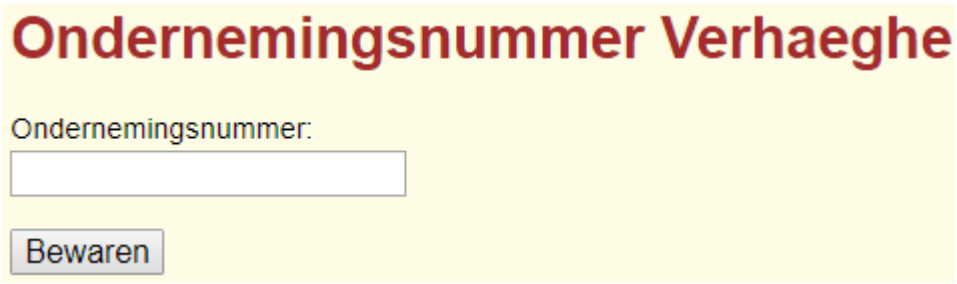
Elke naam is een hyperlink. Als de gebruiker een naam kiest, ziet hij een nieuwe pagina met de gegevens van de gekozen brouwer:

1.3 Ondernemingsnummer

Je voegt aan de detailpagina van de brouwer een hyperlink Ondernemingsnummer invullen toe.

Als de gebruiker hierop klikt ziet hij een nieuwe pagina.
Hij kan daar het ondernemingsnummer van de brouwer invullen.

Een ondernemingsnummer bevat een controle: de laatste 2 cijfers moeten gelijk zijn aan 97 – de rest van de deling van de overige cijfers door 97.
Een voorbeeld van een correct ondernemingsnummer: 426388541



1.4 Aanvraag

Je voegt aan de detailpagina van de brouwer een hyperlink Proefpakket aanvragen toe.

Als de gebruiker hierop klikt ziet hij een nieuwe pagina.
Hij kan zijn voornaam, familienaam en email adres invullen.

De pagina bevat een knop `Stap 2`.

Als de gebruiker hierop klikt, ziet hij een nieuwe pagina.
Hij kan zijn straat en huisnummer invullen en zijn gemeente kiezen uit een lijst met alle gemeenten.

De pagina bevat een knop `Stap 1`. Als de gebruiker hierop klikt ziet hij de vorige pagina.
De pagina bevat een knop `Opslaan`.
Als de gebruiker hierop klikt, sla je de gegevens op als een nieuwe bestelling in de database.

1.5 Meertalig

Je voegt onder in de welkompagina een hyperlink Nederlands toe.
Als de gebruiker hierop klikt, ziet hij de welkompagina in het Nederlands.

Je voegt ook een hyperlink Français toe.
Als de gebruiker hierop klikt, ziet hij de welkompaginain het Frans.

Je onthoudt de gekozen taal in een cookie.

1.6 REST

Je zorgt er voor dat je de data van een gemeente kan opvragen in XML of JSON formaat.

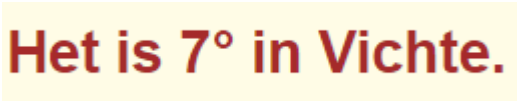
Een GET request naar http://localhost:8080/gemeenten/1.xml geeft bijvoorbeeld volgend resultaat

```
<?xml version="1.0" encoding="UTF-8"?>
<gemeente>
    <id>1</id>
    <postcode>1000</postcode>
    <naam>BRUSSEL</naam>
</gemeente>
```

Als een gemeente niet gevonden wordt, stuur je een response met status code 404.

1.7 Temperatuur

Je toont op de detailpagina van een brouwer een hyperlink Temperatuur.
Als de gebruiker deze aanklikt, toon je in een nieuwe pagina de temperatuur in die gemeente.



Je kan die temperatuur ophalen op
`http://api.openweathermap.org/data/2.5/weather?q=xxx&units=metric&APPID=yyy`

- Je vervangt xxx door de gemeente.
- Je vervangt yyy door je API key.
  Je krijgt die door een account te maken op https://openweathermap.org.

1.8 Mail

Je stuurt, bij het verwerken van de aanvraag van een proefpakket, een mail naar de aanvrager met volgende tekst:

`Bedankt voor uw interesse. U ontvangt uw proefpakket xxx binnenkort.`

Je vervangt xxx door de naam van de brouwer waarvan het proefpakket werd besteld.

1.9 JMS

Als je de aanvraag van een proefpakket verwerkt, stuurt je niet onmiddellijk een mail.
Je stuurt een object van de class `ProefpakketMessage` naar een ActiveMq queue.
Deze class heeft twee properties: `emailAdres` (`String`) en `brouwerNaam` (`String`).
Je applicatie verwerkt ook messages die in deze queue aankomen.
Bij dit verwerken verstuur je de mail die je in de vorige taak gemaakt hebt.

1.10 Security

Je definieert de authority `administrator`. Je definieert in deze authority minstens één gebruiker.
Je onthoudt de authority en gebruikers in de database.

Alle gebruikers (ook niet ingelogde) kunnen alle onderdelen van de website gebruiken, tenzij het onderdeel `Ondernemingsnummer invullen`. Enkel ingelogde gebruikers met de authority `administrator` kunnen dit onderdeel gebruiken.

2 VOORBEELDOPLOSSNGEN

2.1 Proefpakket

2.1.1 application.properties

```
spring.datasource.url=jdbc:mysql://localhost/proefpakket?useSSL=false
spring.datasource.username=cursist
spring.datasource.password=cursist
spring.jpa.hibernate.naming.physical-strategy=\
org.hibernate.boot.model.naming.PhysicalNamingStrategyStandardImpl
logging.level.org.hibernate.SQL=DEBUG
logging.level.org.hibernate.type.descriptor.sql.BasicBinder=TRACE
```

2.1.2 IndexController

```
package be.vdab.proefpakket.web;
// enkele imports
@Controller
@RequestMapping("/")
class IndexController {
  private static final String VIEW = "index";
  @GetMapping
  String index() {
    return VIEW;
  }
}
```

2.1.3 fragments.html

```
<!doctype html>
<html lang='nl' xmlns:th='http://www.thymeleaf.org'>
<head th:fragment='head(title)'>
  <meta charset='UTF-8'>
  <link rel='icon' th:href='@{/images/proefpakket.ico}' type='image/x-icon'>
  <title th:text='${title}'></title>
  <meta name='viewport' content='width=device-width,initial-scale=1'>
  <link rel='stylesheet' th:href='@{/css/proefpakket.css}'>
</head>
<body>
</body>
</html>
```

2.1.4 index.html

```
<!doctype html>
<html lang='nl' xmlns:th="http://www.thymeleaf.org">
<head th:replace="fragments::head(title='Proefpakket')"></head>
<body>
  <h1>Proefpakket.</h1>
  <blockquote>Een pakket bieren van een brouwer naar uw keuze.</blockquote>
  <img alt='brouwerij' th:src='@{/images/brouwerij.jpg}' class='fullwidth'>
</body>
</html>
```

2.1.5 proefpakket.css

```
body {
  background-color: #FFFCE6;
  font-family: sans-serif;
}
h1 {
color: brown;
}
```

```
.fullwidth {
  width: 100%;
}
blockquote {
  margin-left: 0;
  font-style: italic;
}
```

2.2 Brouwers

2.2.1 Adres

```
package be.vdab.proefpakket.valueobjects;
// enkele imports
@Embeddable
public class Adres implements Serializable {
  private static final long serialVersionUID = 1L;
  private String straat;
  private String huisNr;
  @ManyToOne(optional = false, fetch = FetchType.LAZY)
  @JoinColumn(name = "gemeenteid")
  private Gemeente gemeente;
  // getters voor straat, huisNr en gemeente
}
```

2.2.2 Gemeente

```
package be.vdab.proefpakket.entities;
// enkele imports
@Entity
@Table(name = "gemeenten")
public class Gemeente implements Serializable {
  private static final long serialVersionUID = 1L;
  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  private long id;
  private short postcode;
  private String naam;
}
```

2.2.3 Brouwer

```
package be.vdab.proefpakket.entities;
// enkele imports
@Entity
@Table(name="brouwers")
public class Brouwer implements Serializable {
  private static final long serialVersionUID = 1L;
  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  private long id;
  private String naam;
  @Embedded
  private Adres adres;
  private Long ondernemingsNr;
  // getters voor id, naam, adres en ondernemingsNr
}
```

2.2.4 BrouwerRepository

```
package be.vdab.proefpakket.repositories;
// enkele imports
public interface BrouwerRepository extends JpaRepository<Brouwer, Long> {
  List<Brouwer> findByNaamStartingWithOrderByNaam(String beginNaam);
}
```

2.2.5 BrouwerService

```
package be.vdab.proefpakket.services;
// enkele imports
public interface BrouwerService {
  List<Brouwer> findByBeginNaam(String beginNaam);
}
```

2.2.6 DefaultBrouwerService

```
package be.vdab.proefpakket.services;
// enkele imports
@Service
@Transactional(readOnly = true, isolation = Isolation.READ_COMMITTED)
class DefaultBrouwerService implements BrouwerService {
  private final BrouwerRepository brouwerRepository;
  DefaultBrouwerService(BrouwerRepository brouwerRepository) {
    this.brouwerRepository = brouwerRepository;
  }
  @Override
  public List<Brouwer> findByBeginNaam(String beginNaam) {
    return brouwerRepository.findByNaamStartingWithOrderByNaam(beginNaam);
  }
}
```

2.2.7 IndexController

```
package be.vdab.proefpakket.web;
// enkele imports
@Controller
@RequestMapping("/")
class IndexController {
  private static final String VIEW = "index";
  private static final char[] ALFABET =
    "ABCDEFGHIJKLMNOPQRSTUVWXYZ".toCharArray();
  private final BrouwerService brouwerService;
  IndexController(BrouwerService brouwerService) {
    this.brouwerService = brouwerService;
  }
  @GetMapping
  ModelAndView index() {
    return new ModelAndView(VIEW, "alfabet", ALFABET);
  }
  @GetMapping(params="letter")
  ModelAndView brouwers(String letter) {
    return new ModelAndView(VIEW, "alfabet", ALFABET)
      .addObject("brouwers", brouwerService.findByBeginNaam(letter));
  }
}
```

2.2.8 BrouwerController

```
package be.vdab.proefpakket.web;
// enkele imports
@Component
@RequestMapping("brouwers")
class BrouwerController {
  private static final String VIEW = "brouwers/brouwer";
  private static final String REDIRECT_BIJ_BROUWER_NIET_GEVONDEN = "redirect:/";
  @GetMapping("{brouwer}")
  ModelAndView read(@PathVariable Optional<Brouwer> brouwer,
    RedirectAttributes redirectAttributes) {
    if (brouwer.isPresent()) {
      return new ModelAndView(VIEW).addObject(brouwer.get());
```

```
    }
    redirectAttributes.addAttribute("fout", "Brouwer niet gevonden");
    return new ModelAndView(REDIRECT_BIJ_BROUWER_NIET_GEVONDEN);
  }
}
```

2.2.9 index.html

Uitbreiding:

```
<div th:if='${param.fout != null}' th:text='${param.fout}' class='fout'></div>
<div>Kies een brouwer:</div>
<ul id='alfabet'>
  <li th:each='letter:${alfabet}'>
    <a th:href='@{/(letter=${letter})}' th:text='${letter}'></a>
  </li>
</ul>
<ul>
  <li th:each='brouwer:${brouwers}'>
  <a th:href='@{/brouwers/{id}(id=${brouwer.id})}'
    th:text='${brouwer.naam}'></a></li>
</ul>
```

2.2.10 brouwer.html

```
<!doctype html>
<html lang='nl' xmlns:th="http://www.thymeleaf.org">
<head th:replace="fragments::head(title=${brouwer.naam})"></head>
<body>
  <h1 th:text='${brouwer.naam}'></h1>
  <dl>
    <dt>Straat</dt>
    <dd th:text='${brouwer.adres.straat}'></dd>
    <dt>Huisnummer</dt>
    <dd th:text='${brouwer.adres.huisNr}'></dd>
    <dt>Postcode</dt>
    <dd th:text='${brouwer.adres.gemeente.postcode}'></dd>
    <dt>Gemeente</dt>
    <dd th:text='${brouwer.adres.gemeente.naam}'></dd>
    <dt>Ondernemingsnummer</dt>
    <dd th:text='${brouwer.ondernemingsNr}'></dd>
  </dl>
</body>
</html>
```

2.2.11 proefpakket.css

Uitbreiding:

```
#alfabet {
  list-style-type: none;
  padding-left: 0;
}
#alfabet li {
  display:inline;
}
dd {
  margin-left:0;
  font-weight: bold;
}
```

2.3 Ondernemingsnummer

2.3.1 OndernemingsNr

```
package be.vdab.proefpakket.constraints;
// enkele imports
@Target({ FIELD, METHOD, ANNOTATION_TYPE })
@Retention(RUNTIME)
@Constraint(validatedBy = OndernemingsNrValidator.class)
public @interface OndernemingsNr {
  String message() default
    "{be.vdab.proefpakket.constraints.OndernemingsNummer.message}";
  Class<?>[] groups() default {};
  Class<? extends Payload>[] payload() default {};
}
```

2.3.2 OndernemingsNrValidator

```
package be.vdab.proefpakket.constraints;
// enkele imports uit javax.validation
public class OndernemingsNrValidator
  implements ConstraintValidator<OndernemingsNr, Long> {
  @Override
  public void initialize(OndernemingsNr postcode) {
  }
  @Override
  public boolean isValid(Long ondernemingsNr,
    ConstraintValidatorContext context) {
    if (ondernemingsNr == null) {
      return true;
    }
    long laatste2Cijfers = ondernemingsNr % 100L;
    long overigeCijfers= ondernemingsNr / 100;
    return laatste2Cijfers == 97 - overigeCijfers % 97;
  }
}
```

2.3.3 ValidationMessages.properties

```
javax.validation.constraints.NotNull.message=mag niet leeg zijn
be.vdab.proefpakket.constraints.OndernemingsNummer.message=ongeldig ondernemingsnummer
```

2.3.4 Brouwer

@OndernemingsNr voor variabele ondernemingsNr en setter toevoegen.

2.3.5 BrouwerService

Extra method declaratie:

```
void update(Brouwer brouwer);
```

2.3.6 DefaultBrouwerService

```
@Override
@Transactional(readOnly = false, isolation = Isolation.READ_COMMITTED)
public void update(Brouwer brouwer) {
  brouwerRepository.save(brouwer);
}
```

2.3.7 OndernemingsNrForm

```
package be.vdab.proefpakket.web;
// enkele imports
class OndernemingsNrForm {
  @NotNull
  @OndernemingsNr
  private Long ondernemingsNr; // plus een getter en een setter
}
```

2.3.8 BrouwerController

Extra code:

```
private final BrouwerService brouwerService;
BrouwerController(BrouwerService brouwerService) {
this.brouwerService = brouwerService;
}
private static final String ONDERNEMINGS_NR_VIEW="brouwers/ondernemingsnr";
private static final String REDIRECT_NA_ONDERNEMINGSNR =
  "redirect:/brouwers/{id}";
@GetMapping("{brouwer}/ondernemingsnr")
ModelAndView ondernemingsNr(@PathVariable Optional<Brouwer> brouwer,
  RedirectAttributes redirectAttributes) {
  if (brouwer.isPresent()) {
    OndernemingsNrForm form = new OndernemingsNrForm();
    form.setOndernemingsNr(brouwer.get().getOndernemingsNr());
    return new ModelAndView(ONDERNEMINGS_NR_VIEW)
     .addObject(brouwer.get())
     .addObject(form);
  }
  redirectAttributes.addAttribute("fout", "Brouwer niet gevonden");
  return new ModelAndView(REDIRECT_BIJ_BROUWER_NIET_GEVONDEN);
}
@PostMapping("{brouwer}/ondernemingsnr")
ModelAndView ondernemingsNr(@PathVariable Optional<Brouwer> brouwer,
  @Valid OndernemingsNrForm form, BindingResult bindingResult,
  RedirectAttributes redirectAttributes) {
  if (brouwer.isPresent()) {
    if (bindingResult.hasErrors()) {
      return new ModelAndView(ONDERNEMINGS_NR_VIEW).addObject(brouwer.get());
    }
    brouwer.get().setOndernemingsNr(form.getOndernemingsNr());
    brouwerService.update(brouwer.get());
    redirectAttributes.addAttribute("id", brouwer.get().getId());
    return new ModelAndView(REDIRECT_NA_ONDERNEMINGSNR);
  }
  redirectAttributes.addAttribute("fout", "Brouwer niet gevonden");
  return new ModelAndView(REDIRECT_BIJ_BROUWER_NIET_GEVONDEN);
}
```

2.3.9 brouwer.html

Extra code:

```
<div>
<a
th:href='@{/brouwers/{id}/ondernemingsnr(id=${brouwer.id})}'>Ondernemingsnummer
invullen</a>
</div>
```

2.3.10 ondernemingsnr.html

```
<!doctype html>
<html lang='nl' xmlns:th='http://www.thymeleaf.org'>
<head th:replace='fragments::head(title=|Ondernemingsnummer
${brouwer.naam}|)'></head>
<body>
  <h1 th:text='|Ondernemingsnummer ${brouwer.naam}|'></h1>
  <form method='post'
th:action='@{/brouwers/{id}/ondernemingsnr(id=${brouwer.id})}'
th:object='${ondernemingsNrForm}'>
  <label>Ondernemingsnummer:
  <span th:if="${#fields.hasErrors('ondernemingsNr')}"
th:errors='*{ondernemingsNr}'></span>
```

```
  <input type='number' th:field='*{ondernemingsNr}' required autofocus>
  </label>
  <input type='submit' value='Bewaren'>
  </form></body></html>
```

2.3.11 proefpakket.css

Uitbreiding:

```
label {
  cursor: pointer;
}
input, select {
  display: block;
  font-size: 1.1em;
  margin-bottom: 1em;
  margin-top: 0.2em;
}
input:focus, textarea:focus, select:focus {
  background-color: #FFFFD6
}

input[type='checkbox'], input[type='radio'] {
  display: inline;
}

label span, .fout {
  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;
  font-weight: bold;
}
label span:empty, .fout:empty {
  display: none;
}
```

2.4 Aanvraag

2.4.1 brouwer.html

Extra code:

```
<div>
<a th:href='@{/brouwers/{id}/proefpakket(id=${brouwer.id})}'>Proefpakket
aanvragen</a>
</div>
```

2.4.2 pom.xml

Extra dependency:

```
<dependency>
  <groupId>org.jsoup</groupId>
  <artifactId>jsoup</artifactId>
  <version>[1.10,]</version>
  <scope>runtime</scope>
</dependency>
```

2.4.3 Bestelling

```
package be.vdab.proefpakket.entities;
// enkele imports
@Entity
@Table(name = "bestellingen")
public class Bestelling implements Serializable {
  public interface Stap1 {}
  public interface Stap2 {}
  private static final long serialVersionUID = 1L;
  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  private long id;
  private LocalDate datum = LocalDate.now();
  @NotBlank(groups = Stap1.class)
  @SafeHtml(groups = Stap1.class)
  private String voornaam;
  @NotBlank(groups = Stap1.class)
  @SafeHtml(groups = Stap1.class)
  private String familienaam;
  @NotNull(groups = Stap1.class)
  @Email(groups = Stap1.class)
  private String emailAdres;
  @Valid
  @Embedded
  private Adres adres;
  @NotNull
  @ManyToOne(optional = false, fetch = FetchType.LAZY)
  @JoinColumn(name = "brouwerid")
  private Brouwer brouwer;
  // getters voor alle private variabelen (behalve serialVersionUID)
}
```

2.4.4 Adres

```
package be.vdab.proefpakket.valueobjects;
// enkele imports
@Embeddable
public class Adres implements Serializable {
  private static final long serialVersionUID = 1L;
  @NotBlank(groups = Bestelling.Stap2.class)
  @SafeHtml(groups = Bestelling.Stap2.class)
  private String straat;
  @NotBlank(groups = Bestelling.Stap2.class)
  @SafeHtml(groups = Bestelling.Stap2.class)
  private String huisNr;
  @ManyToOne(optional = false, fetch = FetchType.LAZY)
  @JoinColumn(name = "gemeenteid")
  @NotNull(groups = Bestelling.Stap2.class)
  private Gemeente gemeente;
  // getters voor straat, huisNr en gemeente
}
```

2.4.5 GemeenteRepository

```
package be.vdab.proefpakket.repositories;
// enkele imports
public interface GemeenteRepository extends JpaRepository<Gemeente, Long> {
}
```

2.4.6 GemeenteService

```
package be.vdab.proefpakket.services;
// enkele imports
public interface GemeenteService {
  List<Gemeente> findAll();
}
```

2.4.7 DefaultGemeenteService

```
package be.vdab.proefpakket.services;
// enkele imports
@Service
@Transactional(readOnly = true, isolation = Isolation.READ_COMMITTED)
class DefaultGemeenteService implements GemeenteService {
  private final GemeenteRepository gemeenteRepository;
  DefaultGemeenteService(GemeenteRepository gemeenteRepository) {
    this.gemeenteRepository = gemeenteRepository;
  }
  @Override
  public List<Gemeente> findAll() {
    return gemeenteRepository.findAll(Sort.by("naam","postcode"));
  }
}
```

2.4.8 BestellingRepository

```
package be.vdab.proefpakket.repositories;
// enkele imports
public interface BestellingRepository extends JpaRepository<Bestelling, Long> {}
```

2.4.9 BestellingService

```
package be.vdab.proefpakket.services;
import be.vdab.proefpakket.entities.Bestelling;
public interface BestellingService {
  void create(Bestelling bestelling);
}
```

2.4.10 DefaultBestellingService

```
package be.vdab.proefpakket.services;
// enkele imports
@Service
@Transactional(readOnly = true, isolation = Isolation.READ_COMMITTED)
class DefaultBestellingService implements BestellingService {
  private final BestellingRepository bestellingRepository;
  DefaultBestellingService(BestellingRepository bestellingRepository) {
    this.bestellingRepository = bestellingRepository;
  }
  @Override
  @Transactional(readOnly = false, isolation = Isolation.READ_COMMITTED)
  public void create(Bestelling bestelling) {
    bestellingRepository.save(bestelling);
  }
}
```

2.4.11 BrouwerController

Extra regel voor de class:

```
@SessionAttributes("bestelling")
```

Extra private variabelen:

```
private final GemeenteService gemeenteService;
private final BestellingService bestellingService;
```

Uitgebreide constructor:

```
BrouwerController(BrouwerService brouwerService,
  GemeenteService gemeenteService, BestellingService bestellingService) {
  this.brouwerService = brouwerService;
  this.gemeenteService = gemeenteService;
  this.bestellingService = bestellingService;
}
```

Extra code:

```
private static final String PROEFPAKKET_STAP_1_VIEW =
  "brouwers/proefpakketstap1";
@GetMapping("{brouwer}/proefpakket")
ModelAndView proefpakket(@PathVariable Optional<Brouwer> brouwer ,
  RedirectAttributes redirectAttributes) {
  if (brouwer.isPresent()) {
    return new ModelAndView(PROEFPAKKET_STAP_1_VIEW)
      .addObject(brouwer.get())
      .addObject(new Bestelling());
  }
  redirectAttributes.addAttribute("fout", "Brouwer niet gevonden");
  return new ModelAndView(REDIRECT_BIJ_BROUWER_NIET_GEVONDEN);
}
@InitBinder("bestelling")
void initBinder(DataBinder binder) {
  binder.initDirectFieldAccess();
}
private static final String PROEFPAKKET_STAP_2_VIEW =
  "brouwers/proefpakketstap2";
@PostMapping(value = "{brouwer}/proefpakket", params = "stap2")
ModelAndView proefpakketStap1NaarStap2(@PathVariable Optional<Brouwer> brouwer,
  @Validated(Bestelling.Stap1.class) Bestelling bestelling,
  BindingResult bindingResult , RedirectAttributes redirectAttributes) {
  if (brouwer.isPresent()) {
    if (bindingResult.hasErrors()) {
      return new ModelAndView(PROEFPAKKET_STAP_1_VIEW).addObject(brouwer.get());
    }
    return new ModelAndView(PROEFPAKKET_STAP_2_VIEW)
      .addObject(brouwer.get())
      .addObject("gemeenten", gemeenteService.findAll());
  }
  redirectAttributes.addAttribute("fout", "Brouwer niet gevonden");
  return new ModelAndView(REDIRECT_BIJ_BROUWER_NIET_GEVONDEN);
}
@PostMapping(value = "{brouwer}/proefpakket", params = "stap1")
ModelAndView proefpakketStap2NaarStap1(@PathVariable Optional<Brouwer> brouwer,
  Bestelling bestelling , RedirectAttributes redirectAttributes) {
  if (brouwer.isPresent()) {
    return new ModelAndView(PROEFPAKKET_STAP_1_VIEW).addObject(brouwer.get());
  }
  redirectAttributes.addAttribute("fout", "Brouwer niet gevonden");
  return new ModelAndView(REDIRECT_BIJ_BROUWER_NIET_GEVONDEN);
}
private static final String REDIRECT_NA_BESTELLING = "redirect:/";
@PostMapping(value = "{brouwer}/proefpakket", params = "opslaan")
ModelAndView proefpakketOpslaan(@PathVariable Optional<Brouwer> brouwer,
  @Validated(Bestelling.Stap2.class) Bestelling bestelling,
  BindingResult bindingResult, SessionStatus sessionStatus ,
  RedirectAttributes redirectAttributes) {
  if (brouwer.isPresent()) {
    if (bindingResult.hasErrors()) {
```

```
      return new ModelAndView(PROEFPAKKET_STAP_2_VIEW)
        .addObject(brouwer.get())
        .addObject("gemeenten", gemeenteService.findAll());
    }
    bestellingService.create(bestelling);
    sessionStatus.setComplete();
    return new ModelAndView(REDIRECT_NA_BESTELLING);
  }
  redirectAttributes.addAttribute("fout", "Brouwer niet gevonden");
  return new ModelAndView(REDIRECT_BIJ_BROUWER_NIET_GEVONDEN);
}
```

2.4.12 proefpakketstap1.html

```
<!doctype html>
<html lang='nl' xmlns:th='http://www.thymeleaf.org'>
<head
th:replace='fragments::head(title=|Proefpakket ${brouwer.naam} (stap 1) |)'>
</head>
<body>
  <h1 th:text='|Proefpakket ${brouwer.naam} (stap 1) |'></h1>
  <form method='post'
    th:action='@{/brouwers/{id}/proefpakket(id=${brouwer.id})}'
    th:object='${bestelling}'>
  <label>Voornaam:
  <span th:if="${#fields.hasErrors('voornaam')}" th:errors='*{voornaam}'></span>
  <input th:field='*{voornaam}' required autofocus>
  </label>
  <label>Familienaam:
  <span th:if="${#fields.hasErrors('familienaam')}"
    th:errors='*{familienaam}'></span>
  <input th:field='*{familienaam}' required>
  </label>
  <label>Email adres:
  <span th:if="${#fields.hasErrors('emailAdres')}"
    th:errors='*{emailAdres}'></span>
  <input type='email' th:field='*{emailAdres}' required>
  </label>
  <input type='submit' value='Stap 2' name='stap2'>
  </form>
</body>
</html>
```

2.4.13 proefpakketstap2.html

```
<!doctype html>
<html lang='nl' xmlns:th='http://www.thymeleaf.org'>
<head
  th:replace='fragments::head(title=|Proefpakket ${brouwer.naam} (stap 2) |)'>
</head>
<body>
  <h1 th:text='|Proefpakket ${brouwer.naam} (stap 2) |'></h1>
  <form method='post'
    th:action='@{/brouwers/{id}/proefpakket(id=${brouwer.id})}'
    th:object='${bestelling}'>
  <label>Straat:
  <span th:if="${#fields.hasErrors('adres.straat')}"
    th:errors='*{adres.straat}'></span>
  <input th:field='*{adres.straat}' required autofocus>
  </label>
  <label>Huisnummer:
  <span th:if="${#fields.hasErrors('adres.huisNr')}"
    th:errors='*{adres.huisNr}'></span>
  <input th:field='*{adres.huisNr}' required>
```

```
  </label>
  <label>Gemeente:
  <span th:if="${#fields.hasErrors('adres.gemeente')}"
    th:errors='*{adres.gemeente}'></span>
  <select th:field='*{adres.gemeente}'>
  <option th:each='gemeente: ${gemeenten}' th:value='${gemeente.id}'
    th:text='|${gemeente.naam} ${gemeente.postcode}|'></option>
  </select>
  </label>
  <input type='submit' value='Stap 1' name='stap1' formnovalidate>
  <input type='submit' value='Opslaan' name='opslaan'>
  </form>
</body>
</html>
```

2.4.14 ValidationMessages.properties

Extra regels:

```
javax.validation.constraints.Email.message=ongeldig e-mail adres
javax.validation.constraints.NotBlank.message=\
moet meer dan enkel spaties bevatten
org.hibernate.validator.constraints.SafeHtml.message=mag geen script bevatten
```

2.4.15 application.properties

```
server.session.tracking-modes=cookie
```

2.5 Meertalig

2.5.1 messages.properties

```
proefpakket=Proefpakket.
omschrijvingProefpakket=Een pakket bieren van een brouwer naar uw keuze.
brouwerij=brouwerij
kiesEenBrouwer=Kies een brouwer:
```

2.5.2 messages_fr.properties

```
proefpakket=Paquet d'essai.
omschrijvingProefpakket=Un paquet de bières d'un brasseur de votre choix.
brouwerij=brasserie
kiesEenBrouwer=Choisissez un brasseur:
```

2.5.3 WebConfig

```
package be.vdab.proefpakket.web;
// enkele imports
@Configuration
class WebConfig implements WebMvcConfigurer {
  @Bean
  CookieLocaleResolver localeResolver() {
    CookieLocaleResolver resolver = new CookieLocaleResolver();
    resolver.setCookieMaxAge(604_800);
    return resolver;
  }
  @Override
  public void addInterceptors(InterceptorRegistry registry) {
    registry.addInterceptor(new LocaleChangeInterceptor());
  }
}
```

2.5.4 Index.html

```
<!doctype html>
<html lang='nl' xmlns:th="http://www.thymeleaf.org">
<head th:replace="fragments::head(title=#{proefpakket})"></head>
```

```
<body>
  <h1 th:text='#{proefpakket}'></h1>
  <blockquote th:text='#{omschrijvingProefpakket}'></blockquote>
  <img th:alt='#{brouwerij}' th:src='@{/images/brouwerij.jpg}' class='fullwidth'>
  <div th:text='#{kiesEenBrouwer}'></div>
  <ul id='alfabet'>
  <li th:each='letter:${alfabet}'>
  <a th:href='@{/(letter=${letter})}' th:text='${letter}'></a>
  </li>
  </ul>
<ul>
  <li th:each='brouwer:${brouwers}'>
  <a th:href='@{/brouwers/{id}(id=${brouwer.id})}'
    th:text='${brouwer.naam}'></a></li>
</ul>
<a th:href='@{?locale=nl_BE}'>Nederlands</a> <a
th:href='@{?locale=fr_BE}'>Français</a>
</body>
</html>
```

2.5.5 application.properties

```
spring.messages.fallback-to-system-locale=false
```

2.6 REST

2.6.1 Gemeente

Extra regels voor de class:

```
@XmlRootElement
@XmlAccessorType(XmlAccessType.FIELD)
```

2.6.2 GemeenteNietGevondenException

```
package be.vdab.proefpakket.exceptions;
public class GemeenteNietGevondenException extends RuntimeException {
  private static final long serialVersionUID = 1L;
}
```

2.6.3 GemeenteRestController

```
package be.vdab.proefpakket.restservices;
// enkele imports
@RestController
@RequestMapping("/gemeenten")
class GemeenteRestController {
  @GetMapping("{gemeente}")
  Gemeente read(@PathVariable Optional<Gemeente> gemeente) {
    if (gemeente.isPresent()) {
      return gemeente.get();
    }
    throw new GemeenteNietGevondenException();
  }
  @ExceptionHandler(GemeenteNietGevondenException.class)
  @ResponseStatus(HttpStatus.NOT_FOUND)
  void gemeenteNietGevonden() {
  }
}
```

2.7 Temperatuur

2.7.1 application.properties

Extra regel:

```
openWeatherMapURL=http://api.openweathermap.org/data/2.5/weather\
?q={plaats}&units=metric&APPID=yyy
```

2.7.2 Weer

```
package be.vdab.proefpakket.restclients;
class Weer {
  private Main main; // en een getter en een setter
}
```

2.7.3 Main

```
package be.vdab.proefpakket.restclients;
class Main {
  private BigDecimal temp; // en een getter en een setter
}
```

2.7.4 WeerClient

```
package be.vdab.proefpakket.restclients;
import java.math.BigDecimal;
public interface WeerClient {
  BigDecimal getTemperatuur(String plaats);
}
```

2.7.5 KanTemperatuurNietLezenException

```
package be.vdab.proefpakket.exceptions;
public class KanTemperatuurNietLezenException extends RuntimeException {
  private static final long serialVersionUID = 1L;
}
```

2.7.6OpenWeatherMapClient

```
package be.vdab.proefpakket.restclients;
// enkele imports
@Component
class OpenWeatherMapClient implements WeerClient {
  private static final Logger LOGGER =
    LoggerFactory.getLogger(OpenWeatherMapClient.class);
  private final String uriTemplate;
  private final RestTemplate restTemplate;
  OpenWeatherMapClient(@Value("${openWeatherMapURL}") String uriTemplate,
    RestTemplateBuilder restTemplateBuilder) {
    this.uriTemplate = uriTemplate;
    this.restTemplate = restTemplateBuilder.build();
  }
  @Override
  public BigDecimal getTemperatuur(String plaats) {
    try {
      return restTemplate.getForObject(uriTemplate, Weer.class, plaats)
        .getMain().getTemp();
    } catch (Exception ex) {
      LOGGER.error("kan temperatuur niet lezen", ex);
      throw new KanTemperatuurNietLezenException();
  }
}
```

2.7.7 WeerService

```
package be.vdab.proefpakket.services;
import java.math.BigDecimal;
public interface WeerService {
  BigDecimal getTemperatuur(String plaats);
}
```

2.7.8 DefaultWeerService

```
package be.vdab.proefpakket.services;
// enkele imports
@Service
class DefaultWeerService implements WeerService {
  private final WeerClient weerClient;
  DefaultWeerService(WeerClient weerClient) {
    this.weerClient = weerClient;
  }
  @Override
  public BigDecimal getTemperatuur(String plaats) {
    return weerClient.getTemperatuur(plaats);
  }
}
```

2.7.9 WeerController

```
package be.vdab.proefpakket.web;
// enkele imports
@Controller
@RequestMapping("weer")
class WeerController {
  private static final String VIEW = "weer/temperatuur";
  private final WeerService weerService;
  WeerController(WeerService weerService) {
    this.weerService = weerService;
  }
  @GetMapping("{plaats}/temperatuur")
  ModelAndView temperatuur(@PathVariable String plaats) {
    ModelAndView modelAndView = new ModelAndView(VIEW);
    try {
      modelAndView.addObject("temperatuur", weerService.getTemperatuur(plaats));
    } catch (KanTemperatuurNietLezenException ex) {
       modelAndView.addObject("fout", "kan temperatuur niet lezen");
    }
    return modelAndView;
  }
}
```

2.7.10 brouwer.html

extra code:

```
<div>
<a
th:href='@{/weer/{plaats}/temperatuur(plaats=${brouwer.adres.gemeente.naam})}'>
Temperatuur</a>
</div>
```

2.7.11 temperatuur.html

```
<!doctype html>
<html lang='nl' xmlns:th="http://www.thymeleaf.org">
<head th:replace="fragments::head(title=|${plaats}: ${temperatuur}°|)"></head>
<body>
  <h1 th:if='${temperatuur != null}'
    th:text='|Het is ${temperatuur}° in ${plaats}.|'></h1>                    ❶
  <div class='fout' th:if='${fout != null}' th:text='${fout}'></div>
</body>
</html>
```

(1) Spring geeft alle `@PathVariable` parameters uit de Controller method automatisch door aan de view. `${plaats}` verwijst zo naar de parameter `@PathVariable String plaats`

2.8 Mail

2.8.1 application.properties

```
spring.mail.host=smtp.gmail.com
spring.mail.port=465
spring.mail.protocol=smtps
spring.mail.username=eenGebruikersNaam@gmail.com
spring.mail.password=paswoordVanDieGebruiker
```

2.8.2 MailSender

```
package be.vdab.proefpakket.mail;
import be.vdab.proefpakket.entities.Bestelling;
public interface MailSender {
  void proefpakket(String emailAdres, String brouwerNaam);
}
```

2.8.3 KanMailNietZendenException

```
package be.vdab.proefpakket.exceptions;
public class KanMailNietZendenException extends RuntimeException {
  private static final long serialVersionUID = 1L;
}
```

2.8.4 DefaultMailSender

```
package be.vdab.proefpakket.mail;
// enkele imports
@Component
class DefaultMailSender implements MailSender {
  private static final Logger LOGGER =
    LoggerFactory.getLogger(DefaultMailSender.class);
  private final JavaMailSender sender;
  DefaultMailSender(JavaMailSender sender) {
    this.sender = sender;
  }
  @Override
  public void proefpakket(String emailAdres, String brouwerNaam) {
    try {
      SimpleMailMessage message = new SimpleMailMessage();
      message.setTo(emailAdres);
      message.setSubject("Proefpakket " + brouwerNaam);
      message.setText("Bedankt voor uw interesse. U ontvangt uw proefpakket " +
        brouwerNaam + " binnenkort.");
      sender.send(message);
    } catch (MailException ex) {
      LOGGER.error("Kan mail proefpakket niet versturen", ex);
      throw new KanMailNietZendenException();
    }
  }
}
```

2.8.5 DefaultBestellingService

```
package be.vdab.proefpakket.services;
// enkele imports
@Service
@Transactional(readOnly = true, isolation = Isolation.READ_COMMITTED)
class DefaultBestellingService implements BestellingService {
  private final BestellingRepository bestellingRepository;
  private final MailSender mailSender;
  DefaultBestellingService(BestellingRepository bestellingRepository,
    MailSender mailSender) {
    this.bestellingRepository = bestellingRepository;
    this.mailSender = mailSender;
  }
```

```
  @Override
  @Transactional(readOnly = false, isolation = Isolation.READ_COMMITTED)
  public void create(Bestelling bestelling) {
    bestellingRepository.save(bestelling);
    mailSender.proefpakket(bestelling.getEmailAdres(),
      bestelling.getBrouwer.getNaam());
  }
}
```

2.9 JMS

2.9.1 application.properties

Extra regels:

```
spring.activemq.broker-url=tcp://localhost:61616
proefpakketQueue=proefpakketQueue
```

2.9.2 ProefpakketMessage

```
package be.vdab.proefpakket.messaging;
// enkele imports
@XmlRootElement
@XmlAccessorType(XmlAccessType.FIELD)
public class ProefpakketMessage {
  private String emailAdres;
  private String brouwerNaam;
  // je maakt getters en een geparametriseerde constructor
  // je maakt ook een protected default constructor die JAXB nodig heeft
}
```

2.9.3 pom.xml

```
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-oxm</artifactId>
</dependency>
```

2.9.4 MessagingConfig

```
package be.vdab.proefpakket.messaging;
// enkele imports
@Configuration
class MessagingConfig {
  @Bean
  Jaxb2Marshaller marshaller() {
    Jaxb2Marshaller marshaller = new Jaxb2Marshaller();
    marshaller.setClassesToBeBound(ProefpakketMessage.class);
    return marshaller;
  }
  @Bean
  MarshallingMessageConverter converter (
    Jaxb2Marshaller marshaller) {
    return new MarshallingMessageConverter(marshaller, marshaller);
  }
  @Bean
  DefaultJmsListenerContainerFactory factory(
    ConnectionFactory connectionFactory,
    MarshallingMessageConverter converter) {
    DefaultJmsListenerContainerFactory factory =
      new DefaultJmsListenerContainerFactory();
    factory.setConnectionFactory(connectionFactory);
    factory.setMessageConverter(converter);
    return factory;
  }
}
```

2.9.5 DefaultBestellingService

```
package be.vdab.proefpakket.services;
// enkele imports
@Service
@Transactional(readOnly = true, isolation = Isolation.READ_COMMITTED)
class DefaultBestellingService implements BestellingService {
  private final BestellingRepository bestellingRepository;
  private final JmsTemplate jmsTemplate;
  private final String proefpakketQueue;
  DefaultBestellingService(BestellingRepository bestellingRepository,
    JmsTemplate jmsTemplate,
    @Value("${proefpakketQueue}") String proefpakketQueue) {
    this.bestellingRepository = bestellingRepository;
    this.jmsTemplate = jmsTemplate;
    this.proefpakketQueue = proefpakketQueue;
  }
  @Override
  @Transactional(readOnly = false, isolation = Isolation.READ_COMMITTED)
  public void create(Bestelling bestelling) {
    bestellingRepository.save(bestelling);
    ProefpakketMessage message = new ProefpakketMessage(
      bestelling.getEmailAdres(),
      bestelling.getBrouwer().getNaam());
    jmsTemplate.convertAndSend(proefpakketQueue, message);
  }
}
```

2.9.6 ProefpakketListener

```
package be.vdab.proefpakket.messaging;
// enkele imports
@Component
class ProefpakketListener {
  private final MailSender mailSender;
  ProefpakketListener(MailSender mailSender) {
    this.mailSender = mailSender;
  }
  @JmsListener(destination = "${proefpakketQueue}")
  void ontvangBoodschap(ProefpakketMessage message) {
    mailSender.proefpakket(message.getEmailAdres(), message.getBrouwerNaam());
  }
}
```

2.10 Security

2.10.1 Table Users en Authorities

Zelfde structuur als in theorie.
Vergeet niet de gebruiker `cursist` leesrechten te geven op deze tables.

2.10.2 SecurityConfig

```
package be.vdab.proefpakket.security;
// enkele imports
@EnableWebSecurity
class SecurityConfig extends WebSecurityConfigurerAdapter {
  private static final String ADMINISTRATOR = "administrator";
  @Bean JdbcDaoImpl jdbcDaoImpl(DataSource dataSource) {
    JdbcDaoImpl impl = new JdbcDaoImpl();
    impl.setDataSource(dataSource);
    return impl;
  }
```

```
  @Override
  public void configure(WebSecurity web) throws Exception {
  web.ignoring()
    .mvcMatchers("/images/**")
    .mvcMatchers("/css/**")
    .mvcMatchers("/scripts/**");
  }
  @Override
  protected void configure(HttpSecurity http) throws Exception {
    http.formLogin()
      .and().authorizeRequests().mvcMatchers("/brouwers/*/ondernemingsnr")
      .hasAuthority(ADMINISTRATOR);
  }
}
```

3 COLOFON

Domeinexpertisemanager: Jean Smits

Moduleverantwoordelijke: Jean Smits

Medewerkers: Hans Desmet

Versie: 29/6/2018

Nummer dotatielijst: