

hazelcast-examples (0.0.1)

Maksim Kostromin

Version 0.0.1, 2018-12-04 04:41:45 UTC

Table of Contents

Introduction.....	1
Implementation	2
app.....	2
pub-sub.....	2
kryo (subzero).....	3
Links	4

Introduction

docker / docker-compose

```
./mvnw
```

Resources:

- [Hazelcast getting started](#)

Implementation

app

TODO...

pub-sub

```
public static final HazelcastInstance instance = createInstance();
public static final HazelcastInstance client = createClient();
@Slf4j
@Configuration
class PubSubHzCfg {
    @Bean(name = "hzInstance", destroyMethod = "shutdown")
    HazelcastInstance hzInstance() {
        return Hz.instance;
    }
}
```

```
@Bean(name = "domainEventsTopic")
ITopic<DomainEvent> domainEventsTopic(@Qualifier("hzInstance") final HazelcastInstance
hzInstance) {
    return hzInstance.getTopic("domainEventsTopic");
}
```

```
@Bean
ApplicationRunner applicationRunner(@Qualifier("domainEventsTopic") final
ITopic<DomainEvent> domainEventsTopic) {
    return args -> domainEventsTopic.addListener(
        message -> log.info("received message: {}",
            message.getMessageObject(),
            message.getPublishingMember(),
            new Date(message.getPublishTime())));
}

final String topicName = "domainEventsTopic";
final HazelcastInstance client = Hz.client;
final Function<String, ITopic<DomainEvent>> topic = client::getTopic;
final Function<String, DomainEvent> data = message -> DomainEvent
    .builder()
    .createdAt(LocalDateDateTime.now())
    .type(DomainEvent.Type.NONE)
    .jsonData(JSON.stringify(singletonMap("message", message)))
    .build();
```

```

if (args.length > 0) {
    final String message = String.join("", args);
    topic.apply(topicName)
        .publish(data.apply(message));
}

```

kryo (subzero)

```

public static final Config config = createConfig();

```

```

private static Config createConfig() {
    final Config config = new Config();
    SubZero.useAsGlobalSerializer(config);
    return config;
}
@Value
@Builder
@JsonIgnoreProperties(ignoreUnknown = true)
public class DomainEvent /* implements Serializable */ {
    public enum Type {
        NONE
    }
}

```

```

// we don't need any serialization, because SubZero Kryo was added globally
// // private static final long serialVersionUID = -3158689686565990234L;
@JsonFormat(pattern = "yyyy-MM-dd'T'HH:mm:s")
@JsonDeserialize(using = LocalDateTimeDeserializer.class)
@JsonSerialize(using = LocalDateTimeSerializer.class)
final LocalDateTime createdAt;
final Type type;
final String jsonData;
}

```

```

./mvnw clean install -f ./subzero-kryo-app/pom.xml
./subzero-kryo-app/subzero-kryo-app-backend/target/*.jar &
http :8080 ololo=trololo
http :8080 hoho=ofo
http :8080

```

TODO: implement documentation...

Links

- [GitHub repo](#)
- [GitHub pages](#)