



**CSCI 5308 – Advanced Topics in Software Development**

**Group 18 – Budget Surfing**

**External Dependencies Document**

**Work done by,**

- 1. Parth Dhirubhai Gondaliya, B00893694**
- 2. GRM Vishnu, B00871849**
- 3. Hardee Rameshchandra Garala, B00869195**
- 4. Tuan Hamid, B00877065**
- 5. Neelansh Gulati, B00892946**

## **External Dependencies Configuration:**

### **Where to find configurations?**

All external key value pairs are stored in the application.properties file inside resources. The following screenshot is a visual aid to find the location.

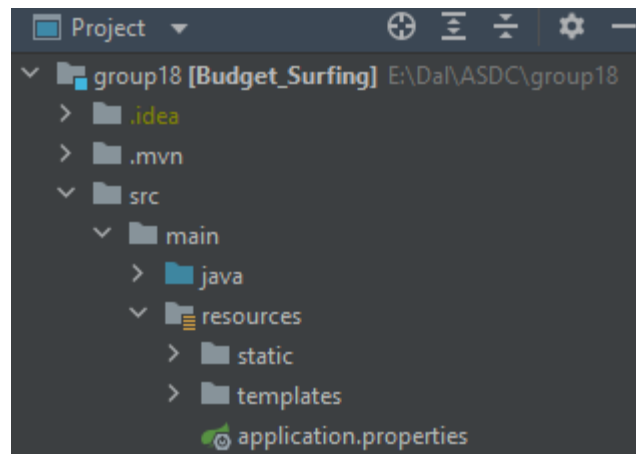


Figure 1 Configuration location

### **Email Sending**

Currently, the project uses “spring-boot-starter-mail” to send emails. This requires a valid smtp configuration to act as the email server. We are using the free SMTP service by Gmail for this purpose. One online guide available to setup Gmail for SMTP can be found at <https://www.jotform.com/help/392-how-to-use-your-gmail-account-as-your-email-sender-via-smtp/> [1].

Configuration for settings are stored in application.properties as seen in the following screenshot.

```
spring.mail.host=smtp.gmail.com
spring.mail.port=587
spring.mail.username=gondaliyaparth12345@gmail.com
spring.mail.password=tzowjxmznzudwxqi
spring.mail.properties.mail.smtp.auth=true
spring.mail.properties.mail.smtp.starttls.enable=true
```

Figure 2 Email configuration

This dependency is used in user module

(com.group18.listener.RegistrationListener)and notifications module

(com.group18.listener.TripListener).

## **Payments (Stripe)**

This project uses stripe-java to act as the external payment processor. As not actual payments are carried out at the moment, we have configured to Stripe's test environment to prevent actual charges occurring. For initial generation of Stripe, a user will have to:

1. Sign up for Stripe account
2. Obtain developer keys by going to <https://dashboard.stripe.com/test/apikeys>
3. Switch Live and Test toggle as required for correct environment keys

Configurations for the Stripe public key and secret key are stored in `application.properties` as shown below.

```
STRIPE_PUBLIC_KEY=pk_test_51Kg0pZ6WjFP75A0d8NsU2RVzgb0GKAVY7hok6faZ0Jw8ldStktP48bbnWT103A1I9U7jFPmvYGIAsDPXzPZN9
STRIPE_SECRET_KEY=sk_test_51Kg0pZ6WjFP75A0dRSJvfCkeZ46Go1Rhpr0f6NsW1xodoc96y9B4cNepbXDPtV3IMhPk93DPHMQsARAjgDKfs
```

Figure 3 Payment configuration

This dependency is used for the payments module and is found in at `com.group18.serviceImpl.StripeServiceImpl`

## Database

This project uses “spring-boot-starter-data-jpa” and “mysql-connector-java” to connect our spring boot project with a MySQL database. Update database property inside `application.properties` located inside resources to required database if needed (default Dalhousie remote server).

```
spring.datasource.url=jdbc:mysql://db-5308.cs.dal.ca:3306/CSCI5308_18_DEVINT?useUnicode=true&useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&serverTimezone=UTC
spring.datasource.username=CSCI5308_18_DEVINT_USER
spring.datasource.password=HugoodaeHei8oob0
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect
spring.jpa.hibernate.ddl-auto=update
```

Figure 4 Database configuration

For a first start of new database via spring, the following property should be used `spring.jpa.hibernate.ddl-auto=create` . Any subsequent starts should change the property to “update” instead of “create”.

The following SQL queries need to be executed on a fresh database server as they consist of system values used by the project.

**INSERT INTO ROLES VALUES (1,'ROLE\_HOST');**

**INSERT INTO ROLES VALUES (1, 'ROLE\_TRAVELLER');**

### **Other Spring Project Dependencies:**

<b>Dependency</b>	<b>Reason</b>
spring-boot-starter-web	Used to build REST application with embedded Tomcat server
spring-boot-starter-web-services	Starter for Spring web services
spring-boot-starter-security	Used for Spring security enforcement
jjwt	Add JSON Web Token support
modelmapper	Used for object mapping
spring-boot-devtools	Developer tools for Spring Boot
spring-boot-starter-test	Used to test application
spring-boot-starter-data-jpa	For using JPA with Hibernate
mysql-connector-java	JDBC driver for MySQL
spring-boot-starter-mail	Starter for Java mail and sending email
spring-boot-starter-thymeleaf	Render Thymeleaf views
springdoc-openapi-ui	Used for API documentation
junit-jupiter	Junit 5 for testing
mockito-core	For mocking objects
stripe-java	Handling stripe integration

## **References:**

- [1]     "*How to Use Your Gmail Account as Your Email Sender via SMTP*," Jotform [Online]. Available at: <https://www.jotform.com/help/392-how-to-use-your-gmail-account-as-your-email-sender-via-smtp/> [Accessed: April 5, 2022].