# Testing PlaceToBeer

10. JUNI

**PLACE TO BEER**

**Verfasst von: Tom**

# Greetings, stranger!

## Introduction

Congratulations, you have been chosen to be one of the very few people to experience the current state of our project "PlaceToBeer"!

This manual will describe how to gather the Data, get the project running, and what cases to test.

If something doesn't work, if you find anything suspicious or noteworthy please let us know at https://placetobeer475840703.wordpress.com/contact/. If you happen to know any team member personally you can contact us in any way you want 😄

This document is all about testing the group-service of our project and checking different functionalities. If you are not familiar with this project, here you can find the documentation where everything is described: https://github.com/placetobeer/ptb-documentation

Cheers!

# Prepare your IDE

We used the **IntelliJ IDE** for our backend and we recommend you do the same. This is not because it won't work for different IDEs but the environment variables were implemented in IntelliJ and we have no idea how this will turn out with any other IDE, so it's up to you 🤷
For the Frontend we chose **WebStorm**, we also recommend you do the same.

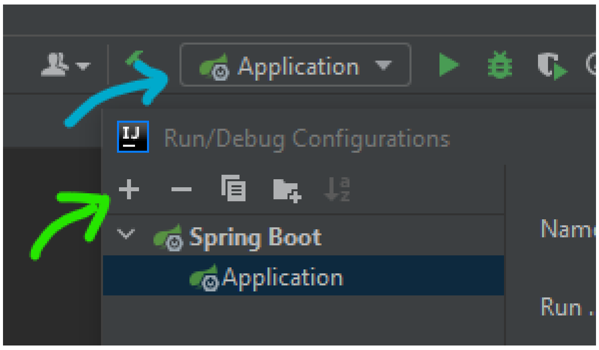
# Setup

### Where to download

You will have to download the backend and frontend separately. To do so please download or clone from the current master branch of each of them.
**Frontend**: https://github.com/placetobeer/ptb-frontend/commit/9d552d24c46b97f8035ca7c15b39b9b296b1284d
**Backend**: https://github.com/placetobeer/ptb-backend/commit/0b6dc26057923f49cca853157b8879e42e03fb94

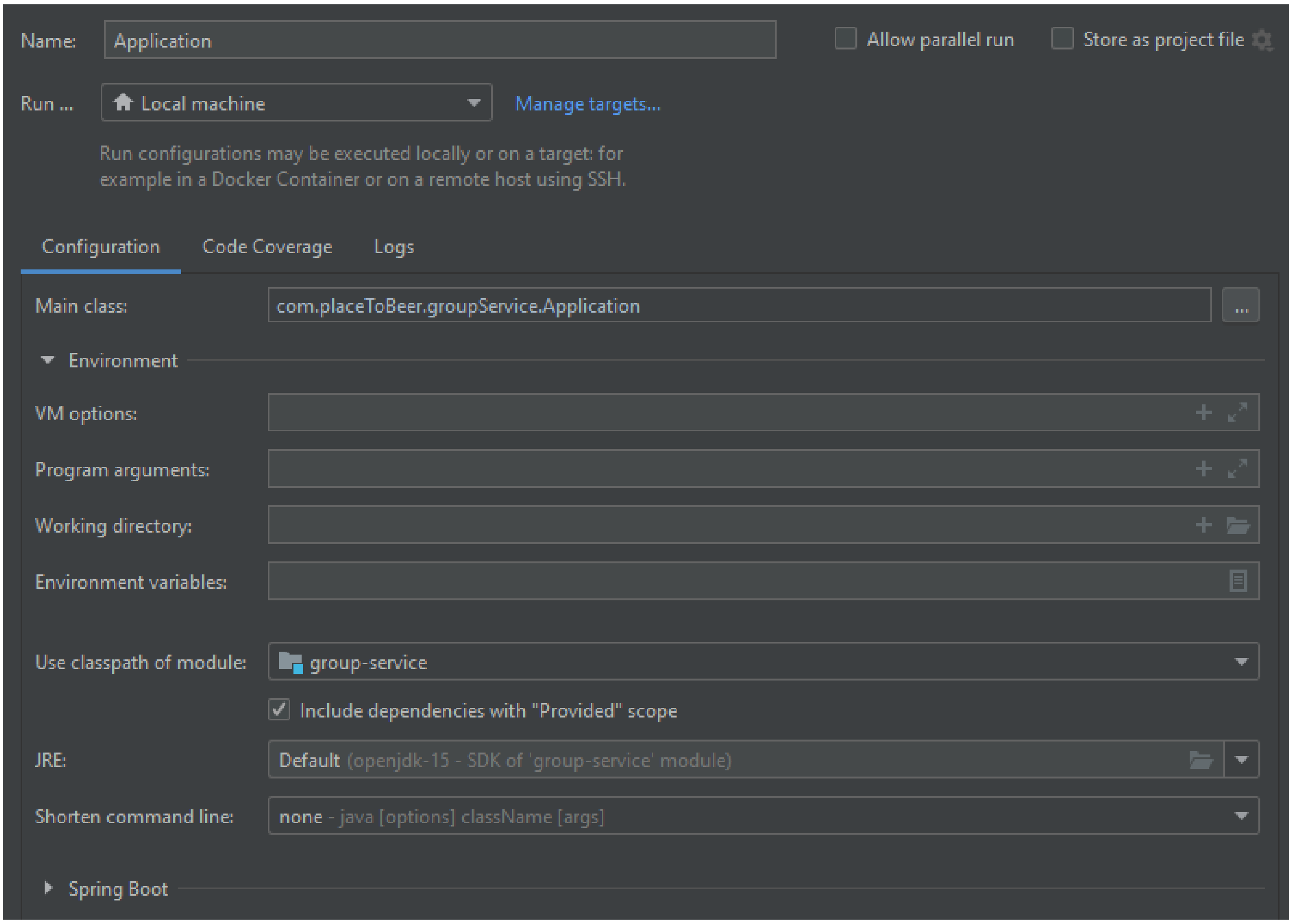
## Set Environment Variables

To finalize setting up the project in IntelliJ, you will need to add *Run Configurations*.



To do so you need to access the Run/ Debug Configurations in the top right corner. In the popup you can add a Spring Boot Application. Give it a name (e.g. “Application”) and add com.placeToBeer.groupService.Application to Main class in the Configuration tab.

By now, your Run Configuration should automatically look like this:



Name: Application

□ Allow parallel run

□ Store as project file

Run ...: Local machine

Run configurations may be executed locally or on a target: for example in a Docker Container or on a remote host using SSH.

Main class: com.placeToBeer.groupService.Application

Environment

VM options:

Program arguments:

Working directory:

Environment variables:

Use classpath of module: group-service

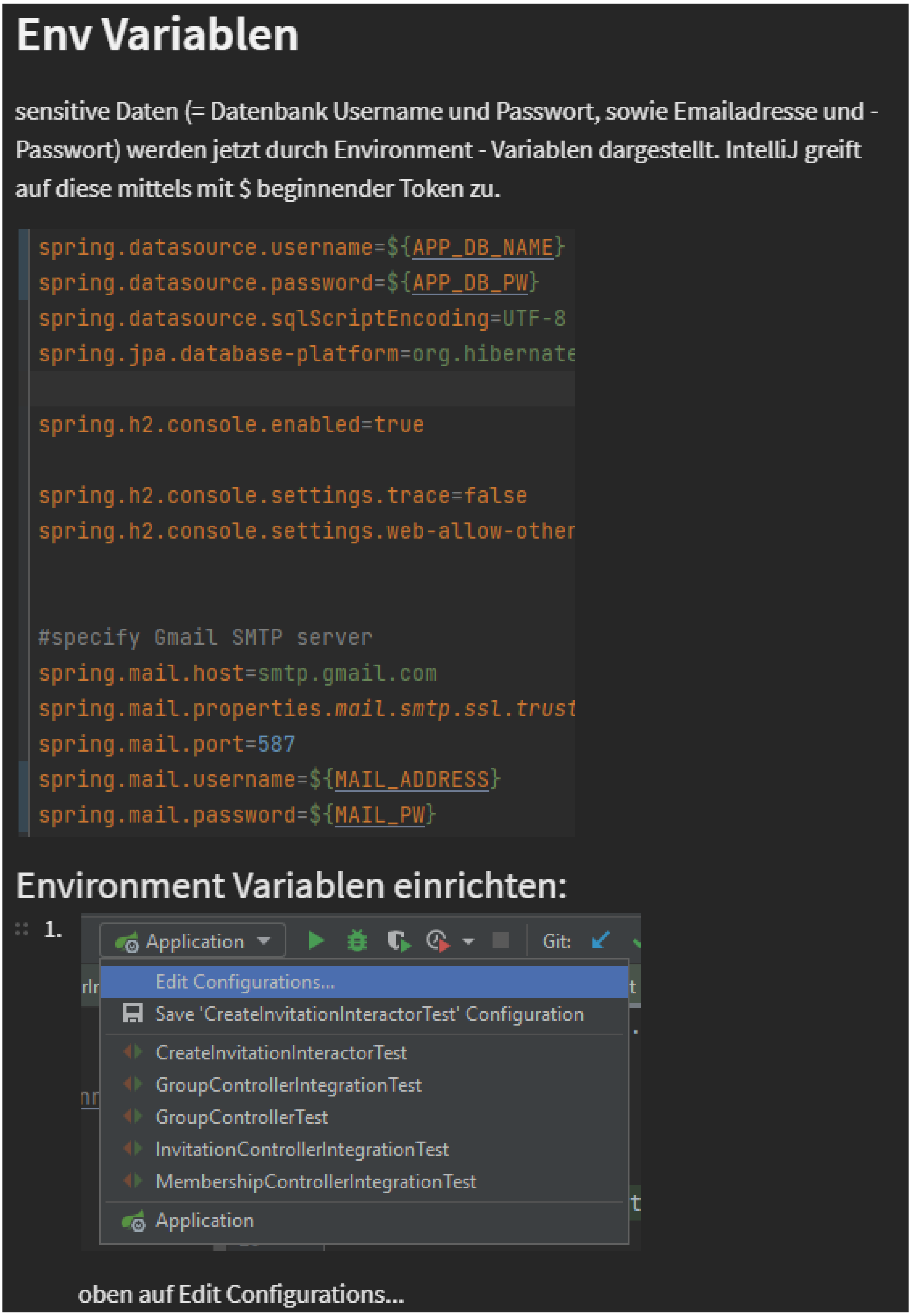
☑ Include dependencies with "Provided" scope

JRE: Default (openjdk-15 - SDK of 'group-service' module)

Shorten command line: none - java [options] className [args]

Spring Boot

To set your environment variables please follow the instructions of this short guide I created a while ago (if the German language bothers you, I recommend checking out the link at the end of it):



## Env Variablen

sensitive Daten (= Datenbank Username und Passwort, sowie Emailadresse und - Passwort) werden jetzt durch Environment - Variablen dargestellt. IntelliJ greift auf diese mittels mit $ beginnender Token zu.

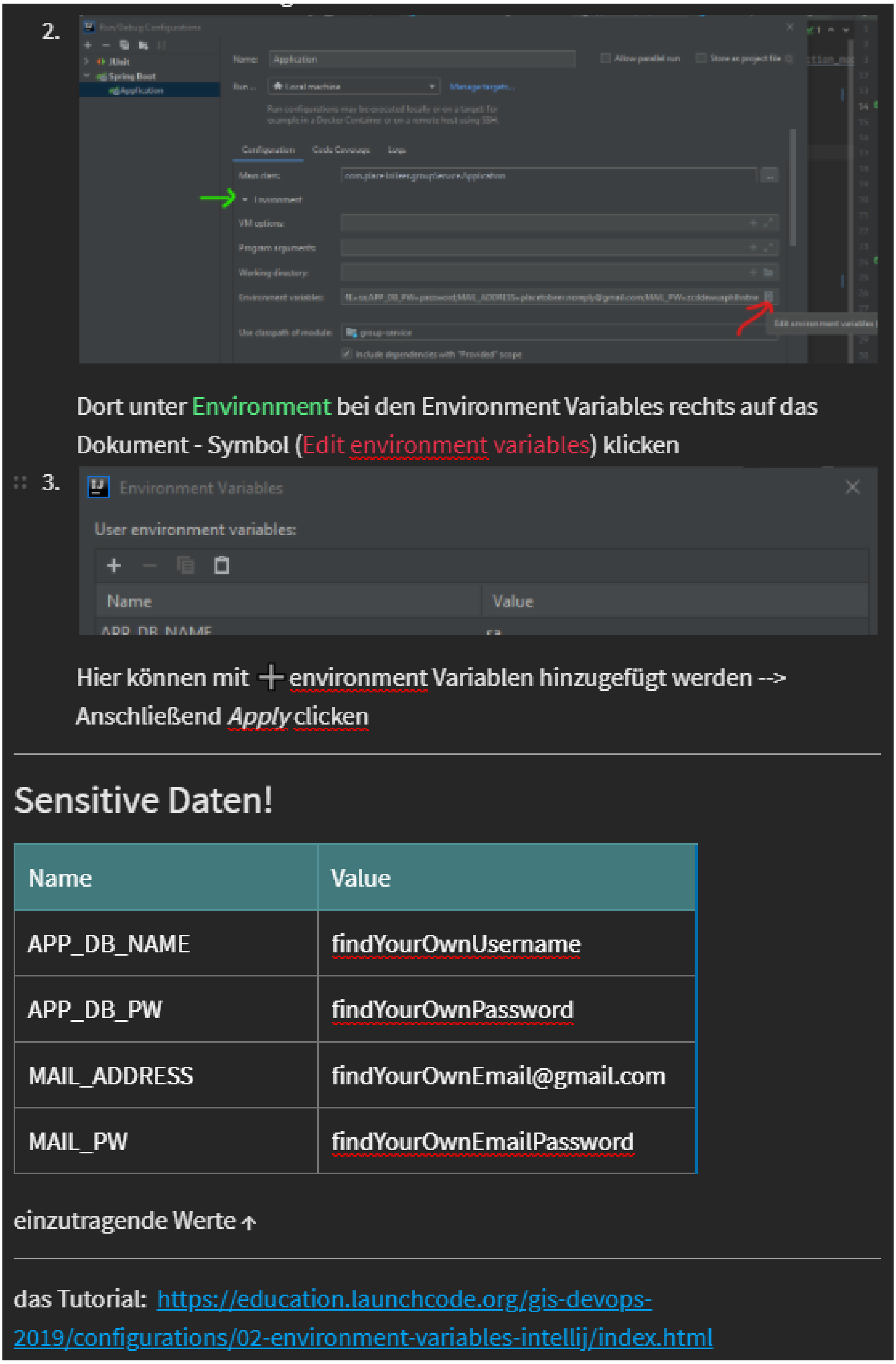
```
spring.datasource.username=${APP_DB_NAME}
spring.datasource.password=${APP_DB_PW}
spring.datasource.sqlScriptEncoding=UTF-8
spring.jpa.database-platform=org.hibernate

spring.h2.console.enabled=true

spring.h2.console.settings.trace=false
spring.h2.console.settings.web-allow-other


#specify Gmail SMTP server
spring.mail.host=smtp.gmail.com
spring.mail.properties.mail.smtp.ssl.trust
spring.mail.port=587
spring.mail.username=${MAIL_ADDRESS}
spring.mail.password=${MAIL_PW}
```

## Environment Variablen einrichten:

1.

Application
Git:
Edit Configurations...
Save 'CreateInvitationInteractorTest' Configuration
CreateInvitationInteractorTest
GroupControllerIntegrationTest
GroupControllerTest
InvitationControllerIntegrationTest
MembershipControllerIntegrationTest
Application

oben auf Edit Configurations...



2.

Run/Debug Configurations

Name: Application

Run on: Local machine

Run configurations may be executed locally or on a target: for example in a Docker Container or on a remote host using SSH.

Main class: com.placetobeer.groupservice.Application

Environment

Environment variables: ...=sa;APP_DB_PW=password;MAIL_ADDRESS=placetobeer.noreply@gmail.com;MAIL_PW=zcddewuaphlhxtne

Use classpath of module: group-service

☑ Include dependencies with "Provided" scope

Dort unter Environment bei den Environment Variables rechts auf das Dokument - Symbol (Edit environment variables) klicken

3.

Environment Variables

User environment variables:

| Name | Value |
|---|---|
| APP_DB_NAME | sa |

Hier können mit + environment Variablen hinzugefügt werden --> Anschließend *Apply* clicken

## Sensitive Daten!

| Name | Value |
|---|---|
| APP_DB_NAME | findYourOwnUsername |
| APP_DB_PW | findYourOwnPassword |
| MAIL_ADDRESS | findYourOwnEmail@gmail.com |
| MAIL_PW | findYourOwnEmailPassword |

einzutragende Werte ↑

das Tutorial: https://education.launchcode.org/gis-devops-2019/configurations/02-environment-variables-intellij/index.html

Concerning the email account – you can contact us and we will give access to the placetobeer.noreply@gmail.com email

### Fill data base

In the backend, you might notice the group-service/src/main/resources/data.sql file, which contains all the Test data you will be working with.

### Prepare Browser

To setup Angular in WebStorm you need to run npm install in the console. After that, you can start the CLI Angular Webserver.

To allow frontend – backend communication you need to manipulate your browser. We recommend to use the CORS Unblock Add-on for **Chrome Browser** (make sure the C in the Add-On’s logo is orange while testing the application). Sometimes we did have some trouble with it. If this should happen to you, please contact us.

Now you should be prepared to apply the test cases on the next page.
Have fun! 🍻

# Test Cases

For each Test, confirm your action by looking into the database at: http://localhost:8080/h2-console

## 1. Run program

- Run backend and frontend
- Then, open http://localhost:4200/
- The /hubpage should be visible
- (these steps are the foundation of every other test case)

## 2. Switch between groups

- Click on groups on the right side
- Switch active group (members are shown)

## 3. Check out group settings

- Click on ⚙ at active group

## 4. Check role dependency

- Click on ⚙ at active group
- Check out your role in current group (member, admin, owner)
- Inspect what you can do to another group member:

| ROLE | GRANT ADMIN | KICK MEMBER | REVOKE ADMIN |
|---|---|---|---|
| OWNER | Member | Admin, Member | Admin |
| ADMIN | / | Member | / |
| MEMBER | / | / | / |

# 5. Create a group (send invitations)

1.

- Click on Create Group Button
- Create a group without any other member

2.

- Click on Create Group Button
- Add member by inviting people by email
  - (checkout data.sql for existing user emails)
  - (you can also add your own email address, then check if you received an email)
- Create group

# 6. Delete a group

- Delete a group (in group settings)

# 7. Decline an invitation

- Decline a pending Invitation on the right side

# 8. Accept an invitation

- Accept a pending Invitation on the right side
- Refresh, check if (new) group is visible now

**Thank you for participating!**
**Sincerely,**
**the PlaceToBeer Team**