



Rabbit MQ



Mottagnings klassen annoteras `@Service` och minst en metod ska annoteras med `@RabbitListener`. Nu kan metoden lyssna på en kö som man definerar i sin konfiguration.

Service klassen måste anropas med `@Autowired` av en klass som är annoterad med `@EnableRabbit`. Denna klass måste ha minst två `@Bean` annoteringar med standard metoder för att tolka meddelanden.



Sender klassen måste annoteras @Service följt av

```
@Autowired
```

```
private RabbitMessagingTemplate rabbitMessagingTemplate;
```

```
@Autowired
```

```
private MappingJackson2MessageConverter converter;
```

Därefter är det "bara" att skapa metoder som skickar ut data till kanalen.



För att integrera JMS meningsfullt i projektet med minsta möjliga jobb, gör ett enkel extra SpringBoot projekt som lyssnar av på en kanal, som tar emot t.ex ett namn på åden som loggar in, och sen skriver in en timestamp och skickar det till en databas.

Det behövs således inget större värde i säkerhet, meningen är att ni ska ha en humm om vad JMS är och hur det används. I verkliga livet hade man fått bygga mer säkerhet i det men till projektet är tidsgränsen rätt snäv som den är.