Hotel Booking Application
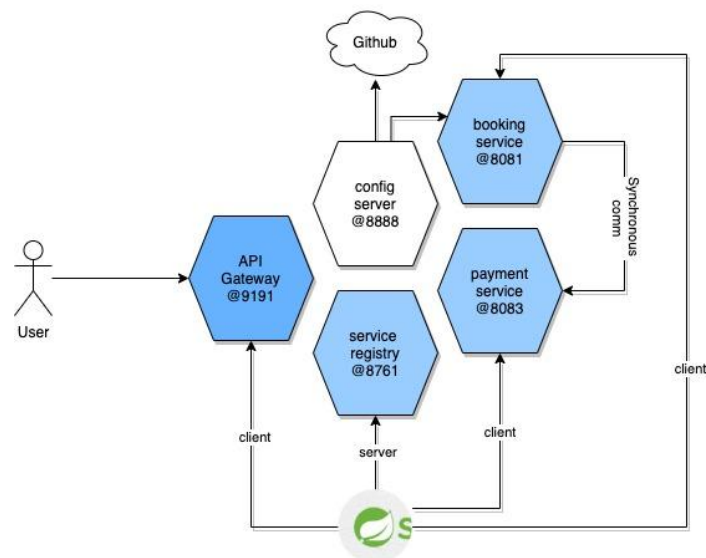
Architecture

Code Overview

- Payment Service
  - Special Maven Dependencies:

```xml
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-config</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-bootstrap</artifactId>
</dependency>
```

  - Code Overview:
- Booking Service
  - Special Maven Dependencies:

```xml
<!-- Additional dependencies Start -->
<dependency>
    <groupId>org.modelmapper</groupId>
```

```xml
    <artifactId>modelmapper</artifactId>
    <version>2.4.4</version>
</dependency>
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-feign</artifactId>
    <version>1.4.7.RELEASE</version>
</dependency>
<dependency>
    <groupId>org.jetbrains</groupId>
    <artifactId>annotations</artifactId>
    <version>RELEASE</version>
    <scope>compile</scope>
</dependency>
<dependency>
    <groupId>joda-time</groupId>
    <artifactId>joda-time</artifactId>
    <version>2.9.7</version>
</dependency>
<!-- for config-server:Start-->
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-config</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-bootstrap</artifactId>
</dependency>
<!-- for config-server:End-->
<!-- for AOP:Start-->
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-aop</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-validation</artifactId>
</dependency>
<dependency>
    <groupId>org.apache.commons</groupId>
    <artifactId>commons-lang3</artifactId>
</dependency>
<!-- for config-server:End-->
<!-- Additional dependencies End -->
```

○ Code Overview:

The booking service is the main processing logic of the microservice architecture.
It has following features,

- Invokes the Feign Client to include the references of Payment Service
- Performs Create and Update operation on the booking entity
- It also invoke the Create operation of the payment service
- It generates a random number to provide the list of the rooms booked in a given request.
- It performs the AOP based exception handling that throws an error if

- ● Booking Id for a selected payment doesn't exist
- ● Payment method is invalid (it must be UPI or CARD)

- ● Config Server
  - ○ Special Maven Dependencies:

```xml
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-config-server</artifactId>
</dependency>
```

  - ○ Code Overview:
- ● Service Registry
  - ○ Special Maven Dependencies:

```xml
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-netflix-eureka-server</artifactId>
</dependency>
```

  - ○ Code Overview:
    The service registry port is a way to provide the microservices running (which have references of eureka client)

- ● API Gateway
  - ○ Special Maven Dependencies:

```xml
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-actuator</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-gateway</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
</dependency>
```

  - ○ Code Overview:
    N/A

Deployment Sequence

- Config Server
- Service Registry
- API Gateway
- Payment Service
- Booking Service