Global Beer Shop
Deployment Instructions

May 2019

SPE Project Team:

Rafael D'Arce - rd17647
Bogdan Stelea - bs17580
Angus Parsonson - ap17691
Charles Figuero - cf17559

Table of contents

Where can I find everything? **1**

How do I run the system? **2**

How do I configure and customise the website for our business? **2**

Payments **4**

Email **5**

Database **5**

Remotely connecting to the existing database **6**

User Interface **8**

We've tried to make it as easy as possible for you to deploy this website and get your business online - however, it does require some basic programming knowledge. We have provided several resources to help you; but feel free to contact any of us if you need any help.

Where can I find everything?

You can download the whole folder directly from the GitHub repository, or you can clone the repository using a Git client on your machine. (You will be required to use Git later on if you want to make edits to the website).
(https://product.hubspot.com/blog/git-and-github-tutorial-for-beginners)

To download all the code, execute the following command on your Command Prompt.

```
git clone https://github.com/bstelea/spe_project
```

How do I run the system?

In order to see the web application running you need to open a terminal window and input the following command:

```
mvn spring-boot:run
```

You need to make sure that you have the *.properties files in the locations specified below in order for the server to be able to run locally. After the command is inputted, you will be able to see the website by putting the url "**localhost:8080"** in your favourite browser.

We already have the website up and running, connected to our own Database and Payment service.

We would strongly recommend having the website on a server you have access to, so that you have better control of it. We would recommend you host the website on Oracle Cloud Services yourselves (https://www.oracle.com/uk/cloud/) as they also provide a great Database hosting service which you will require too. (Explained later)

If you require help migrating the website to your own server, feel free to contact us for help.

How do I configure and customise the website for our business?

The backend of the website is written in Spring and Java. If you want to know more about Spring and be able to make changes yourself to the functionality of the product, have a look at this link (https://spring.io/).

You are already connected to external Email, Database and Payment accounts we had set up for development, which you can use to play about with. But if you actually want to deploy the website so that you can properly manage stock, send emails and handle real transactions, you will have to provide your own instances of these services.

The system uses the *config.properties* and *secret.properties* files to configure its external services:

The secret.properties folder needs to be located in the same folder as the application.properties file, namely spe_project/src/main/resources. The file is composed of the following:

```
db_username = globalbeershop_db_user
db_password = [Password1]
email_username = globalbeershopmail@gmail.com
email_password = beerCan123
```

These are the details needed to connect to the externally hosted mysql database, and the username and password to the email used to receive notifications from the website.

The config.properties file needs to be located in the project directory, namely spe_project and it contains the following:

```
BT_ENVIRONMENT=sandbox
BT_MERCHANT_ID=2ps3z2gtt56w9bp2
BT_PUBLIC_KEY=gc8qc8rd5sdcmww8
BT_PRIVATE_KEY=16ad5f92e136eaa01b8f3aff55aa018c
```

These are the details needed for the Braintree configuration in order to make it possible to process both card and PayPal payments.

Both files are mentioned in the .gitignore file, therefore they are not present in the github repository. That is why they need to be modified directly on the server machine on order to see the changes live.

In order to access the virtual machine using ssh (Secure Shell), you need the public and private key of the VM.

Once these are placed in a directory on a linux running machine, the following commands need to be typed into the terminal to ensure connection to the VM:

```
ssh -i id_rsa ubuntu@129.213.16.132
```

, where “id_rsa” is the name of the private key file and “129.213.16.132” is the IP address of the VM.

Once accessed, the process is straightforward for changing the *.properties files that are not indexed by github - these 2 need to be placed in the home root folder.

Once you have made a change, it can be pushed automatically to our running server thanks to the use of CircleCI.

CircleCI is a free automation tool designed for implementing Continuous Integration and Continuous Deployment. You can create a free account on their website (https://circleci.com/).

CircleCI allows you to automate the process of integration testing and deploying once it detects a push to the git branch that it monitors. In our case, it will run the integration tests and deploy through ssh to the virtual machine that hosts the server.

In order to achieve this type of automation, you will have to create a config.yml file in a folder called .circleci. An example of how such a file should look is present on the github repository of the project.

Furthermore, in order to allow CircleCI to deploy to the virtual machine on which the server is located you need to create environment variables on the dashboard of the platform. Once that is done, you can proceed to editing the config.yml file.

You will have to mention which test classes you wish the script to run when it detects a push. The location in the file where you need to make this change is annotated with **#run tests!.** Then you will have to update the fingerprint of the ssh keys added previously in the configuration file. Then you will have to edit the deployment command.

The main deployment command used is:

```
ssh $SSH_USER@$SSH_HOST "cd spe_project && git pull && cp ~/config.properties . && cp ~/secret.properties ./src/main/resources && mvn clean package -DskipTests=True && cd target && sudo chown gbs:gbs globalbeershop-0.0.1-SNAPSHOT.jar && sudo chmod 500 globalbeershop-0.0.1-SNAPSHOT.jar && sudo service gbs restart"
```

, where “$SSH_USER” and “SSH_HOST” are the user and host names that are referenced on the circleci website dashboard. The next command updates the content of the spe_project folder by pulling the latest changes pushed to git, copies the needed *.properties files in the required places in the project, changes the ownership of the java app created by spring to the user that was assigned to the service responsible for running the Spring server. The last command of the sequence restarts the Spring service and finishes the deployment cycle.

This will ensure a smooth deployment process by blocking any modification made to the code that does not meet the requirements imposed by the test classes to be deployed to the live server, and also remove the hassle of manually accessing the virtual machine that hosts the server, pulling the latest changes, changing the ownership and read-write access of the java file and then restarting the service.

Payments

Braintree is a payment platform provided by PayPal that delivers PayPal, Credit/Debit Cards and popular Digital Wallets (e.g. Apple Pay). (https://www.braintreepayments.com)

We have chosen it as the payment service for GBS as it offers very simple integration into e-commerce websites, as well as provides an online dashboard that allows for easy management of transactions.

The system is currently configured to connect to our own Braintree Sandbox account (Sandbox accounts allow you to experience and test Braintree with fake credit). If you are testing the system in Sandbox mode - as it currently is configured to - then you can use the following fake PayPal details to simulate real transactions:

> Email: rafaeldarce-buyer@hotmail.co.uk
> Password: [Password1]

However, if you want customers to actually be able to buy and pay for beers on your website, you will need to create your own, real Braintree merchant account. Braintree themselves offer extensive documentations and guides to how to set up and use their service. (https://articles.braintreepayments.com/get-started/overview)

Once you have your own merchant account, you will have to update the system with its credentials. (https://articles.braintreepayments.com/control-panel/important-gateway-credentials). In *config.properties,* update the Environment, Merchant ID, and public & private key variables with those from your new Merchant account.

Make sure to change Environment to “Production” (not “Sandbox”) so that real payment methods are used, and that your actual Merchant account is set-up to receive GBP payments.

Email

Emails are crucial for the website - as this is how users can activate the accounts that they registered, how both you and the customers are notified about any orders placed and how your are notified with any “Contact Us” forms submitted through the website..

We are aware that you are already using a Gmail-based Email account for Global Beer Shop business, so we have set up the system to use our own Gmail account.

To configure it to use yours, update the email username and password fields in the *secret.properties* file with that of the real Global Beer Shop email address.

Database

The website is currently using a MySQL 5 database we created with test data for development, we are hosting it on Oracle Cloud Services. You will have to host your own MySQL 5 database so that you can manage all the data yourselves, we would also recommend hosting it on Oracle too. (https://cloud.oracle.com/mysql)

Don’t worry about the schema, our system will automatically create the necessary tables in whatever database it is connected to. So as long as you create the correct type of database and configure the system to connect to it, then it will work.

In *secret.properties*, update the appropropriate variables with your new databases URL along with the login (username and password) details you use to access it.

Make sure your database’s firewall settings allow for connections to it from the internet.

Remotely connecting to the existing database

The database that is in use right now is located on another virtual machine created in the same “Oracle My Services” suite of services. You can read more about what a virtual machine here (https://www.howtogeek.com/196060/beginner-geek-how-to-create-and-use-virtual-machines/).

In order to connect to the machine and make changes manually to the database, we will use a similar approach that was used before in order to connect to the server to modify the *.properties files; we will connect to the virtual machine through ssh.

In order to do that , we again need the public and private key that allow us to connect to the virtual machine through ssh. After placing the two files in the same directory on a linux running machine and opening a terminal window in that directory, you need to input the following command:

```
ssh -i privateKey opc@129.150.126.140
```

, where “privateKey” is the name of the private key file and “129.150.126.140” is the IP address of the virtual machine that hosts the database. For more details on how to set up this machine and install the mysql server on it check the following links:

- https://www.oracle.com/webfolder/technetwork/tutorials/obe/cloud/mysql/getting-started/getting-started.html
- https://www.oracle.com/webfolder/technetwork/tutorials/obe/cloud/mysql/connecting/connecting.html

Once connected to the virtual machine we have to input a series of commands in the terminal. First off

```
sudo su - oracle
```

, will change the user to oracle, as the machine was set up initially

```
mysql -u root -p
```

, will ensure the connection to the mysql database, when prompted to enter the password, type “[Password1]”

From here you can see all the databases that can be used from this server. In order to make changes to the live website database contents use the globalbeershop_db database by inputting the command

```
use globalbeershop_db;
```

From here you can see all the tables that are automatically generated by the Spring server, recognized from the @Entity annotation present in the code.

The script used for entering the default values for the beers in the shop is the following:

```
INSERT INTO beer (name, price, stock, country, brewer, type, abv, image, description,
continent)
VALUES ('Heineken', 5.70, 3, 'Netherlands', 'Heineken Brewers', 'Lager', 5.0,
'/img/beer-template.jpg', 'This is the description for the Heineken beer', 'Europe');
INSERT INTO beer (name, price, stock, country, brewer, type, abv, image, description,
continent)
VALUES ('Coors Light', 5.0, 5, 'United States', 'Coors Brewing', 'Lager', 4.2,
'/img/beer-template.jpg', 'This is the description for the Coors Light beer', 'North Asia');
INSERT INTO beer (name, price, stock, country, brewer, type, abv, image, description,
continent)
VALUES ('Guinness', 6.0, 20, 'Ireland', 'Diageo', 'Stout', 4.2, '/img/beer-template.jpg', 'This
is the description for the Guinness beer', 'Europe');
INSERT INTO beer (name, price, stock, country, brewer, type, abv, image, description,
continent)
VALUES ('Bergenbier', 2.0, 58, 'Romania', 'Bergenbier Carpatian', 'Lager', 4.0,
'/img/beer-template.jpg', 'This is the description for the Bergenbier beer', 'Europe');
INSERT INTO beer (name, price, stock, country, brewer, type, abv, image, description,
continent)
VALUES ('Ursus', 1.5, 40, 'Romania', 'Ursus Carpatian', 'Lager', 4.2,
'/img/beer-template.jpg', 'This is the description for the Ursus beer', 'South America');
INSERT INTO beer (name, price, stock, country, brewer, type, abv, image, description,
continent)
VALUES ('Timisoreana', 1.0, 45, 'Romania', 'Timisoreana Carpatian', 'Lager', 4.2,
'/img/beer-template.jpg', 'This is the description for the Timisoreana beer', 'Oceania');
INSERT INTO beer (name, price, stock, country, brewer, type, abv, image, description,
continent)
VALUES ('Corona', 3.0, 34, 'Mexico', 'Corona Cerveza', 'Lager', 4.2,
'/img/beer-template.jpg', 'This is the description for the Corona beer', 'North America');
INSERT INTO beer (name, price, stock, country, brewer, type, abv, image, description,
continent)
VALUES ('Staropramen', 5.0, 5, 'Czech Republic', 'Staropramen Brewing', 'Lager', 4.2,
'/img/beer-template.jpg', 'This is the description for the Staropramen beer', 'Middle East');
INSERT INTO beer (name, price, stock, country, brewer, type, abv, image, description,
continent)
VALUES ('Neumarkt', 0.5, 5, 'Romania', 'Backyard', 'Lager', 4.2, '/img/beer-template.jpg',
'This is the description for the Neumarkt beer', 'Africa');
INSERT INTO beer (name, price, stock, country, brewer, type, abv, image, description,
continent)
VALUES ('Fosters', 2.5, 5, 'United Kingdom', 'Fosters Brewing', 'Lager', 4.2,
'/img/beer-template.jpg', 'This is the description for the Fosters beer', 'South Asia');
INSERT INTO beer (name, price, stock, country, brewer, type, abv, image, description,
continent)
```

VALUES ('Silva', 0.7, 63, 'Romania', 'Silva Brewing', 'Lager', 4.2, '/img/beer-template.jpg', 'This is the description for the Silva beer', 'Europe');

, where you can see that the values of the specific product corresponds to the fields that a beer entity has.

If you wish to make your own changes to the current database, you can simply modify the above command to include the beers that you want to be featured on your website. For further custom changes to the database contents, please use mysql commands. There are plenty of resources available, and each question regarding changes to the values present in the database can be addressed by a simple Google search.

User Interface

All of the text, images and general appearance you see when you visit the website are implemented using HTML, CSS and JavaScript - they are relatively simple to understand with the following tutorials (https://www.w3schools.com/html/, https://www.w3schools.com/css/) so you can make any changes you want to the website content.

The files for the User Interface of the website are all located in *src → main → resources → templates.*

The directory spe_project/src/main/resources/templates contains all the .html files contained in the project. Each one of them has a name corresponding to the page it links to, e.g. index.html corresponds to the home page, shop.html corresponds to the shop page etc. You will also see that in order to create webpages that are dynamically populated with objects created on the backed Spring server, such as the beers that are available in the database being displayed on the shop page in a grid, we have used the Thymeleaf template engine in addition to HTML. In order to find out more about Thymeleaf syntax and its use, check the following link(https://www.thymeleaf.org/).

Also, in order to ensure a better layout and a smoother transition to the mobile responsive version of the website we have used Bootstrap. Again, to check more about it follow this link (https://getbootstrap.com/).

The static directory of the resources folder contains all the CSS and JavaScript files that were used to create the animations in use on the website.