

Application Lifecycle Management



Rodrigo Martins Pagliares

Application Lifecycle Management



Agenda

- 1. Downloading/Cloning the HostelApp**
- 2. Opening the HostelApp in IntelliJ Ultimate Edition**
- 3. The **backend** module of HostelApp**
- 4. The **frontend** module of HostelApp**
- 5. Running the **backend** module of HostelApp**
- 6. Running the **frontend** module of HostelApp**
- 7. Services view in IntelliJ**
- 8. Online Demo**
- 9. What's **next**?**

Downloading/Cloning the HostelApp



pagliares / HostelApp

Type ⌘ to search



Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

HostelApp

Public

Pin

Unwatch 1

Fork 0

Star 0

main

1 Branch

0 Tags

Go to file

Add file

Code



pagliares Update README.md

5cbee20 · 37 minutes ago

6 Commits

.idea

First working version of HostelApp

1 hour ago

Hostel_App_Back_End_API

First working version of HostelApp

1 hour ago

Hostel_App_Front_End_MVC

First working version of HostelApp

1 hour ago

Images

Include landing page screenshot to be used in README.md

46 minutes ago

Slides

First working version of HostelApp

1 hour ago

.gitignore

First working version of HostelApp

1 hour ago

README.md

Update README.md

37 minutes ago

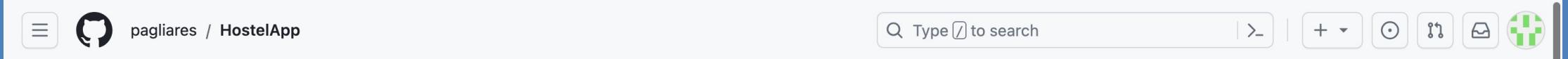
README



HostelApp - a web application for the Hostel Sparkling Water

This repository contains the source-code and Manamegent - ALM - at the undeargraduate C

<https://github.com/pagliares/HostelApp>



HostelApp

Public

main · 1 Branch · 0 Tags

Go to file

Add file

Code

pagliares Update README.md



First working version



First working version



First working version



Include landing page



First working version



First working version



40 minutes ago



Upda...



40 minutes ago



40 minutes ago



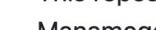
40 minutes ago



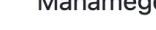
40 minutes ago



40 minutes ago



40 minutes ago



40 minutes ago

Local

Codespaces

Clone

HTTPS SSH GitHub CLI

<https://github.com/pagliares/HostelApp.git>

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Download ZIP

About

No description, website, or topics provided.

Readme

Activity

0 stars

1 watching

0 forks

Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

HostelApp Waterfall

This repository is managed by ALM at the undergraduate Computer Science course at UNIFAL - Alfenas - MG - Brazil.

Download or clone the repository. I will download in this demonstration.



IntelliJ IDEA



Eclipse

Apache NetBeans
17

SpringToolSuite4



Visual Studio Code



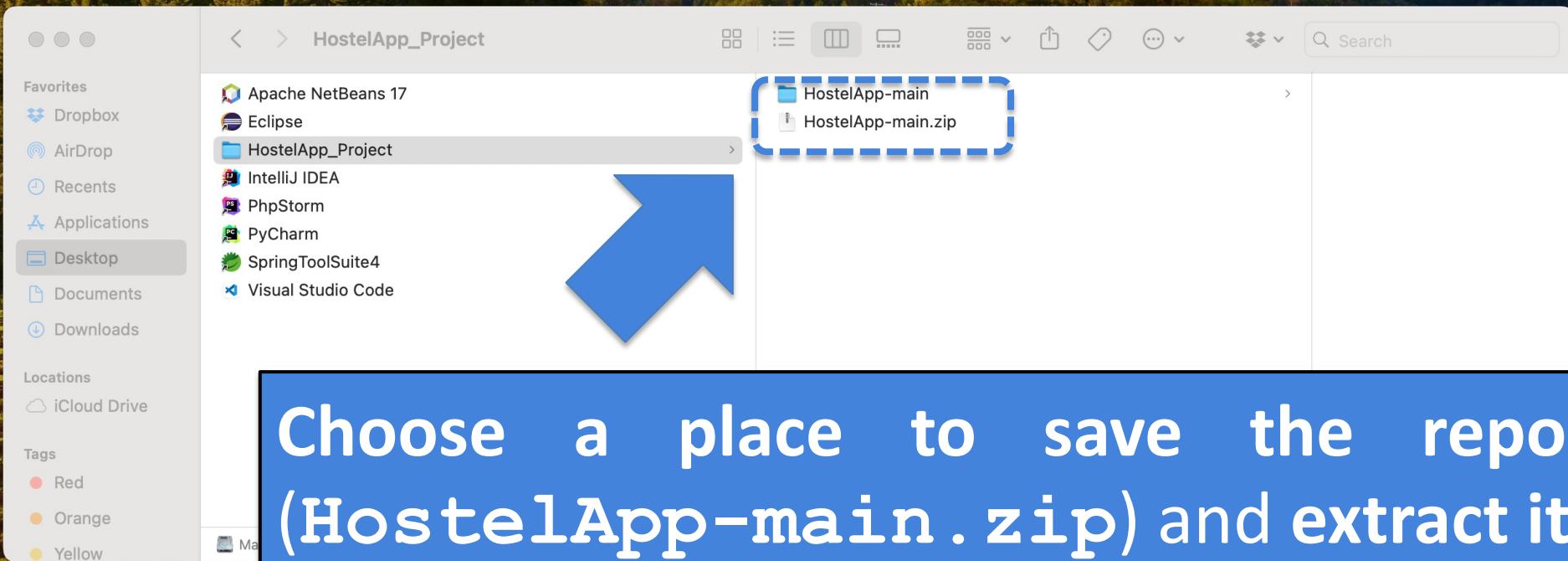
PyCharm



PhpStorm



HostelApp_Project



Choose a place to save the repository
(**HostelApp-main.zip**) and extract it.

IntelliJ IDEA





IntelliJ IDEA



Eclipse



Apache NetBeans



SpringToolSuite4



Visual Studio Code



PyCharm

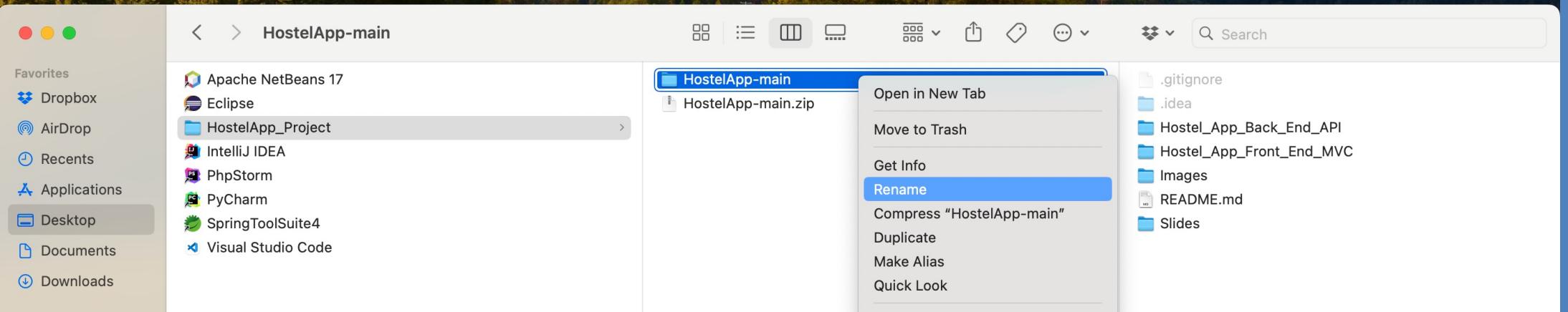


PhpStorm

17



HostelApp_Project



"-main" is a **suffix** that **indicates** this **project** was **downloaded** from a **main branch** in **Github**. We can **rename** the **directory name**, **removing** the **suffix**.



IntelliJ IDEA



Eclipse



Apache NetBeans



SpringToolSuite4

17



Visual Studio Code



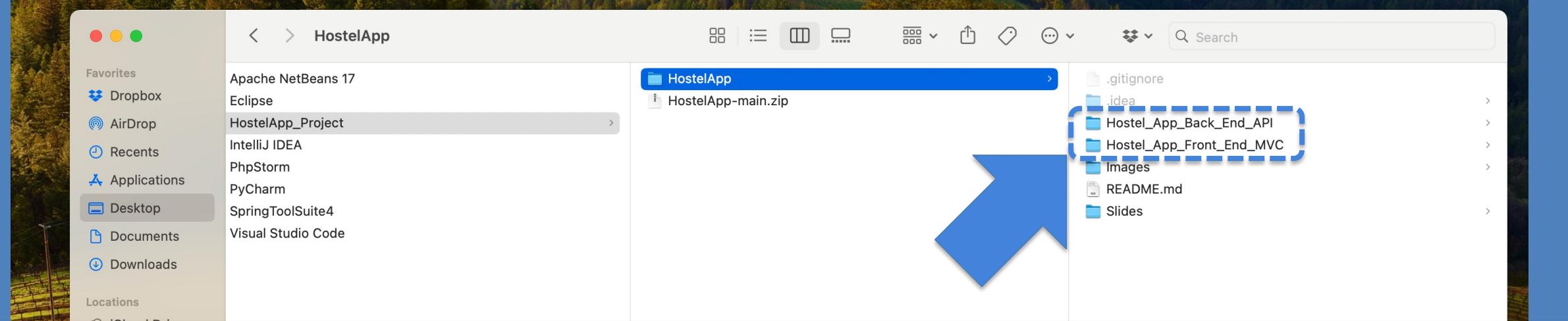
PyCharm



PhpStorm



HostelApp_Project



HostelApp is the **folder** we are going to **open** in **IntelliJ**. It is a **multi-module** Maven project contains the **two modules** that constitutes the **application**:

(Hostel_App_Back_End_API e Hostel_App_Front_End_MVC)



IntelliJ IDEA



Eclipse



Apache NetBeans

SpringToolSuite4
17

Visual Studio Code



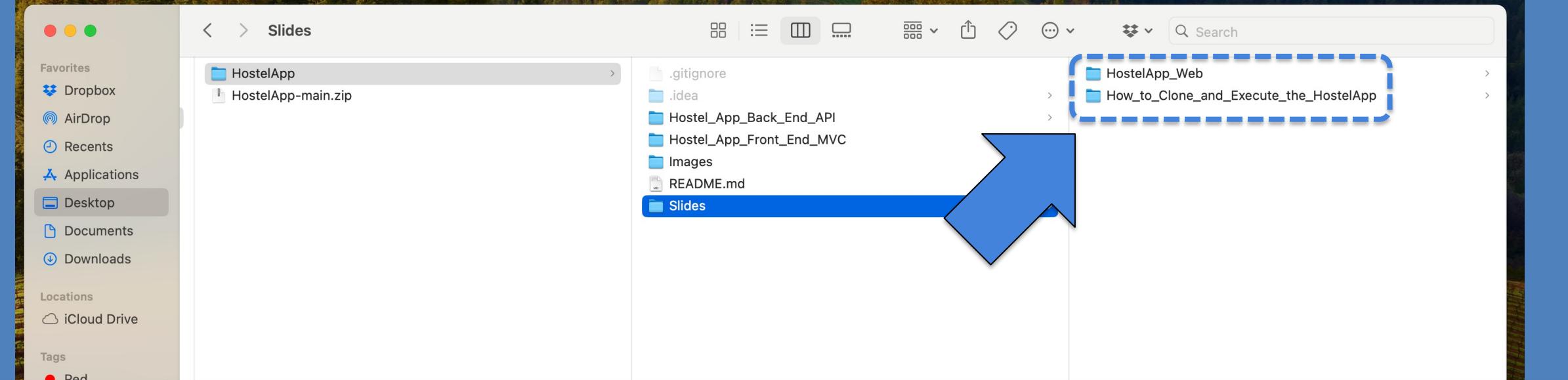
PyCharm



PhpStorm

Screenshot
2024-01...10.41.48

HostelApp_Project



The **folder** HostelApp also **contains** some **Slides** with **step-by-step instructions** on how do **clone/download** and **execute** the HostelApp (this slides) and **slides** demonstrating **how to use HostelApp**.

Opening the HostelApp in IntelliJ Ultimate Edition

IntelliJ IDEA

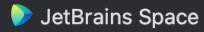
Ultimate 2023.2

JET
BRAINS



Projects

Remote Development Beta



Customize

Plugins 2

Learn

Welcome to IntelliJ IDEA

Create a new project to start from scratch.

Open existing project from disk or version control.



New Project

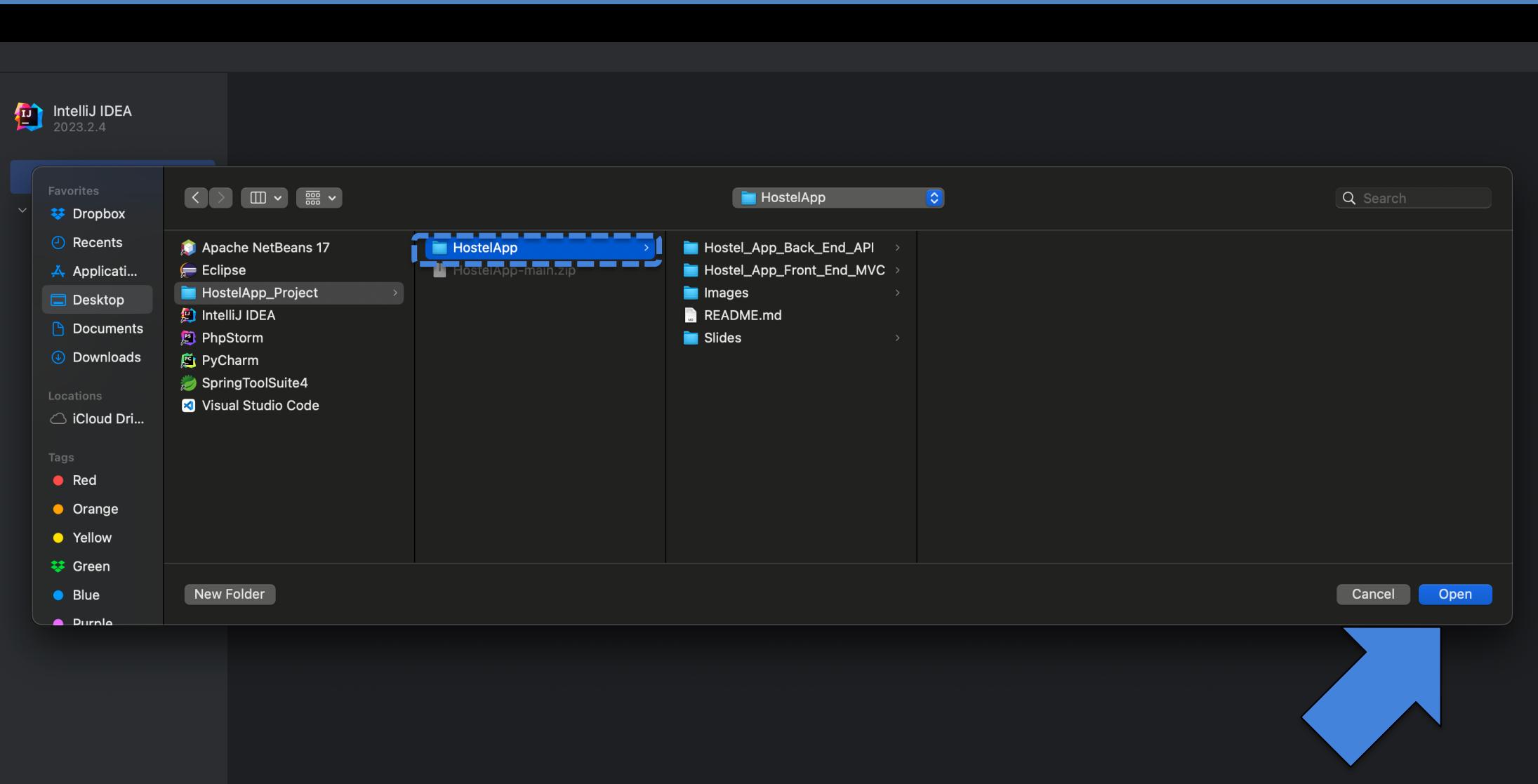


Open



Get from VCS



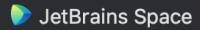


Navigate to the HostelApp directory and click Open.



Projects

Remote Development Beta



Customize

Plugins

Learn

Welcome to IntelliJ IDEA

Create a new project to start from scratch.

Open existing project from disk or version control.



Trust and Open Project 'HostelApp'?

IntelliJ IDEA provides features that may execute potentially malicious code from this folder.

If you don't trust the source, preview the project in the safe mode to only browse its code.

Trust projects in ~/Desktop/HostelApp_Project



Don't Open

Preview in Safe Mode

Trust Project

Click Trust Project.



The project is now open.

HA HostelApp Version control CustomerServicesApplication

Project HostelApp ~/Desktop/HostelApp_Project/HostelApp .idea Hostel_App_Back_End_API [customer-services] Hostel_App_Front_End_MVC [customer-app] Images Slides .gitignore README.md External Libraries Scratches and Consoles

M README.md

HostelApp - a web application for the Hostel Sparkling Water

This repository contains the source-code and several other artifacts used in the course Application Lifecycle Management - ALM - at the undergraduate Computer Science course at UNIFAL - Alfenas - MG - Brazil.

localhost:8080

Sparkling Water Hostel Customers Login

Sparkling Water hostel

The *hostel Sparkling Water* is a fictitious hostel located in Caxambu-MG, Brazil. Caxambu is a small town (its population in 2020 was estimated at 22,000), renowned for its spa which has twelve sources of sparkling mineral water flowing 24 hours a day and a cold-water geyser. Caxambu is the only place in the world with 12 mineral water springs, each with different chemical properties.

Park

The Water Park is the main attraction in town.



Photo of the bathhouse located at the Water Park

How to get to the hostel?

From Rio de Janeiro
Via BR-116 (Presidente Dutra) west past Resende-RJ to the village of Engenheiro Passos-RJ and take BR-354 north.

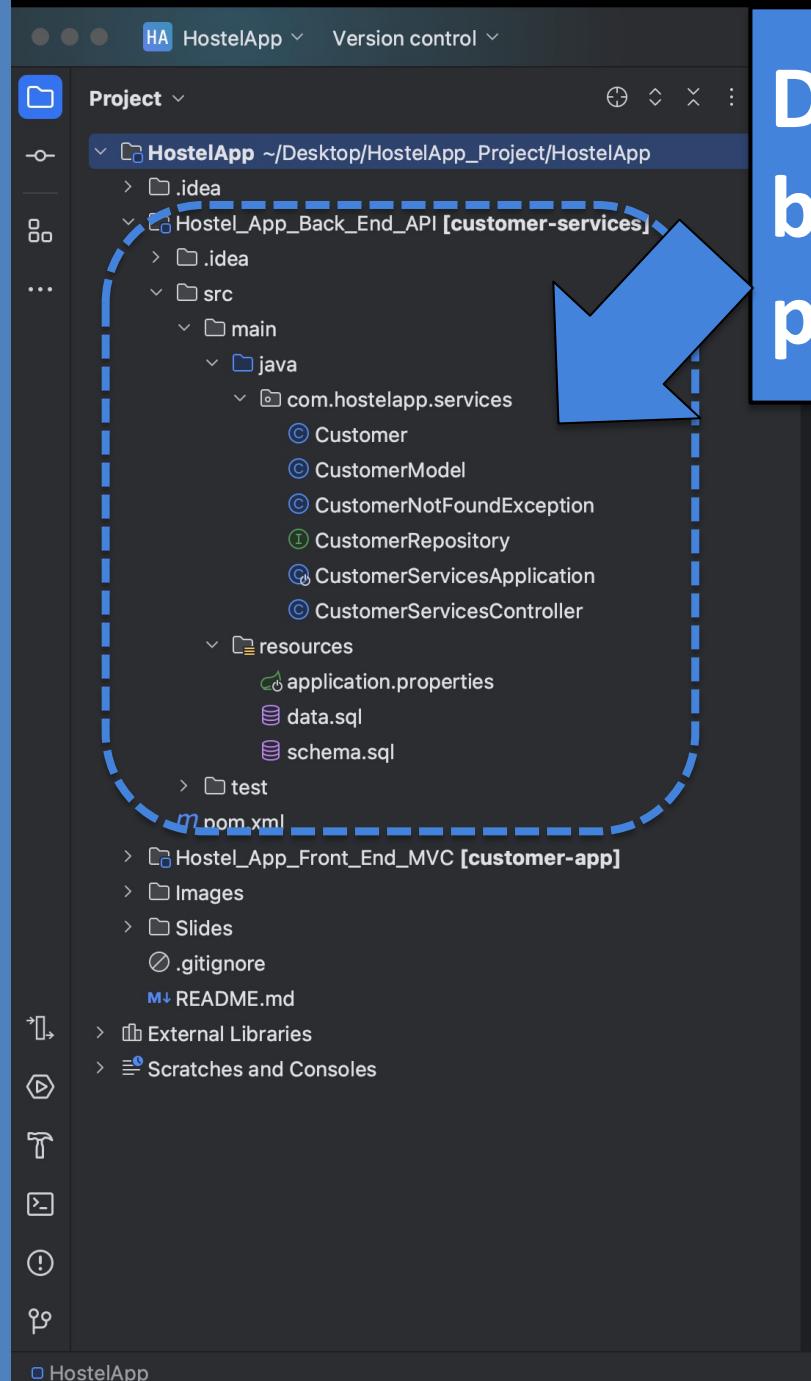
From Belo Horizonte or São Paulo
Via BR-381 to Campanha-MG, then west on BR-267.

Famous tourism quotes

"Never go on trips with anyone you do not love." Ernest Hemingway
"To travel is to discover that everyone is wrong about other countries." Aldous Huxley

1:1 LF UTF-8 4 spaces

The **backend** module of
HostelApp



Directory structure and files for the backend module. The structure of the project is based on by Apache Maven.

This repository contains the source-code and several other artifacts used in the course Application Lifecycle Management - ALM - at the undergraduate Computer Science course at UNIFAL - Alfenas - MG - Brazil.

localhost:8080

Sparkling Water Hostel Customers Login

Sparkling Water hostel

The *hostel Sparkling Water* is a fictitious hostel located in Caxambu-MG, Brazil. Caxambu is a small town (its population in 2020 was estimated at 22,000), renowned for its spa which has twelve sources of sparkling mineral water flowing 24 hours a day and a cold-water geyser.

Caxambu is the only place in the world with 12 mineral water springs, each with different chemical properties.

Park

The Water Park is the main attraction in town.



Photo of the bathhouse located at the Water Park

How to get to the hostel?

From Rio de Janeiro
Via BR-116 (Presidente Dutra) west past Resende-RJ to the village of Engenheiro Passos-RJ and take BR-354 north.

From Belo Horizonte or São Paulo
Via BR-381 to Campanha-MG, then west on BR-267.

Famous tourism quotes

"Never go on trips with anyone you do not love." *Ernest Hemingway*

"To travel is to discover that everyone is wrong about other countries." *Aldous Huxley*

Project

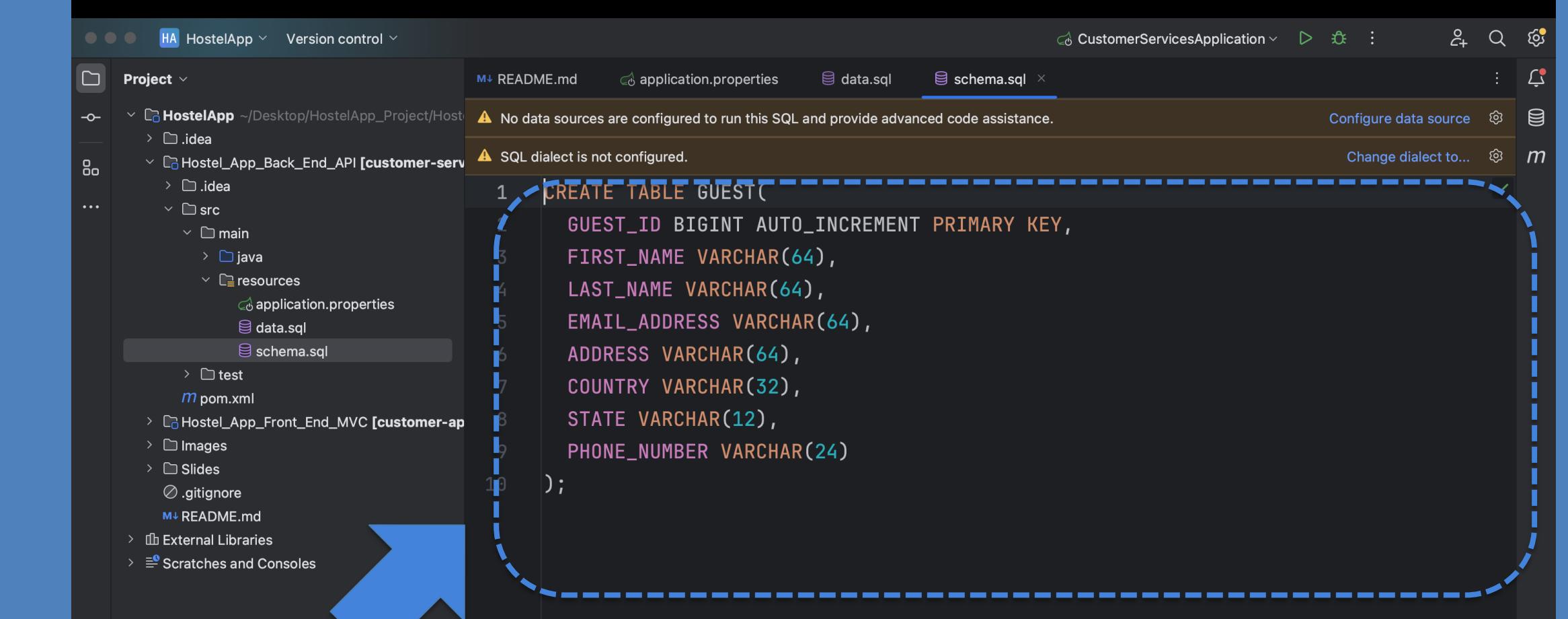
- HostelApp ~/Desktop/HostelApp_Project/HostelApp
- .idea
- Hostel_App_Back_End_API [customer-service]
 - .idea
 - src
 - main
 - java
 - resources
 - application.properties
 - data.sql
 - schema.sql
 - test
 - pom.xml
- Hostel_App_Front_End_MVC [customer-app]
- Images
- Slides
- .gitignore
- README.md
- External Libraries
- Scratches and Consoles

README.md application.properties

```
1 spring.jpa.hibernate.ddl-auto=none
2 server.port=8100
```



The backend module runs on port 8100.



The screenshot shows a Java IDE interface with a dark theme. On the left, the project structure for 'HostelApp' is visible, including a 'Hostel_App_Back_End_API' module which contains a 'src' folder with 'main' and 'resources' subfolders, and a 'schema.sql' file. The 'schema.sql' file is currently selected and open in the main editor area. The code in the editor is:

```
CREATE TABLE GUEST(
    GUEST_ID BIGINT AUTO_INCREMENT PRIMARY KEY,
    FIRST_NAME VARCHAR(64),
    LAST_NAME VARCHAR(64),
    EMAIL_ADDRESS VARCHAR(64),
    ADDRESS VARCHAR(64),
    COUNTRY VARCHAR(32),
    STATE VARCHAR(12),
    PHONE_NUMBER VARCHAR(24)
);
```

At the top of the editor, there are two warning messages: 'No data sources are configured to run this SQL and provide advanced code assistance.' and 'SQL dialect is not configured.' There are also buttons for 'Configure data source' and 'Change dialect to...'. A large blue arrow points from the bottom text block to the 'schema.sql' file in the project tree.

The **backend module has a database that stores customers records**. We name the **table GUEST**. HostelApp uses a H2 embedded database. This SQL script is configured to run every time you restart the backend module.

This is the data inserted into the database table GUEST when we run the application.

The screenshot shows a code editor with a dark theme. On the left is a file tree for a Java project named 'Hostel_App_Front_End_MVC [customer-ap]'. The 'resources' folder contains 'application.properties', 'data.sql', and 'schema.sql'. A blue arrow points from the 'data.sql' file in the tree to the code editor. The code editor displays several 'INSERT INTO GUEST' statements, each with a line number and a dashed blue oval highlighting the entire statement. The statements insert data for Holden Caulfield, Humbert Humbert, Leopold Bloom, Rabbit Angstrom, Sherlock Holmes, Atticus Finch, and another guest whose first name is partially visible.

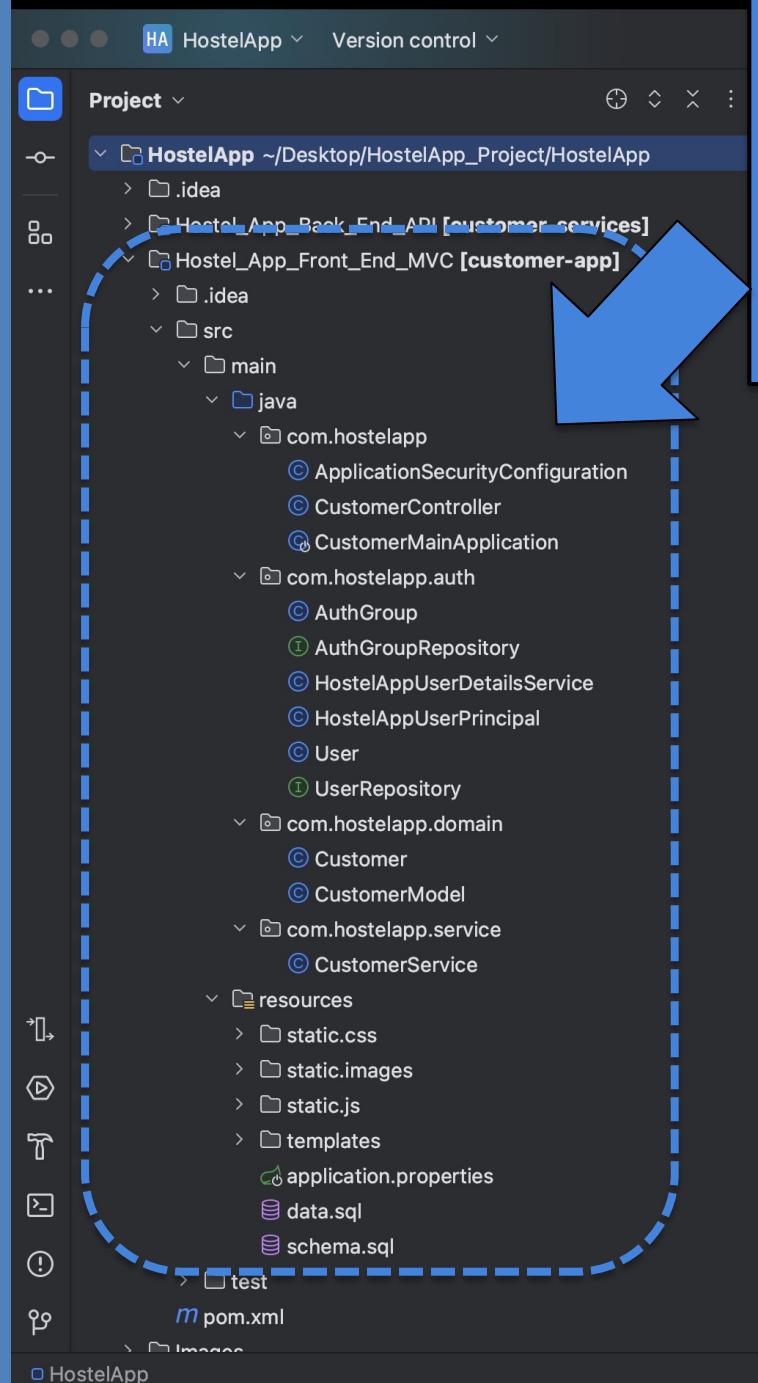
```
3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
```

```
INSERT INTO GUEST (LAST_NAME, FIRST_NAME, EMAIL_ADDRESS, COUNTRY, ADDRESS, STATE, F  
VALUES('Caulfield', 'Holden', 'holden@mit.edu', 'United States', '3998 Davis Lane', 'Massachusetts', 'MA')  
INSERT INTO GUEST (LAST_NAME, FIRST_NAME, EMAIL_ADDRESS, COUNTRY, ADDRESS, STATE, F  
VALUES('Humbert', 'Humbert', 'humbert@gmail.com', 'United States', '499 McKinley Avenue', 'Illinoi  
INSERT INTO GUEST (LAST_NAME, FIRST_NAME, EMAIL_ADDRESS, COUNTRY, ADDRESS, STATE, F  
VALUES('Bloom', 'Leopold', 'bloom@blogs.com', 'United States', '4239 Marigold Lane', 'New Jersey', 'NJ')  
INSERT INTO GUEST (LAST_NAME, FIRST_NAME, EMAIL_ADDRESS, COUNTRY, ADDRESS, STATE, F  
VALUES('Angstrom', 'Rabbit', 'angstrom@hotmail.com', 'United States', '4306 Jacobs Street', 'Wisconsin', 'WI')  
INSERT INTO GUEST (LAST_NAME, FIRST_NAME, EMAIL_ADDRESS, COUNTRY, ADDRESS, STATE, F  
VALUES('Holmes', 'Sherlock', 'holmes@aol.com', 'United States', '1395 Dola Mine Road', 'Michigan', 'MI')  
INSERT INTO GUEST (LAST_NAME, FIRST_NAME, EMAIL_ADDRESS, COUNTRY, ADDRESS, STATE, F  
VALUES('Finch', 'Atticus', 'finch@hotmail.com', 'United States', '3566 Parkway Drive', 'Alabama', 'AL')
```

HostelApp > Hostel_App_Back_End_API > src > main > resources > data.sql

1:1 LF UTF-8 4 spaces

The frontend module of
HostelApp



Directory structure and files for the frontend module. The structure of the project is based on by Apache Maven.

This repository contains the source-code and several other artifacts used in the course Application Lifecycle Management - ALM - at the undergraduate Computer Science course at UNIFAL - Alfenas - MG - Brazil.

A screenshot of a web browser displaying a frontend application for a hostel. The URL is localhost:8080. The page title is "Sparkling Water Hostel".

Sparkling Water hostel
The *hostel Sparkling Water* is a fictitious hostel located in Caxambu-MG, Brazil. Caxambu is a small town (its population in 2020 was estimated at 22,000), renowned for its spa which has twelve sources of sparkling mineral water flowing 24 hours a day and a cold-water geyser.
Caxambu is the only place in the world with 12 mineral water springs, each with different chemical properties.

Park
The Water Park is the main attraction in town.

Photo of the bathhouse located at the Water Park

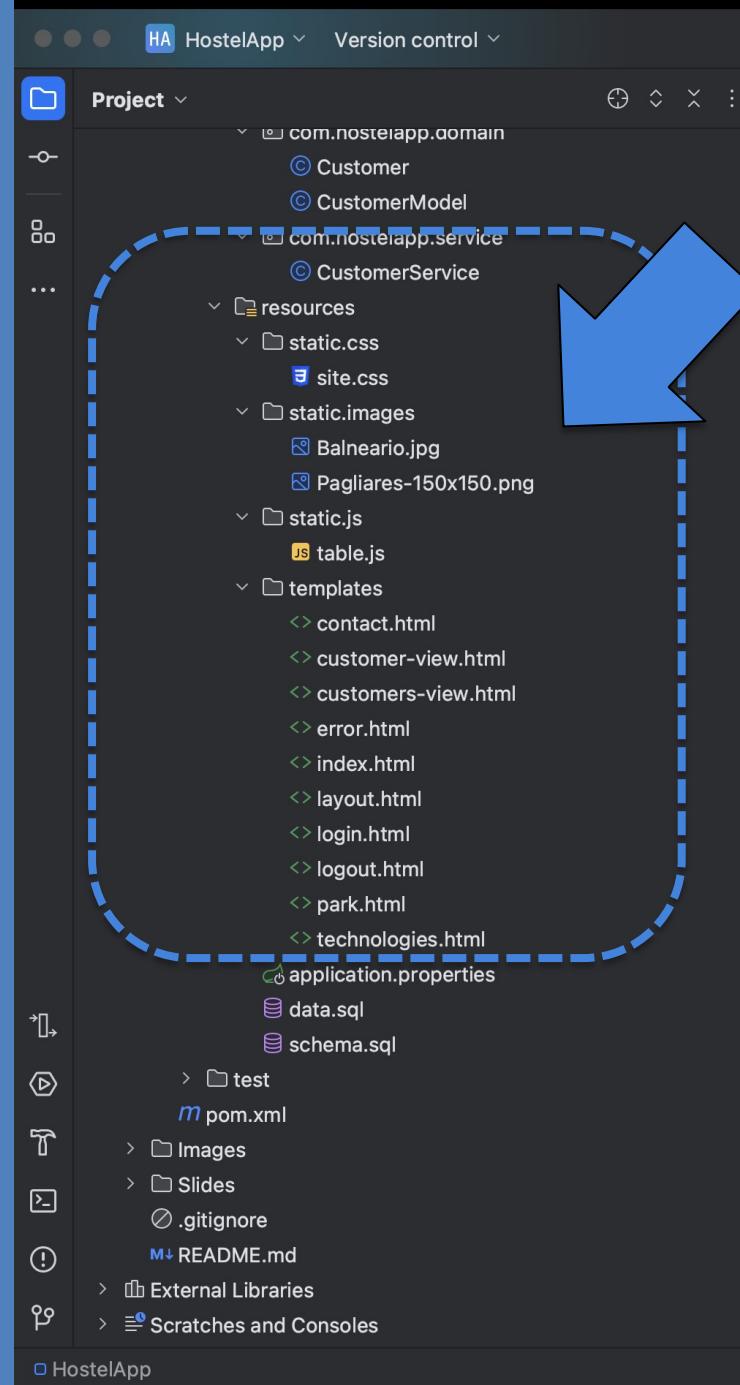
How to get to the hostel?

From Rio de Janeiro
Via BR-116 (Presidente Dutra) west past Resende-RJ to the village of Engenheiro Passos-RJ and take BR-354 north.

From Belo Horizonte or São Paulo
Via BR-381 to Campanha-MG, then west on BR-267.

Famous tourism quotes

"Never go on trips with anyone you do not love." *Ernest Hemingway*
"To travel is to discover that everyone is wrong about other countries." *Aldous Huxley*



Details of the `resources` directory, showing its contents, including Thymeleaf views (`.html` files).

This repository contains the source-code and several other artifacts used in the course Application Lifecycle Management - ALM - at the undergraduate Computer Science course at UNIFAL - Alfenas - MG - Brazil.

A screenshot of a web browser window showing the 'Customers' page of the 'Sparkling Water Hostel' website. The URL is 'localhost:8080'.

The page content includes:

- Sparkling Water hostel**: A section describing the fictitious hostel located in Caxambu-MG, Brazil, known for its spa with twelve sources of sparkling mineral water.
- How to get to the hostel?**: Instructions for travel from Rio de Janeiro and Belo Horizonte or São Paulo.
- Park**: Information about the Water Park, which is the main attraction in town, with a photo of the bathhouse.
- Famous tourism quotes**: Two quotes by Ernest Hemingway and Aldous Huxley.

HA HostelApp Version control CustomerServicesApplication

Project

com.nosteiaapp.domain
Customer
CustomerModel
com.hostelapp.service
CustomerService
resources
static.css
site.css
static.images
Balneario.jpg
Pagliares-150x150.jpg
static.js
table.js
templates
contact.html
customer-view.html
customers-view.html
error.html
index.html
layout.html
login.html
logout.html
park.html
technologies.html

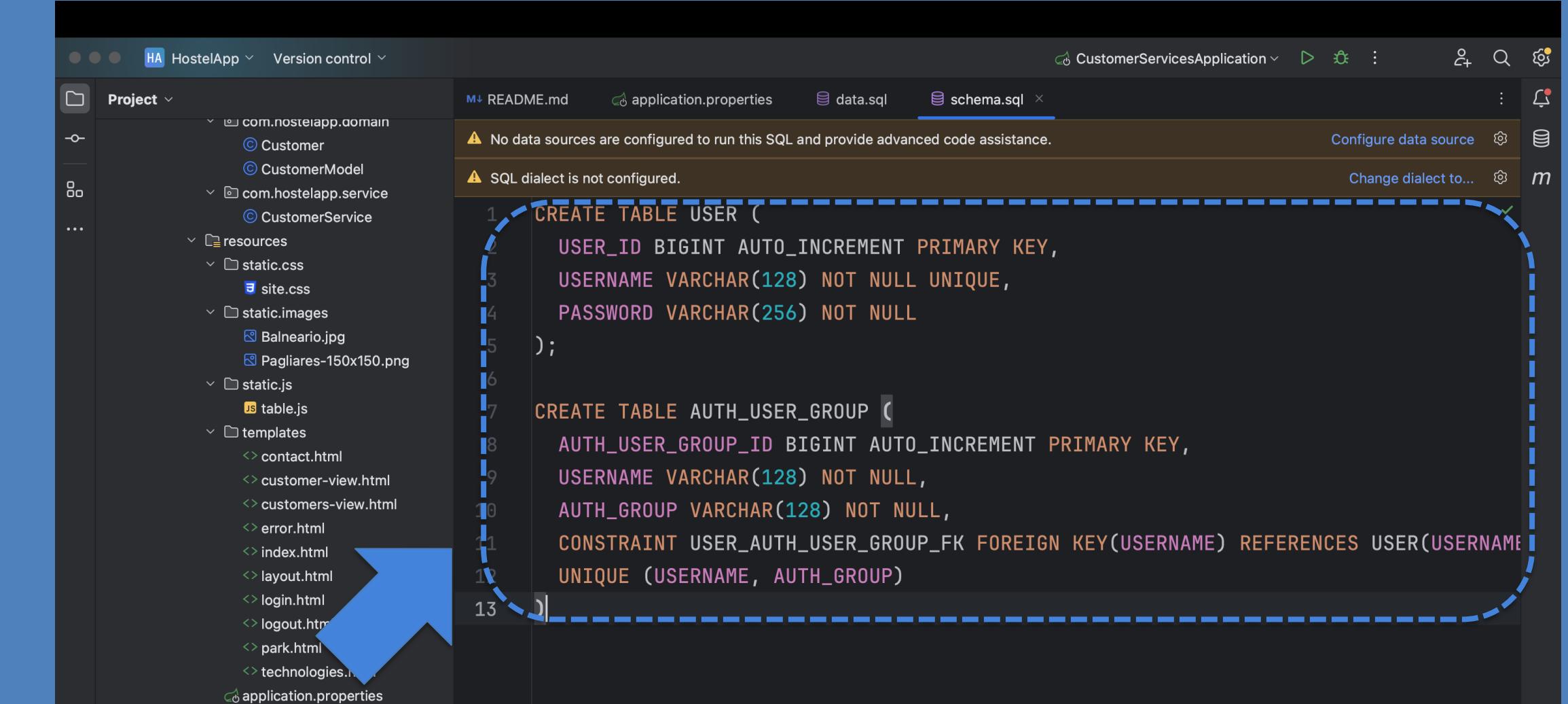
application.properties

hostelApp.customer.service.url=http://localhost:8100
spring.jpa.hibernate.ddl-auto=None

README.md

The front-end module listens on port 8080 and points to the backend module on port 8100.

HostelApp > Hostel_App_Front_End_MVC > src > main > resources > application.properties



The screenshot shows a Java IDE interface with the following details:

- Project:** HostelApp
- Version control:** Version control
- CustomerServicesApplication** tab is active.
- schema.sql:** The current file being edited.
- Content:** SQL code for creating two tables: `USER` and `AUTH_USER_GROUP`.

```
CREATE TABLE USER (
    USER_ID BIGINT AUTO_INCREMENT PRIMARY KEY,
    USERNAME VARCHAR(128) NOT NULL UNIQUE,
    PASSWORD VARCHAR(256) NOT NULL
);

CREATE TABLE AUTH_USER_GROUP (
    AUTH_USER_GROUP_ID BIGINT AUTO_INCREMENT PRIMARY KEY,
    USERNAME VARCHAR(128) NOT NULL,
    AUTH_GROUP VARCHAR(128) NOT NULL,
    CONSTRAINT USER_AUTH_USER_GROUP_FK FOREIGN KEY(USERNAME) REFERENCES USER(USERNAME)
    UNIQUE (USERNAME, AUTH_GROUP)
```
- Warnings:**
 - No data sources are configured to run this SQL and provide advanced code assistance.
 - SQL dialect is not configured.
- Buttons:** Configure data source, Change dialect to..., and a refresh button.

The **authentication** is done on the **frontend module**. These are the **database tables created when we run the application**. HostelApp uses H2 embedded database.

The screenshot shows a code editor interface with a dark theme. On the left, the project structure for 'HostelApp' is visible, including packages like com.hostelapp.domain and com.hostelapp.service, along with resources such as static.css, static.images, and templates. A large blue arrow points from the bottom left towards the data.sql file in the center.

The central area contains several tabs: README.md, application.properties, data.sql (which is currently selected), and schema.sql. The data.sql tab displays the following SQL code:

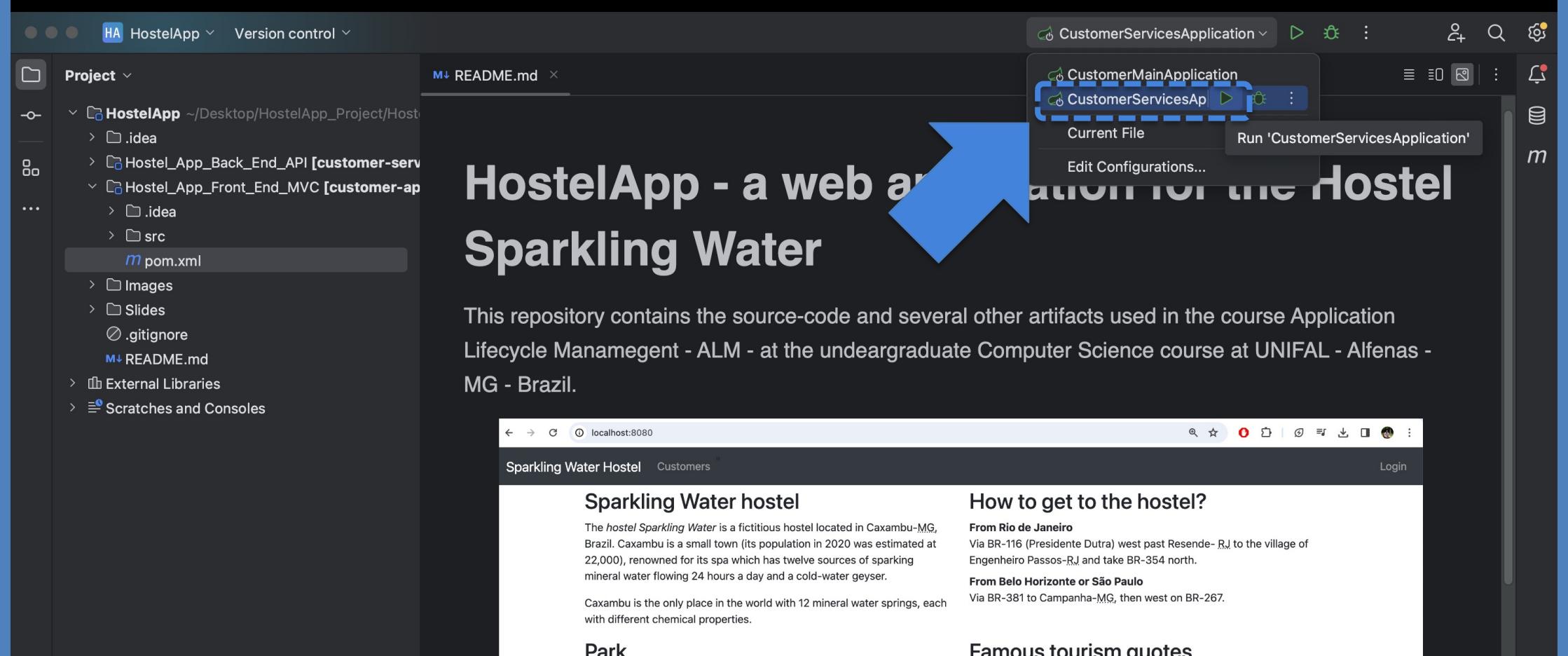
```
1 INSERT INTO USER (USERNAME, PASSWORD) VALUES ('florentino', '$2a$11$dp4wMyugYx5');
2 INSERT INTO USER (USERNAME, PASSWORD) VALUES ('fermina', '$2a$11$3N0320V1TGjap3xMpA');
3 INSERT INTO AUTH_USER_GROUP (USERNAME, AUTH_GROUP) VALUES('florentino', 'USER');
4 INSERT INTO AUTH_USER_GROUP (USERNAME, AUTH_GROUP) VALUES('florentino', 'ADMIN');
5 INSERT INTO AUTH_USER_GROUP (USERNAME, AUTH_GROUP) VALUES('fermina', 'USER');
```

Two warning messages are shown at the top of the data.sql tab:

- No data sources are configured to run this SQL and provide advanced code assistance. (Configure data source)
- SQL dialect is not configured. (Change dialect to...)

The **authentication** is done on the **frontend module**. This is the data inserted into the **database tables** **USER** and **AUTH_USER_GROUP** when we run the application.

**Running the backend module
of HostelApp**



We need to run the backend module (CustomerServicesApp) **BEFORE** the frontend module, since the frontend makes HTTP calls to the backend.

HA HostelApp Version control

CustomerServicesApplication

Project

HostelApp ~/Desktop/HostelApp_Project/HostelApp

- .idea
- Hostel_App_Back_End_API [customer-service]
- Hostel_App_Front_End_MVC [customer-application]
 - .idea
 - src
- pom.xml
- Images
- Slides
- .gitignore
- README.md

External Libraries

Scratches and Consoles

Run CustomerServicesApplication

Console Actuator

HostelApp > README.md

HostelApp - a web application for the Hostel Sparkling Water

This repository contains the source-code and several other artifacts used in the course Application Lifecycle Management - ALM - at the undergraduate Computer Science course at UNIFAL - Alfenas - MG - Brazil

Starting the backend module.

Brazil. Caxambu is a small town (its population in 2020 was estimated at 22,000), renowned for its spa which has twelve sources of sparkling mineral water flowing 24 hours a day and a cold-water geyser. Caxambu is the only place in the world with 12 mineral water springs, each with different chemical properties.

Via BR-116 (Residente Dutra) west past Resende-RJ to the village of Engenheiro Passos-RJ and take BR-354 north.

From Belo Horizonte or São Paulo
Via BR-381 to Campanha-MG, then west on BR-267.

1:1 LF UTF-8 4 spaces

HA HostelApp Version control CustomerServicesApplication

Project HostelApp ~/Desktop/HostelApp_Project/HostelApp .idea Hostel_App_Back_End_API [customer-serv Hostel_App_Front_End_MVC [customer-ap .idea src pom.xml

HostelApp - a web application for the Hostel Sparkling Water

We can see in the Run View Console that web container (Tomcat) started on port 8100, as defined in the application.properties).

Run CustomerServicesApplication

Console Actuator

```
o.s.j.e.a.AnnotationMBeanExporter
o.s.b.w.embedded.tomcat.TomcatWebServer
c.h.s.CustomerServicesApplication
o.a.c.c.C.[Tomcat].[localhost].[/]
o.s.web.servlet.DispatcherServlet
o.s.web.servlet.DispatcherServlet

: Located MBean 'dataSource': registering with JMX server as MBean [com.zaxxer.hikari.HikariDataSource]
: Tomcat started on port(s): 8100 (http) with context path ''
: Started CustomerServicesApplication in 4.617 seconds (JVM running for 16s)
: Initializing Spring FrameworkServlet 'dispatcherServlet'
: FrameworkServlet 'dispatcherServlet': initialization started
: FrameworkServlet 'dispatcherServlet': initialization completed in 42 ms
```

The screenshot shows a code editor interface with the following details:

- Project Bar:** HA HostelApp, Version control.
- File Explorer:** Shows the **HostelApp** project structure:
 - .idea
 - Hostel_App_Back_End_API [customer-service]
 - Hostel_App_Front_End_MVC [customer-app]
 - .idea
 - src
 - pom.xml
- README.md:** The file is open and contains the following text:

HostelApp - a web application for the Hostel Sparkling Water

We can also see in the Run View Console the REST endpoints the backend responds.

The screenshot shows the IntelliJ IDEA Run View Console with the following details:

- Run Tab:** CustomerServicesApplication
- Console Tab:** Actuator
- Output:** A list of RequestMappingHandlerMapping entries, each showing a mapped URL pattern and its corresponding Java method mapping. A large blue arrow points from the text above to this list.

```
s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "{'/customers'}" onto public java.util.List<com.hostelapp.service.Customer> com.hostelapp.controller.CustomerController.getCustomers()
s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "{'/customers'}" onto public org.springframework.http.ResponseEntity<com.hostelapp.service.Customer> com.hostelapp.controller.CustomerController.getCustomerById(String id)
s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "{'/customers/{id}'}" onto public com.hostelapp.service.Customer com.hostelapp.controller.CustomerController.getCustomerById(String id)
s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "{'/customers/{id}'}" onto public com.hostelapp.service.Customer com.hostelapp.controller.CustomerController.updateCustomer(com.hostelapp.service.Customer customer)
s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "{'/customers/{id}'}" onto public void com.hostelapp.controller.CustomerController.deleteCustomer(String id)
s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "{'/error'}" onto public org.springframework.http.ResponseEntity<org.springframework.http.HttpStatus> com.hostelapp.controller.ErrorController.error()
```

A screenshot of a code editor interface. The top bar shows tabs for 'HA HostelApp' and 'Version control'. The main area shows a 'Project' view with a tree structure for a 'HostelApp' project. The 'pom.xml' file is selected in the tree. The right pane displays the contents of a 'README.md' file:

HostelApp - a web application for the Hostel Sparkling Water

This repository contains the source-code and several other artifacts used in the course Application Lifecycle Managemen - ALM - at the undeargraduate Computer Science course at UNIFAL - Alfenas -

Endpoints we can use to monitor the backend status.

The bottom part of the screenshot shows the 'Actuator' tab in the code editor. A large blue arrow points from the text above down to the 'Actuator' tab. The tab is active, and the console output shows several Spring Boot Actuator endpoints mapped:

```
o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/**/favicon.ico] onto handler of type [class org.springframework.web.servlet.resource.ResourceHttpRequestHandler]
s.b.a.e.w.s.WebMvcEndpointHandlerMapping : Mapped "{[actuator/health]}",methods=[GET],produces=[application/vnd.spring-boot.actuator.v1+json, application/json]
s.b.a.e.w.s.WebMvcEndpointHandlerMapping : Mapped "{[actuator/info]}",methods=[GET],produces=[application/vnd.spring-boot.actuator.v1+json, application/json]
s.b.a.e.w.s.WebMvcEndpointHandlerMapping : Mapped "{[actuator]},methods=[GET],produces=[application/vnd.spring-boot.actuator.v1+json, application/json]
o.s.j.e.a.AnnotationMBeanExporter : Registering beans for JMX exposure on startup
o.s.j.e.a.AnnotationMBeanExporter : Bean with name 'dataSource' has been autodetected for JMX exposure
o.s.j.e.a.AnnotationMBeanExporter : Located MBean 'dataSource': registering with JMX server as MBean [com.zaxxer.hikari:name=dataSource]
```

The status bar at the bottom indicates the file is 'README.md' and the encoding is 'UTF-8'.

Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

Sat Jan 06 11:30:24 BRT 2024

There was an unexpected error (type=Not Found, status=404).

No message available



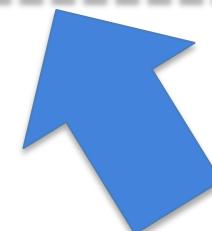
Since we decided to not include the views in the backend module, this error screens appears if you try to access the root context of Tomcat. This is what we expect in this example.

<http://localhost:8100>

```
[{"id":1,"firstName":"Jay","lastName":"Gatsby","emailAddress":"jay@gmail.com","address":"1187 Fleming Street","country":"United States","state":"AL","phoneNumber":"+1-205-555-0178"}, {"id":2,"firstName":"Holden","lastName":"Caulfield","emailAddress":"holden@mit.edu","address":"3998 Davis Lane","country":"United States","state":"CO","phoneNumber":"+1-303-555-0137"}, {"id":3,"firstName":"Humbert","lastName":"Humbert","emailAddress":"humbert@gmail.com","address":"499 McKinley Avenue","country":"United States","state":"CO","phoneNumber":"+1-303-555-0156"}, {"id":4,"firstName":"Leopold","lastName":"Bloom","emailAddress":"bloom@blogs.com","address":"4239 Marigold Lane","country":"United States","state":"FL","phoneNumber":"+1-561-555-0145"}, {"id":5,"firstName":"Rabbit","lastName":"Angstrom","emailAddress":"angstrom@hotmail.com","address":"4306 Jacobs Street","country":"United States","state":"FL","phoneNumber":"+1-561-555-0135"}, {"id":6,"firstName":"Sherlock","lastName":"Holmes","emailAddress":"holmes@aol.com","address":"1395 Dola Mine Road","country":"United States","state":"AK","phoneNumber":"+1-907-555-0187"}, {"id":7,"firstName":"Atticus","lastName":"Finch","emailAddress":"finch@hotmail.com","address":"3566 Parkway Drive","country":"United States","state":"AZ","phoneNumber":"+1-480-555-0198"}, {"id":8,"firstName":"Molly","lastName":"Bloom","emailAddress":"molly@microsoft.com","address":"4206 Mulberry Avenue","country":"United States","state":"AR","phoneNumber":"+1-501-555-0120"}, {"id":9,"firstName":"Stephen","lastName":"Dedalus","emailAddress":"dedalus@apple.com","address":"359 Hide A Way Road","country":"United States","state":"CA","phoneNumber":"+1-510-555-0183"}, {"id":10,"firstName":"Lily","lastName":"Bart","emailAddress":"bart@gmail.com","address":"639 Airplane Avenue","country":"United States","state":"CT","phoneNumber":"+1-860-555-0154"}, {"id":11,"firstName":"Holly","lastName":"Golightly","emailAddress":"golightly@gmail.com","address":"3786 Scenic Way","country":"United States","state":"IL","phoneNumber":"+1-847-555-0127"}, {"id":12,"firstName":"Gregor","lastName":"Samsa","emailAddress":"samsa@yahoo.com","address":"1833 Don Jackson Lane","country":"United States","state":"HI","phoneNumber":"+1-808-555-0162"}, {"id":13,"firstName":"Aureliano","lastName":"Buendia","emailAddress":"buendia@yahoo.com","address":"2195 Eagle Street","country":"United
```

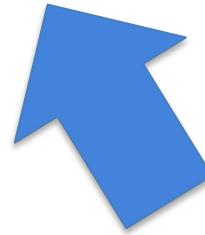
If you access the **customers** path, a list of all customers (guests) stored in the database of the backend module appears.

```
[{"id":20,"firstName":"Scarlett","lastName":"Onara","emailAddress":"onara@gmail.com","address":"3502 Station Street","country":"United States","state":"CA","phoneNumber":"+1-510-555-0187"}, {"id":21,"firstName":"Scout","lastName":"Finch","emailAddress":"finch@gmail.com","address":"2552 Cedar Street","country":"United States","state":"AR","phoneNumber":"+1-501-555-0132"}, {"id":22,"firstName":"Philip","lastName":"Marlowe","emailAddress":"marlowe@hotmail.com","address":"2810 Polk Street","country":"United States","state":"AZ","phoneNumber":"+1-480-555-0147"}, {"id":23,"firstName":"Cosimo","lastName":"di Rondo","emailAddress":"dirondo@gmail.com","address":"4772 Pinewood Drive","country":"United States","state":"AK","phoneNumber":"+1-907-555-0178"}]
```



<http://localhost:8100/customers>

```
{"id":5,"firstName":"Rabbit","lastName":"Angstrom","emailAddress":"angstrom@hotmail.com","address":"4306 Jacobs Street","country":"United States","state":"FL","phoneNumber":"+1-561-555-0135"}
```



Getting data from the customer number 5.

<http://localhost:8100/customers/5>

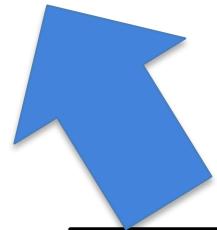
Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback

Fri Jan 05 12:01:09 BRT 2024

There was an unexpected error (type=Not Found, status=404).

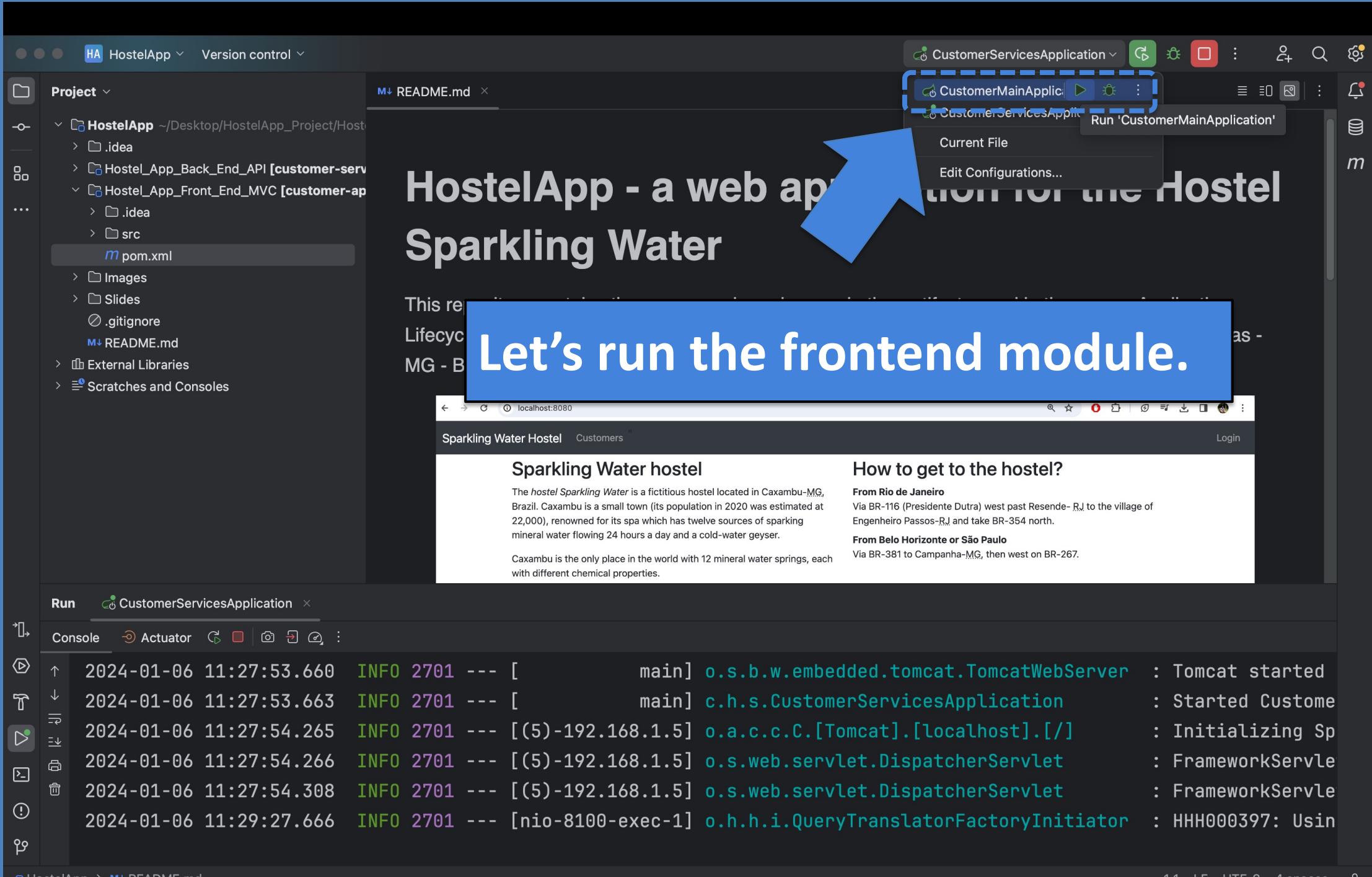
Customer not found with id: 50



Inexistent customer number 50.

<http://localhost:8100/customers/50>

Running the frontend module
of HostelApp



HA HostelApp Version control CustomerMainApplication

Project HostelApp ~/Desktop/HostelApp_Project/HostelApp .idea Hostel_App_Back_End_API [customer-service] Hostel_App_Front_End_MVC [customer-app] .idea src target pom.xml Images Slides .gitignore README.md External Libraries Scratches and Consoles

M README.md

HostelApp - a web application for the Hostel

Starting. Notice that each running module has a dedicated tab in this Run view within IntelliJ.

Sparkling Water hostel

The hostel Sparkling Water is a fictitious hostel located in Caxambu-MG, Brazil. Caxambu is a small town (its population in 2020 was estimated at 22,000), renowned for its spa which has twelve sources of sparkling mineral water flowing 24 hours a day and a cold-water geyser. Caxambu is the only place in the world with 12 mineral water springs, each with different chemical properties.

How to get to the hostel?

From Rio de Janeiro
Via BR-116 (Presidente Dutra) west past Resende-RJ to the village of Engenheiro Passos-RJ and take BR-354 north.

From Belo Horizonte or São Paulo
Via BR-381 to Campanha-MG, then west on BR-267.

Run CustomerServicesApplication CustomerMainApplication

Console Actuator

Spring Boot (v2.0.0.RELEASE)

HostelApp > README.md

1:1 LF UTF-8 4 spaces

HA HostelApp Version control CustomerMainApplication

Project HostelApp ~/Desktop/HostelApp_Project/HostelApp .idea Hostel_App_Back_End_API [customer-service] Hostel_App_Front_End_MVC [customer-application] .idea src target pom.xml Images Slides .gitignore README.md External Libraries Scratches and Consoles

M README.md

HostelApp - a web application for the Hostel Sparkling Water

This repository contains the source-code and several other artifacts used in the course Application

Lifecycle MG - E Spa

Tomcat is up and running on port 8080.

Sparkling Water hostel

The *hostel Sparkling Water* is a fictitious hostel located in Caxambu-MG, Brazil. Caxambu is a small town (its population in 2020 was estimated at 22,000), renowned for its spa which has twelve sources of sparkling mineral water flowing 24 hours a day and a cold-water geyser. Caxambu is the only place in the world with 12 mineral water springs, each with different chemical properties.

How to get to the hostel?

From Rio de Janeiro
Via BR-116 (Presidente Dutra) west past Resende-RJ to the village of Engenheiro Passos-RJ and take BR-354 north.

From Belo Horizonte or São Paulo
Via BR-381 to Campanha-MG, then

Run CustomerServicesApplication CustomerMainApplication

Console Actuator

```
o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
o.apache.catalina.core.StandardService : Starting service [Tomcat]
o.apache.catalina.core.StandardEngine : Starting Servlet Engine: Apache Tomcat/8.5.28
o.a.catalina.core.AprLifecycleListener : The APR based Apache Tomcat Native library which allows optimal performance
o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
o.s.web.context.ContextLoader : Root WebApplicationContext: initialization completed in 1983 ms
com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
```

HostelApp > README.md 1:1 LF UTF-8 4 spaces

HA HostelApp Version control CustomerMainApplication

Project HostelApp ~/Desktop/HostelApp_Project/HostelApp .idea Hostel_App_Back_End_API [customer-service] Hostel_App_Front_End_MVC [customer-app] .idea src target pom.xml Images Slides .gitignore README.md External Libraries Scratches and Consoles

HostelApp - a web application for the Hostel

The SQL scripts shown early are executed, the database tables created and populated.

Sparkling Water hostel

The *hostel Sparkling Water* is a fictitious hostel located in Caxambu-MG, Brazil. Caxambu is a small town (its population in 2020 was estimated at 22,000), renowned for its spa which has twelve sources of sparkling mineral water flowing 24 hours a day and a cold-water geyser. Caxambu is the only place in the world with 12 mineral water springs, each with different chemical properties.

How to get to the hostel?

From Rio de Janeiro
Via BR-116 (Presidente Dutra) west past Resende-RJ to the village of Engenheiro Passos-RJ and take BR-354 south.

From Belo Horizonte or São Paulo
Via BR-381 to Campanha-MG, the

Run CustomerServicesApplication CustomerMainApplication

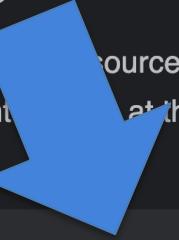
Console Actuator

```
o.s.jdbc.datasource.init.ScriptUtils
o.s.jdbc.datasource.init.ScriptUtils
o.s.jdbc.datasource.init.ScriptUtils
o.s.jdbc.datasource.init.ScriptUtils
o.s.b.w.servlet.FilterRegistrationBean
o.s.b.w.servlet.FilterRegistrationBean
o.s.b.w.servlet.FilterRegistrationBean
```

Executing SQL script from URL [file:/Users/pagliares/Desktop/HostelApp_POM/target/maven-archetype-plugin/1.4.1-SNAPSHOT/bin/resources/db/liquibase/changesets/initial/0001_initial.sql]
Executed SQL script from URL [file:/Users/pagliares/Desktop/HostelApp_POM/target/maven-archetype-plugin/1.4.1-SNAPSHOT/bin/resources/db/liquibase/changesets/initial/0002_initial_data.sql]
Executing SQL script from URL [file:/Users/pagliares/Desktop/HostelApp_POM/target/maven-archetype-plugin/1.4.1-SNAPSHOT/bin/resources/db/liquibase/changesets/initial/0003_initial_data.sql]
Executed SQL script from URL [file:/Users/pagliares/Desktop/HostelApp_POM/target/maven-archetype-plugin/1.4.1-SNAPSHOT/bin/resources/db/liquibase/changesets/initial/0004_initial_data.sql]
: Mapping filter: 'characterEncodingFilter' to: [//*]
: Mapping filter: 'hiddenHttpMethodFilter' to: [//*]
: Mapping filter: 'httpPutFormContentFilter' to: [//*]

1:1 LF UTF-8 4 spaces

We can also see in the Run View Console the REST endpoints the backend responds.



The screenshot shows the IntelliJ IDEA interface with the 'Run' view active. The left sidebar displays project files like 'target', 'pom.xml', 'Images', 'Slides', '.gitignore', 'README.md', and 'External Libraries'. The main window title is 'Sparkling water'. Below it, a message reads: 'This repository contains the source-code and several other artifacts used in the course Application Lifecycle Management taught at the undergraduate Computer Science course at UNIFAL - Alfenas - MG - Brazil.' The 'Console' tab is selected, showing a list of 'RequestMappingHandlerMapping' entries. Each entry shows a URL pattern and its corresponding method mapping. A large blue dashed box highlights the first few entries in the list. The bottom status bar indicates the file is 'HostelApp > README.md' and the encoding is 'UTF-8'.

```
s.m.m.a.RequestMappingHandlerMapping : Mapped "{{/ | /index},methods=[GET]}" onto public java.lang.String com.hostelapp.CustomerMainApplication.index()
s.m.m.a.RequestMappingHandlerMapping : Mapped "{{/login},methods=[GET]}" onto public java.lang.String com.hostelapp.CustomerMainApplication.login()
s.m.m.a.RequestMappingHandlerMapping : Mapped "{{/logout-success},methods=[GET]}" onto public java.lang.String com.hostelapp.CustomerMainApplication.logoutSuccess()
s.m.m.a.RequestMappingHandlerMapping : Mapped "{{/customers},methods=[GET]}" onto public java.lang.String com.hostelapp.CustomerMainApplication.customers()
s.m.m.a.RequestMappingHandlerMapping : Mapped "{{/customers/add},methods=[GET]}" onto public java.lang.String com.hostelapp.CustomerMainApplication.addCustomer()
s.m.m.a.RequestMappingHandlerMapping : Mapped "{{/customers},methods=[POST]}" onto public org.springframework.web.servlet.ModelAndView com.hostelapp.CustomerMainApplication.addCustomer()
s.m.m.a.RequestMappingHandlerMapping : Mapped "{{/customers/{id}},methods=[GET]}" onto public java.lang.String com.hostelapp.CustomerMainApplication.getCustomer()
s.m.m.a.RequestMappingHandlerMapping : Mapped "{{/customers/{id}},methods=[POST]}" onto public java.lang.String com.hostelapp.CustomerMainApplication.updateCustomer()
s.m.m.a.RequestMappingHandlerMapping : Mapped "{{/contact},methods=[GET]}" onto public java.lang.String com.hostelapp.CustomerMainApplication.contact()
s.m.m.a.RequestMappingHandlerMapping : Mapped "{{/technologies},methods=[GET]}" onto public java.lang.String com.hostelapp.CustomerMainApplication.technologies()
s.m.m.a.RequestMappingHandlerMapping : Mapped "{{/park},methods=[GET]}" onto public java.lang.String com.hostelapp.CustomerMainApplication.park()
s.m.m.a.RequestMappingHandlerMapping : Mapped "{{/error},produces=[text/html]}" onto public org.springframework.web.servlet.ModelAndView com.hostelapp.CustomerMainApplication.error()
s.m.m.a.RequestMappingHandlerMapping : Mapped "{{/error}}" onto public org.springframework.http.ResponseEntity<java.util.Map<java.lang.String,java.lang.Object>> com.hostelapp.CustomerMainApplication.error()
```

Endpoints we can use to monitor the frontend status.



This repository contains the source-code and several other artifacts used in the course Application Lifecycle Management - ALM - for the undergraduate Computer Science course at UNIFAL - Alfenas - MG - Brazil.

```
HostelApp > M README.md
```

pom.xml

- > Images
- > Slides
- ∅ .gitignore
- M README.md
- > External Libraries

Run CustomerServicesApplication × CustomerMainApplication ×

Console Actuator

```
b.a.w.s.WelcomePageHandlerMapping : Adding welcome page template: index
b.d.a.OptionalLiveReloadServer : LiveReload server is running on port 35729
a.e.w.s.WebMvcEndpointHandlerMapping : Mapped "[actuator/health]", methods=[GET], produces=[application/vnd.spring-boot.actuator+json]
a.e.w.s.WebMvcEndpointHandlerMapping : Mapped "[actuator/info]", methods=[GET], produces=[application/vnd.spring-boot.actuator+json]
a.e.w.s.WebMvcEndpointHandlerMapping : Mapped "[actuator]", methods=[GET], produces=[application/vnd.spring-boot.actuator+json]
j.e.a.AnnotationMBeanExporter : Registering beans for JMX exposure on startup
j.e.a.AnnotationMBeanExporter : Bean with name 'dataSource' has been autodetected for JMX exposure
j.e.a.AnnotationMBeanExporter : Located MBean 'dataSource': registering with JMX server as MBean [com.zaxxer.hikari:name=dataSource]
b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
hostelapp.CustomerMainApplication : Started CustomerMainApplication in 6.134 seconds (JVM running for 11.752)
c.c.C.[Tomcat].[localhost].[] : Initializing Spring FrameworkServlet 'dispatcherServlet'
web.servlet.DispatcherServlet : FrameworkServlet 'dispatcherServlet': initialization started
web.servlet.DispatcherServlet : FrameworkServlet 'dispatcherServlet': initialization completed in 66 ms
```

Sparkling Water hostel

The *hostel Sparkling Water* is a fictitious hostel located in Caxambu-MG, Brazil. Caxambu is a small town (its population in 2020 was estimated at 22,000), renowned for its spa which has twelve sources of sparkling mineral water flowing 24 hours a day and a cold-water geyser.

Caxambu is the only place in the world with 12 mineral water springs, each with different chemical properties.

Park

The Water Park is the main attraction in town.



Photo of the bathhouse located at the Water Park

How to get to the hostel?

From Rio de Janeiro

Via BR-116 (Presidente Dutra) west past Resende-RJ to the village of Engenheiro Passos-RJ and take BR-354 north.

From Belo Horizonte or São Paulo

Via BR-381 to Campanha-MG, then west on BR-267.

Famous tourism quotes

"Never go on trips with anyone you do not love." *Ernest Hemingway*

"To travel is to discover that everyone is wrong about other countries."
Aldous Huxley

Testing the frontend.



<http://localhost:8080>

Services view in IntelliJ

HA HostelApp Version control CustomerMainApplication

IntelliJ provides a view called Services that can aids developers to start/stop the modules. Let's configure it.

.gitignore README.md External Libraries Scratches and Consoles

Lifecycle Manamegent - ALM - at the undeargraduate Computer Science course at UNIFAL - Alfenas - MG - Brazil.

localhost:8080 Sparkling Water Hostel Customers Login

Sparkling Water hostel

The *hostel Sparkling Water* is a fictitious hostel located in Caxambu-MG, Brazil. Caxambu is a small town (its population in 2020 was estimated at 22,000), renowned for its spa which has twelve sources of sparkling mineral water flowing 24 hours a day and a cold-water geyser. Caxambu is the only place in the world with 12 mineral water springs, each with different chemical properties.

How to get to the hostel?

From Rio de Janeiro
Via BR-116 (Presidente Dutra) west past Resende- RJ to the village of Engenheiro Passos-RJ and take BR-354 north.

From Belo Horizonte or São Paulo
Via BR-381 to Campanha-MG, then west on BR-267.

Double-click on the server node to connect

Services All Services Docker Add Service

HostelApp README.md 1:1 LF UTF-8 4 spaces

HA HostelApp Version control CustomerMainApplication

Project

- HostelApp ~/Desktop/HostelApp_Project/HostelApp
- .idea
- Hostel_App_Back_End_API [customer-service]
- Hostel_App_Front_End_MVC [customer-app]
- .idea
- src
- target
- pom.xml
- Images
- Slides
- .gitignore
- README.md
- External Libraries
- Scratches and Consoles

M README.md

HostelApp - a web application for the Hostel Sparkling Water

Select Run Configuration Type.

Course Application
at UNIFAL - Alfenas -

Sparkling Water Hostel Customers

Login

Sparkling Water hostel

The *hostel Sparkling Water* is a fictitious hostel located in Caxambu-MG, Brazil. Caxambu is a small town (its population in 2020 was estimated at 22,000), renowned for its spa which has twelve sources of sparkling mineral water flowing 24 hours a day and a cold-water geyser. Caxambu is the only place in the world with 12 mineral water springs, each with different chemical properties.

How to get to the hostel?

From Rio de Janeiro
Via BR-116 (Presidente Dutra) west past Resende-RJ to the village of Engenheiro Passos-RJ and take BR-354 north.

From Belo Horizonte or São Paulo
Via BR-381 to Campanha-MG, then west on BR-267.

Services All Services

Docker Run Configuration Type Docker Connection Docker Connections From Docker Contexts Docker Registry

Double-click on the server node to connect

HostelApp > README.md

1:1 LF UTF-8 4 spaces

HA HostelApp Version control CustomerMainApplication

Project

HostelApp ~/Desktop/HostelApp_Project/HostelApp .idea Hostel_App_Back_End_API [customer-service] Hostel_App_Front_End_MVC [customer-application] .idea src target pom.xml Images Slides .gitignore README.md External Libraries Scratches and Consoles

M README.md

HostelApp - a web application for the Hostel Sparkling Water

This repository contains the source-code and several other artifacts used in the course Application Lifecycle Management - ALM - at the undergraduate Computer Science course at UNIFAL - Alfenas - MG - Brazil.

localhost:8080

Sparkling Water Hostel Customers Login

Add Configuration Type

Maven Mocha Node.js Nodeunit npm NW.js OpenAPI/Swagger Codegen Protractor React Native Remote JVM Debug Shell Script Spring Boot TestNG Vitest XSLT 8 more items... Select one or more types

Select Spring Boot.

Caxambu is the only place in the world with 12 mineral water springs, each with different chemical properties.

How to get to the hostel?

From Rio de Janeiro
Via BR-116 (Presidente Dutra) west past Resende-RJ to the village of Engenheiro Passos-RJ and take BR-354 north.

From Belo Horizonte or São Paulo
Via BR-381 to Campanha-MG, then west on BR-267.

Double-click on the server node to connect

1:1 LF UTF-8 4 spaces

HA HostelApp Version control CustomerMainApplication

Project HostelApp ~/Desktop/HostelApp_Project/HostelApp .idea Hostel_App_Back_End_API [customer-service] Hostel_App_Front_End_MVC [customer-app] .idea src target pom.xml Images Slides .gitignore README.md External Libraries Scratches and Consoles

M README.md

HostelApp - a web application for the Hostel Sparkling Water

This repository contains the source-code and several other artifacts used in the course Application Lifecycle Managemen - ALM - at the undergraduate Computer Science course at UNIFAL - Alfenas - MG - Brazil.

localhost:8080

Sparkling Water Hostel Customers Login

Sparkling Water hostel

The hostel *Sparkling Water* is a fictitious hostel located in Caxambu-MG, Brazil. Caxambu is a small town (its population in 2020 was estimated at 22,000), renowned for its spa which has twelve sources of sparkling mineral water flowing 24 hours a day and a cold-water geyser. Caxambu is the only place in the world with 12 mineral water springs, each with different chemical properties.

How to get to the hostel?

From Rio de Janeiro
Via BR-116 (Presidente Dutra) west past Resende- RJ to the village of Engenheiro Passos- RJ and take BR-354 north.

From Belo Horizonte or São Paulo
Via BR-381 to Campanha-MG, then west on BR-267.

Services All Services

Spring Boot Running CustomerMainApplication [devtools] :8080 CustomerServicesApplication :8100 Docker

Console Actuator

```
...www.ServletRegistrationBean ... Mapping filter ... characterEncodingFilter ... .w.servlet.FilterRegistrationBean : Mapping filter: 'hiddenHttpMethodFilter' ... .w.servlet.FilterRegistrationBean : Mapping filter: 'httpPutFormContentFilter' ... .w.servlet.FilterRegistrationBean : Mapping filter: 'requestContextFilter' to ... .legatingFilterProxyRegistrationBean : Mapping filter: 'springSecurityFilterChain' ... .w.servlet.FilterRegistrationBean : Mapping filter: 'httpTraceFilter' to: /* ... .w.servlet.FilterRegistrationBean : Mapping filter: 'webMvcMetricsFilter' to: ... .w.servlet.ServletRegistrationBean : Servlet dispatcherServlet mapped to [/]
```

HostelApp > README.md 1:1 LF UTF-8 4 spaces

HA HostelApp Version control CustomerMainApplication

Project HostelApp ~/Desktop/HostelApp_Project/HostelApp .idea Hostel_App_Back_End_API [customer-serv Hostel_App_Front_End_MVC [customer-ap .idea src target pom.xml Images Slides .gitignore README.md External Libraries Scratches and Consoles

HostelApp - a web application for the Hostel Sparkling Water

We can now view the modules deployed and server status in a centralized way.

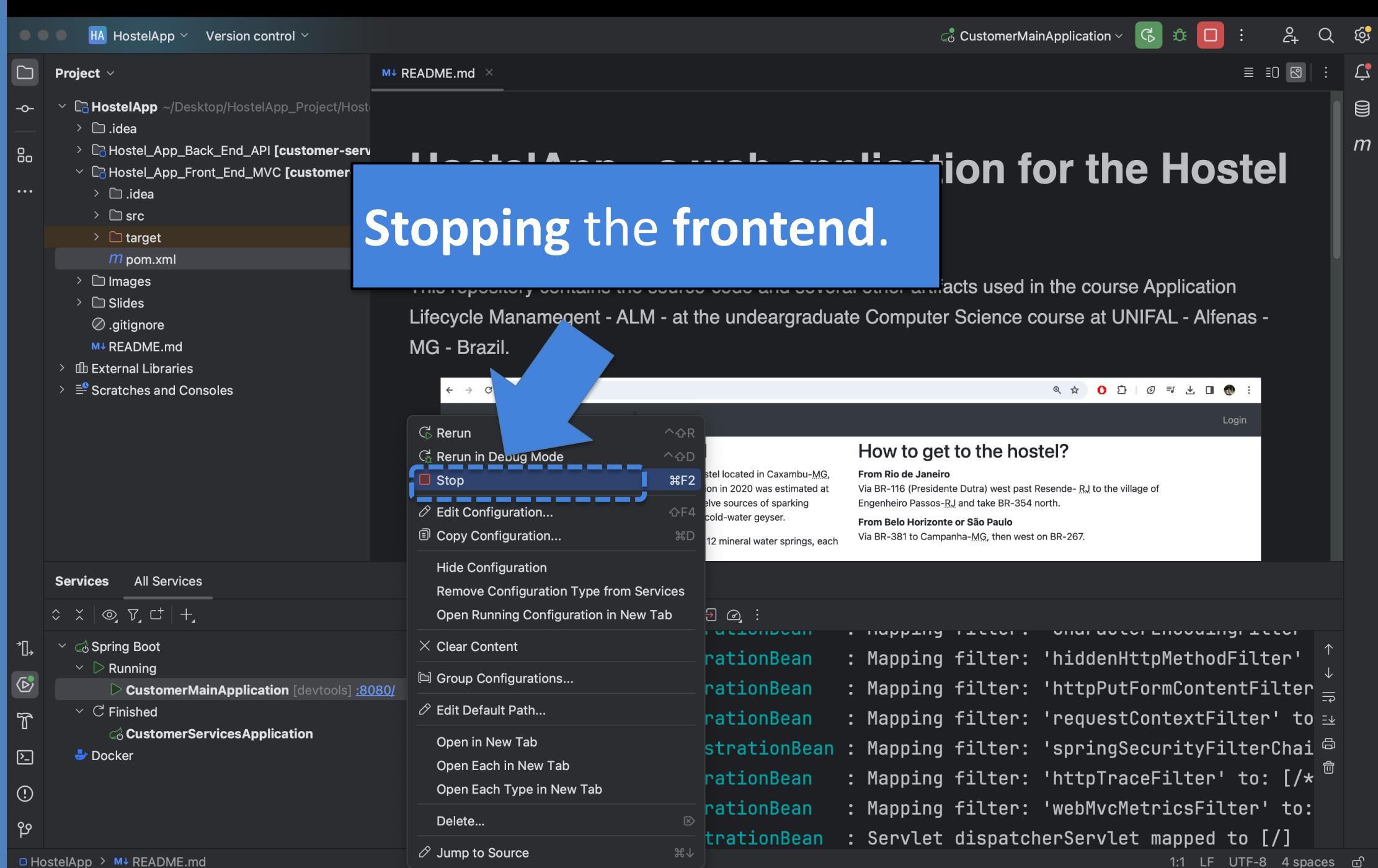
Stop ⌘F2 Edit Configuration... ⌘F4 Configuration... ⌘D Hide Configuration Remove Configuration Type from Services Open Running Configuration in New Tab

Clear Content Group Configurations... Edit Default Path... Open in New Tab Open Each in New Tab Open Each Type in New Tab Delete... Jump to Source ⌘↓

Services All Services Spring Boot Running CustomerMainApplication [devtools] :8000 CustomerServicesApplication :8100 Docker

200 INFO 2701 --- [main] s.b.a.e.w.s.WebMvcEndpo 527 INFO 2701 --- [main] s.b.a.e.w.s.WebMvcEndpo 528 INFO 2701 --- [main] s.b.a.e.w.s.WebMvcEndpo 607 INFO 2701 --- [main] o.s.j.e.a.AnnotationMB 609 INFO 2701 --- [main] o.s.j.e.a.AnnotationMB 620 INFO 2701 --- [main] o.s.j.e.a.AnnotationMB 660 INFO 2701 --- [main] o.s.b.w.embedded.tomca

HostelApp > README.md 1:1 LF UTF-8 4 spaces



HA HostelApp Version control CustomerMainApplication

Project

HostelApp ~/Desktop/HostelApp_Project/HostelApp

- .idea
- Hostel_App_Back_End_API [customer-service]
- Hostel_App_Front_End_MVC [customer-app]
- .idea
- src
- target

pom.xml

Images

Slides

.gitignore

README.md

M README.md

README.md

HostelApp - a web application for the Hostel Sparkling Water

This repository contains the source-code and several other artifacts used in the course Application Lifecycle Management - ALM - at the undergraduate Computer Science course at UNIFAL - Alfenas - MG - Brazil.

Both modules stopped (finished).

Services All Services

Spring Root

CustomerMainApplication [devtools]

CustomerServicesApplication

Docker

How to get to the hostel?

In Rio de Janeiro

BR-116 (Presidente Dutra) west past Resende-RJ to the village of Engenheiro Passos-RJ and take BR-354 north.

From Belo Horizonte or São Paulo

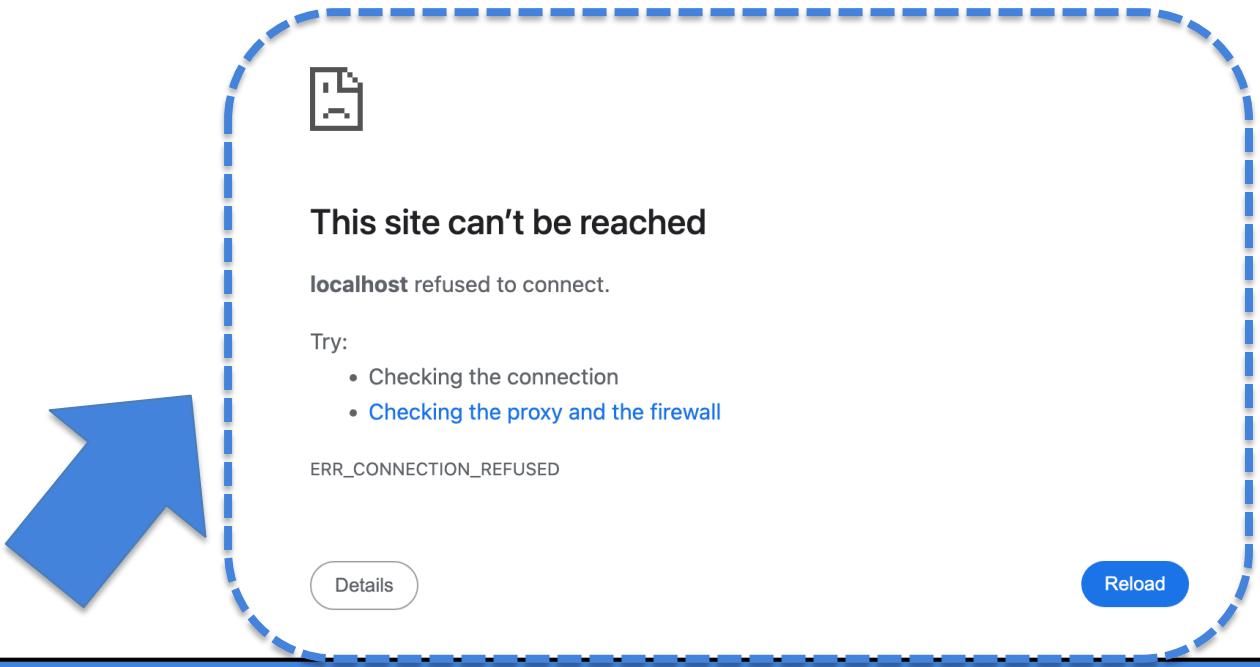
Via BR-381 to Campanha-MG, then west on BR-267.

Console Actuator

```
www.servlet.FilterRegistrationBean : Mapping filter: 'hiddenHttpMethodFilter'  
www.servlet.FilterRegistrationBean : Mapping filter: 'httpPutFormContentFilter'  
www.servlet.FilterRegistrationBean : Mapping filter: 'requestContextFilter' to  
DelegatingFilterProxyRegistrationBean : Mapping filter: 'springSecurityFilterChain'  
www.servlet.FilterRegistrationBean : Mapping filter: 'httpTraceFilter' to: /*  
www.servlet.FilterRegistrationBean : Mapping filter: 'webMvcMetricsFilter' to:  
www.servlet.ServletRegistrationBean : Servlet dispatcherServlet mapped to [/]
```

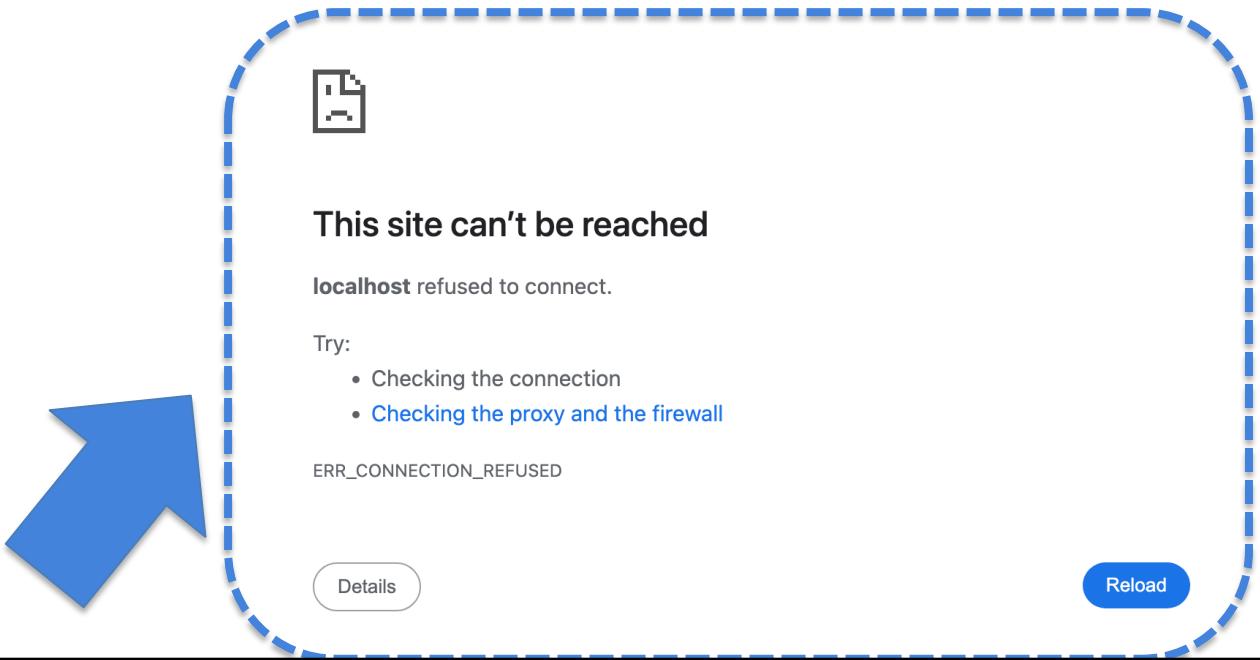
HostelApp > README.md

1:1 LF UTF-8 4 spaces



After stopping, it is not possible to access the frontend anymore.

<http://localhost:8080>



After stopping, it is not possible to access the backemd anymore.

<http://localhost:8100>

Online Demo

A screencast with demonstration of the web application is available at YouTube

YouTube BR

Pesquisar

localhost:8080/index

Sparkling Water Hostel Customers Login

Sparkling Water hostel

The hostel *Sparkling Water* is a fictitious hostel located in Caxambu-MG, Brazil. Caxambu is a small town (its population in 2020 was estimated at 22,000), renowned for its spa which has twelve sources of sparkling mineral water flowing 24 hours a day and a cold-water geyser.

Caxambu is the only place in the world with 12 mineral water springs, each with different chemical properties.

Park

The Water Park is the main attraction in town.



Photo of the bathhouse located at the Water Park

How to get to the hostel?

From Rio de Janeiro
Via BR-116 (Presidente Dutra) west past Resende- RJ to the village of Engenheiro Passos-RJ and take BR-354 north.

From Belo Horizonte or São Paulo
Via BR-381 to Campanha-MG, then west on BR-267.

Famous tourism quotes

"Never go on trips with anyone you do not love." Ernest Hemingway

"To travel is to discover that everyone is wrong about other countries." Aldous Huxley

Demonstração da aplicação web para o Hostel Sparkling Water

Não listado

1 visualização • 18 de fev. de 2022

Rodrigo Pagliares 1,79 mil inscritos

ANALYTICS EDITAR VÍDEO

https://bit.ly/3JxW6II

What's next?

Demo

Demonstration of the web application
(Hostel Sparkling Water)

I created a set of slides with step-by-step instructions on how to use the HostelApp. Check it out!

The screenshot shows the homepage of the Hostel Sparkling Water website. It features a red header with the hostel's name. Below the header, there are several sections: a "Park" section with a photo of a bathhouse and a short description; a "Famous tourism quotes" section with two quotes from Ernest Hemingway and Aldous Huxley; and a "Directions" section with driving instructions from Belo Horizonte or São Paulo. The footer contains copyright information.

Homepage just after accessing the web application for the hostel Sparkling Water

In this example, Florentino is a reservation manager and can see the list of customers AND register new customers. Fermina is a receptionist. She can see the list of customers of the hostel, but she CANNOT create new customers. Let's authenticate with the user **florentino** (with password equals to password)

Slides/Authentication_Authorization_HostelApp.pdf

Questions?