



CHALLENGE 2 – WEEK 8

INSTRUCTIONS

- **E-mail with challeng link must be sent by:** July 12, 2023 at 6 p.m.
- **Last commit:** July 12, 2023 at 6 p.m.
- The challenge is in a group
- Apply the agile methodology, choosing a Scrum Master (SM) and Product Owner (P.O.)
- Everyone should play the role of developer
- The Scrum Master will be responsible for the dailies and for organizing Trello.
- The Product Owner will be responsible for creating the stories, creating the repository on GitHub and sending the repository link by email to other team members and instructors.

E-mail

- **To:** include instructors and team members
- **“C.C” (with copy):** include SM and coordinators
- **Email subject:** SP – Challenge 02 – Squad 0 – Team Name
- **Email body:** should contain team member’s names, their roles, and Trello and GitHub links.



Scholarship Program
SP – Backend Journey (Spring Boot)
AWS Cloud Context
CHALLENGE

Para	Juliane Valiati Maranhão ×	Mateus de Oliveira e Silva ×	Mateus Cardoso de Moraes ×	Cassiano Jovino Mandelli Stefani ×
Cc	Liliv Hana Vasconcelos ×	Guilherme Moreira de Paiva ×	Vitor de Oliveira Mendes ×	Fabiane Telles Leite Maciel ×

Scholarship Program – Challenge 02 – Squad 0 – Team Name

Name of team members:

- Name A (Scrum Master / Developer)
- Name B (Product Owner / Developer)
- Name C (Developer)
- Name D (Developer)
- Name E (Developer)

Links:

- Trello:
- GitHub Repository:

SETTINGS

- Java 17
- Spring Boot 3 and Spring 6
- **Group (groupId):** br.com.compassuol.pb.challenge
- **Artifact (artifactId):** ecommerce
- **Repository name:** CompassUOL-SP-Challenge-02-Squad-0-Team-Name

Project

☐ Gradle - Groovy
 ☒ Gradle - Kotlin
 ☒ Maven

Language

☒ Java
 ☐ Kotlin
 ☐ Groovy

Spring Boot

☐ 3.1.1 (SNAPSHOT)
 ☒ 3.1.0
 ☐ 3.0.8 (SNAPSHOT)
 ☐ 3.0.7
 ☐ 2.7.13 (SNAPSHOT)
 ☐ 2.7.12

Project Metadata

Group

br.com.compassuol.sp.challenge

Artifact

ecommerce

Name

ecommerce

Description

Demo project for Spring Boot

Package name

br.com.compassuol.sp.challenge.ecommerce

Packaging

☒ Jar
 ☐ War

Java

☐ 20
 ☒ 17
 ☐ 11
 ☐ 8

Note:

- Scholarship Program or SP
- Using minus sign (-) or underscore (_)
- Avoid using space



CONTACTS

Instructors

Carlos Fernando Gonçalves (Java Spring Boot + AWS)

- E-mail: carlos.goncalves@compasso.com.br

Cassiano Jovino Mandelli Stefani (Java Spring Boot + AWS)

- E-mail: cassiano.stefani@compasso.com.br
- GitHub: <https://github.com/CjMS2103>

Juliane Valiati Maran (Java Spring Boot)

- E-mail: juliane.maran@compasso.com.br
- GitHub: <https://github.com/JulianeMaran32>

Mateus Cardoso de Moraes (Java Spring Boot)

- E-mail: mateus.moraes@compasso.com.br
- GitHub: <https://github.com/mateusCoder>

Mateus de Oliveira e Silveira (Java Spring Boot)

- E-mail: mateus.silveira@compasso.com.br
- GitHub: <https://github.com/mosilveira>

Scrum Master

Guilherme Moreira de Paiva

- E-mail: guilherme.paiva@compasso.com.br

Vitor de Oliveira Mendes

- E-mail: vitor.mendes@compasso.com.br



Coordinators

Fabiane Telles Leite Maciel

- Email: fabiane.maciел@compasso.com.br

Liliv Hana Vasconcelos

- E-mail: liliv.vasconcelos@compasso.com.br



CHALLENGE DESCRIPTION

E-Commerce intends to improve its system. The company hires you and your team to implement a program that includes customers, payment, orders and products management service. Having the following requirements:

- Relational Database Connection (PostgreSQL or MySQL)
- Error Handling
- Unit tests with a minimum coverage of 70%

1. CUSTOMER

REST API Customer is responsible for managing the Customer Registration.

Endpoints

Method	Base path
POST	/v1/customers
GET	/v1/customers/:id
PUT	/v1/customers/:id

Customer Table

Attribute	Type	Validation
customerId	Numeric	ID should be generated automatically
name		Must be at least 3 characters
cpf		It must be a valid and unique CPF for each customer
email		It must be a valid and unique email for each customer
active	Boolean	Must only accept Boolean value (true or false)

Payload – Request

```

1  {
2    "name": "Cristiano Ronaldo",
3    "cpf": "12345678910",
4    "email": "cristiano.ronaldo@email.com",
5    "active": true
6  }
```



2. PRODUCTS

REST API Products is responsible for managing the Product Registration.

Endpoints

Method	Base path
GET	/v1/products
POST	/v1/products
GET	/v1/products/:id
PUT	/v1/products/:id
DELETE	/v1/products/:id

Product Table

Attribute	Type	Validation
productId	Numeric	ID should be generated automatically
name		Must be at least 3 characters
price		Product value. Example: BRL 9.99 or R\$ 9,99
description		Must be at least 3 characters
Note: all fields are mandatory!		

Payload – Request

```
1  {  
2    "name": "Shirt",  
3    "description": "M cotton shirt",  
4    "price": 50.00  
5  }
```




3. ORDER

REST API Order is responsible for managing Purchase Orders.

Endpoints

Method	Base path
GET	/v1/orders
POST	/v1/orders
GET	/v1/orders/customers/:customerId

Order Table

Attribute	Type	Validation
orderId	Numeric	ID should be generated automatically
customerId	Customer	
date		ISO 8601
status	Enum	Must accept only CREATED and CONFIRMED values
Note: All fields are mandatory! It must not allow the registration of a non-existent customer or product		

Payload – Request

```
1  {
2    "customerId": 1,
3    "products": [
4      {
5        "productId": 1,
6        "quantity": 2
7      },
8      {
9        "productId": 2,
10       "quantity": 5
11     }
12   ],
13   "date": "2023-06-01",
14   "status": "CREATED"
15 }
```

4. PAYMENT

REST API Payment is responsible for managing Payments.

Endpoints

Method	Base path
POST	/v1/payments

Payment Table

Attribute	Type	Validation
paymentId	Numeric	ID should be generated automatically
paymentMethod	Enum	CREDIT_CARD, DEBIT_CARD, TRANSFER, PIX and CASH
paymentDate		ISO 8601
orderId	Order	
Note: All fields are mandatory! When registering a payment, the status of the order must be changed to CONFIRMED.		

Payload – Request

```

1  {
2    "orderId": 1,
3    "paymentMethod": "CREDIT_CARD",
4    "paymentDate": "2023-06-01"
5  }
```