

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE  
FAKULTA ELEKTROTECHNIKY A INFORMATIKY**

**SPRING AMQP & SPRING SECURITY  
SEMINÁRNA PRÁCA**

**2022**

**Adam Klimko, Andrej Mikuš, Marek Mlynček**

# Obsah

<b>1</b>	<b>AMQP</b>	<b>1</b>
1.1	Čo je AMQP a ako funguje . . . . .	1
1.2	História AMQP . . . . .	1
1.3	AMQP terminológia . . . . .	2
1.4	Komponenty AMQP . . . . .	3
1.5	Vývoj API s AMQP . . . . .	3
<b>2</b>	<b>Spring AMQP</b>	<b>5</b>
<b>3</b>	<b>Spring Security</b>	<b>6</b>
	<b>Zoznam použitej literatúry</b>	<b>7</b>

# Zoznam skratiek

**AMQP**    Advanced Message Queuing Protocol

# 1 AMQP

Bezplatné a rýchle operácie open-source nástroja z nich urobili preferovanú voľbu pred ich rovesníkmi s uzavretým zdrojom. Bez toho, aby boli používateľom kladené nejaké tvrdé a rýchle obmedzenia, aplikácie open-source sa v súčasnosti stali normou. AMQP Standard je bežne používaný protokol na odosielanie správ používaný v procese vývoja aplikácií s otvoreným zdrojovým kódom. V tejto kapitole osvetlíme jeho význam, užitočnosť a kľúčovú terminológiu.

## 1.1 Čo je AMQP a ako funguje

AMQP označuje Advanced Message Queuing Protocol. Globálne uznávaný štandard, ktorý v podstate funguje na aplikačnej vrstve, používa sa hlavne na rozvoj bezkonkurenčnej komunikačnej prevádzkyschopnosti medzi klientom a makléorskými stranami. Vydavateľ nesie zodpovednosť za generovanie správ, kým ich klienti zhromažďujú a spravujú. Úlohou brokerov, ako je RabbitMQ, v tomto celom procese je zabezpečiť, aby správa, ktorá je súčasťou výmeny, prechádzala priamo od vydavateľa ku klientovi. Keď už hovoríme o jeho kľúčových funkciách, smerovanie, orientácia správ a radenie do frontu sú tie najdôležitejšie. Použitie AMQP robí interoperabilitu dosiahnutým cieľom s rôznymi konfiguráciami a infraštruktúrou. Umožňuje vývojárom uviesť do činnosti všetky klientske knižnice a sprostredkovateľa, ktoré sú v súlade s protokolom.

## 1.2 História AMQP

AMQP, ktorý prvýkrát vytvoril John O' Hara v roku 2003, sa od svojho vzniku veľmi vyvinul a má bohatú históriu. Kým John založil AMQP, pracoval s JPMorgan Chase a spoločnosť si ponechala všetky práva AMQP a použila ich na uzavretie zmluvy s iMatrix Corporation. Obe tieto dve firmy použili AMQP na návrh dokumentácie protokolov, ako aj C brokerov. Toto používanie AMPQ pokračovalo až do polovice roku 2006. Spoločnosť JPMorgan Chase sa však rozhodla urobiť AMQP slávnym a pripraveným na použitie aj pre iné firmy. To je dôvod, prečo sa v polovici roku 2005 spoločnosť snažila vytvoriť skupinu používateľov pre AMQP, ktorá zahŕňala špičkových IT gigantov ako Cisco, Red Hat, IONA Technologies a TWIST. Skupina sa pomaly rozširovala a predstavovala 23 spoločností. Niektorí z nových členov boli Barclays, IIT Software, Bank of America, Credit Suisse, Progress Software a Informatica. AMQP 1.0 bol spustený v októbri 2011 a krátko nato, teda 1. novembra, bol svetu predstavený technický výbor OASIS. Jeho hlavnou zodpovednosťou bolo dohliadať na pokrok AMQP 1.0. Prvý a druhý návrh boli podľa

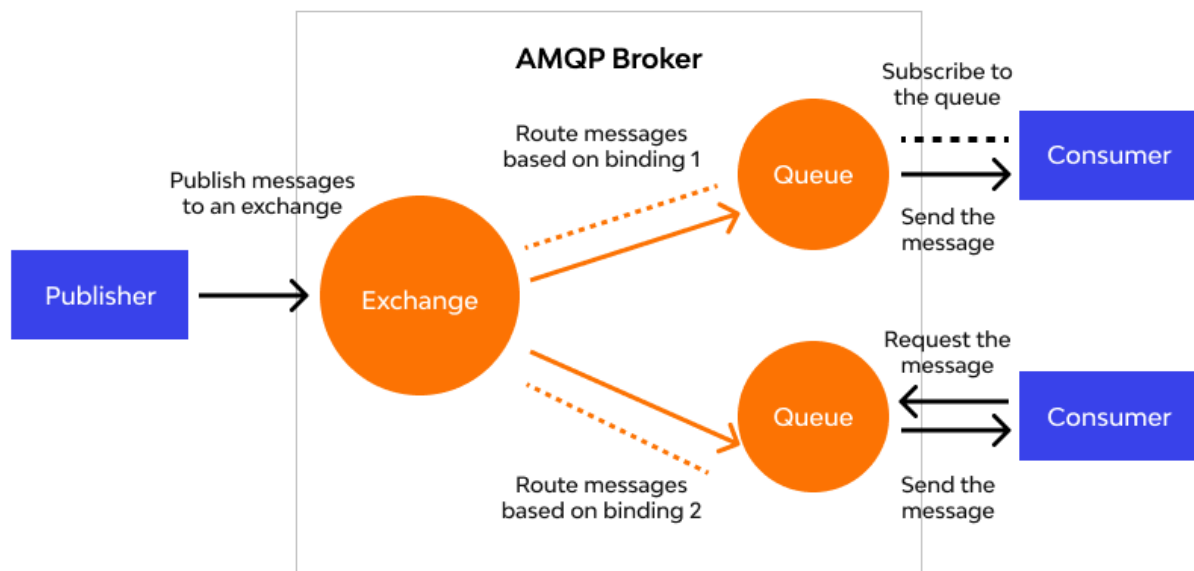
spätnej väzby pracovného výboru spustené vo februári 2012 a v júni 2012. O niekoľko mesiacov neskôr, koncom októbra 2012, sa AMQP stal autorizovaným štandardom OASIS. V roku 2014 bol štandard OASIS AMQP akceptovaný aj medzinárodnými štandardmi ako IEC a ISO.

### 1.3 AMQP terminológia

Keď človek plánuje uviesť AMQP do praxe, je nevyhnutné poznať niektoré z jeho kľúčových terminológií. V tejto časti príspevku sme vysvetlili tie najpoužívanéjšie.

- Broker (alebo server) hrá kľúčovú úlohu pri aktivácii protokolu AMQP. Je zodpovedný za budovanie spojení, ktoré zaisťujú lepšie smerovanie údajov a radenie do frontu na strane klienta.
- O generovanie frontov a potvrdzovanie správ sa stará spotrebiteľ.
- O presmerovanie údajov prevzatých z búrz a ich zaradovanie do frontov sa stará výrobca.

Obr. 1: AMQP Broker



## 1.4 Komponenty AMQP

Okrem vyššie uvedenej terminológie AMQP si vývojári musia byť vedomí určitých komponentov používaných na nasadenie modelu AMQP.

### **Exchange**

Exchange nesie zodpovednosť za načítanie správ a ich starostlivé umiestňovanie do správneho frontu. Jeho 4 kategórie sú: Fanout, Hlavičky, Téma a Priame. Aby sme vás o tom podrobnejšie informovali, je to nenahraditeľná súčasť brokera.

### **Channel (Kanál)**

Kanál označuje multiplexné virtuálne pripojenie medzi partnermi AMQP, ktoré je vybudované v rámci existujúceho pripojenia.

### **Message queue (Fronta správ)**

Je to identifikovaná entita, ktorá pomáha prepojiť správy s ich zdrojmi alebo miestom pôvodu.

### **Binding (Väzba)**

Väzby označujú množinu preddefinovaných inštrukcií týkajúcich sa frontov, ako aj výmen. Spravuje odosielanie správ a ich doručovanie.

### **Virtual host**

Virtual host je platforma, ktorá ponúka segregáčné zariadenie vo vnútri makléra. Na základe používateľov a ich prístupových práv môže byť naraz funkčných viacero vhostov.

## 1.5 Vývoj API s AMQP

Pomocou AMQP môžu rozhrania API:

- Priamo odosielať správy
- Ukladať správy do vyrovnávacej pamäte vo fronte na odosielanie na základe spúšťača;
- Môže smerovať informácie alebo spájať výmeny s určenými frontami;
- Vytvára spojenie s cieľom zabezpečiť efektívnu komunikáciu;
- Môže poslať automatické alebo manuálne potvrdenie.

Môžete zvážiť niekoľko scenárov, v ktorých možno AMQP použiť v rozhraniach API, napríklad:

- Pridávanie peňazí do svojich digitálnych peňaženiek
- Transakcia kreditnou alebo debetnou kartou v maloobchodných predajniach

- V messengeroch alebo komunikačných systémoch

Výmena správ využívaná AMQP je asynchrónna. Keďže transakcie sú rozhodujúce udalosti a nechcete v nich mať nekonzistentnosť, môžu existovať ďalšie servisné udalosti, ktoré budú sledovať výsledok každej správy, aby sa ďalšia udalosť mohla spustiť včas. Asynchrónna povaha AMQP spôsobí potrebu spätných volaní v porovnaní s HTTP. Je tiež zložitejšie navrhnúť koncové body „načítania“ pomocou tohto protokolu [1].

## 2 Spring AMQP

Projekt Spring AMQP aplikuje základné koncepty Spring na vývoj riešení na odosielanie správ založených na AMQP. Poskytuje „šablónu“ ako vysokoúrovňovú abstrakciu na odosielanie a prijímanie správ. Poskytuje tiež podporu pre POJO riadené správami s „kontajnerom poslucháča“. Tieto knižnice uľahčujú správu zdrojov AMQP a zároveň podporujú používanie vkladania závislostí a deklaratívnej konfigurácie. Vo všetkých týchto prípadoch uvidíte podobnosti s podporou JMS v Spring Framework. Projekt pozostáva z dvoch častí; spring-amqp je základná abstrakcia a spring-rabbit je implementácia RabbitMQ [2].

Vlastnosti:

- Kontajner poslucháča na asynchrónne spracovanie prichádzajúcich správ
- RabbitTemplate na odosielanie a prijímanie správ
- RabbitAdmin pre automatické deklarovanie frontov, výmen a väzieb



## 3 Spring Security

Spring Security je výkonný a vysoko prispôsobiteľný rámec autentifikácie a kontroly prístupu. Je to de-facto štandard pre zabezpečenie aplikácií na báze Springu. Spring Security je framework, ktorý sa zameriava na poskytovanie autentifikácie a autorizácie pre Java aplikácie. Rovnako ako všetky projekty Springu, skutočná sila zabezpečenia Springu spočíva v tom, ako ľahko sa dá rozšíriť, aby vyhovovala vlastným požiadavkám [3].

Vlastnosti:

- Komplexná a rozširiteľná podpora pre autentifikáciu aj autorizáciu
- Ochrana proti útokom, ako je fixácia relácií, clickjacking, falšovanie požiadaviek medzi stránkami atď
- Integrácia Servlet API
- Voliteľná integrácia s Spring Web MVC

# Zoznam použitej literatúry

1. *What is AMQP Protocol All you need to know* [online]. www.wallarm.com, [b.r.]. [cit. 2022-11-24]. Dostupné z : <https://www.wallarm.com/what/what-is-amqp>.
2. *Spring AMQP*. spring.io, [b.r.]. Dostupné tiež z: <https://spring.io/projects/spring-amqp>.
3. *Spring Projects*. Spring.io, 2019. Dostupné tiež z: <https://spring.io/projects/spring-security>.