

# OpenServiceBroker Redis

# Inhaltsübersicht

## 1. Übersicht

1.1 Projektstruktur

1.2 Core

1.3 Model

1.4 Helm

1.5 Security

1.6 Redis

1.7 Tests

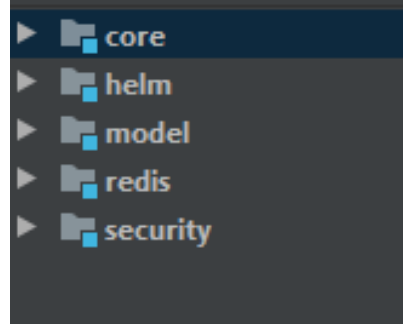
## 2. Demo

## 3. Lessons Learned

# 1. Übersicht

# 1.1 Projektstruktur

- Java basierte Spring Boot Applikation
- Modularer Aufbau (modul  $\hat{=}$  package)



## 1.2 Core

- ServiBrokerImpl.java
  - Zuständig für die Endpoints
  - Ruft HelmDeployer zum Erzeugen der Services

# 1.3 Model

- Daten für ServiceCatalog im ServiceSchema.json

## 1.4 Helm

- HelmDeployer lädt Helmchart und befüllt es gemäß der auszuliefernden Größe.
- Funktioniert mit MicroBean Helm
- Schnittstelle zu Helm / Kubernetes

# 1.5 Security

- Spring boot starter security
- Basic http Authentifizierung



# 1.6 Redis

- Spring ApplicationContext

```
@SpringBootApplication(exclude={SecurityAutoConfiguration.class},  
                        scanBasePackages = {"de.thbin.epro"})  
public class Application {
```

# 1.7 Tests

- osb-checker-kotlin
- Integrationsstests

```
@SpringBootTest(webEnvironment = SpringBootTest.WebEnvironment.DEFINED_PORT)
@RunWith(SpringRunner.class)
public class ApplicationTest {

    @Autowired
    TestRestTemplate restTemplate;
```

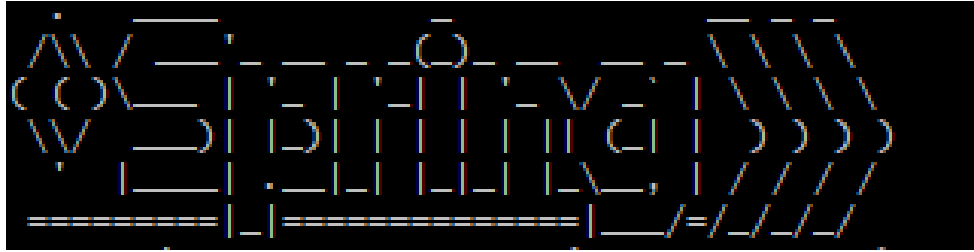
## 2. Demo

# 3. Lessons Learned

# 3. Lessons Learned Christian



Git + GitHub



SPRING

# 3. Lessons Learned

## Jannik

- Je mehr Bestandteile in der Anwendung, desto schwieriger die Fehlersuche.
- Ins Detail gehen für jeden Bestandteil!
- Vorsicht vor „Aufgabenkapselung“

# 3. Lessons Learned

## Maximilian

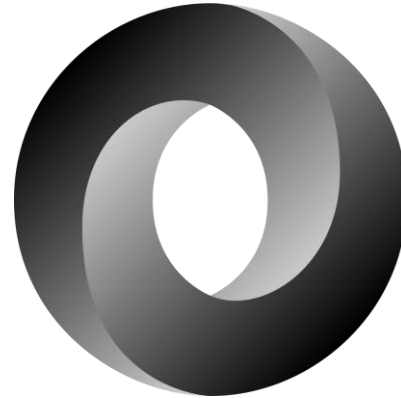
- Git sauber einrichten
- Code Reviews
- Häufiger Austausch untereinander (Ergebnisse, Fortschritt etc)

# 3. Lessons Learned

## Michael



Git + Github



Json



# Fragen?