

## Проектная работа: News Aggregator

## Описание проекта

Приложение представляет собой микросервисное решение для агрегирования новостей из различных источников, включая курсы криптовалют с Binance.

# Используемые технологии

- **Java 21**
- **Spring Boot 3.5**
- **Spring Cloud 2025.0.0**
- **Lombok, Swagger, Webflux, WebSocketClient**
- **ActiveMQ, OpenFeign, Thymeleaf**
- **Keycloak (OpenID)**
- **MongoDB**
- **Docker, Kubernetes**

# Взаимодействие с внешними системами

- Получения новостей из Google Ok Surf выполняется с использованием архитектурный стиль REST.
- Получение актуальных курсов криптовалют из Binance выполняется по Websocket.

# Межсервисное взаимодействие

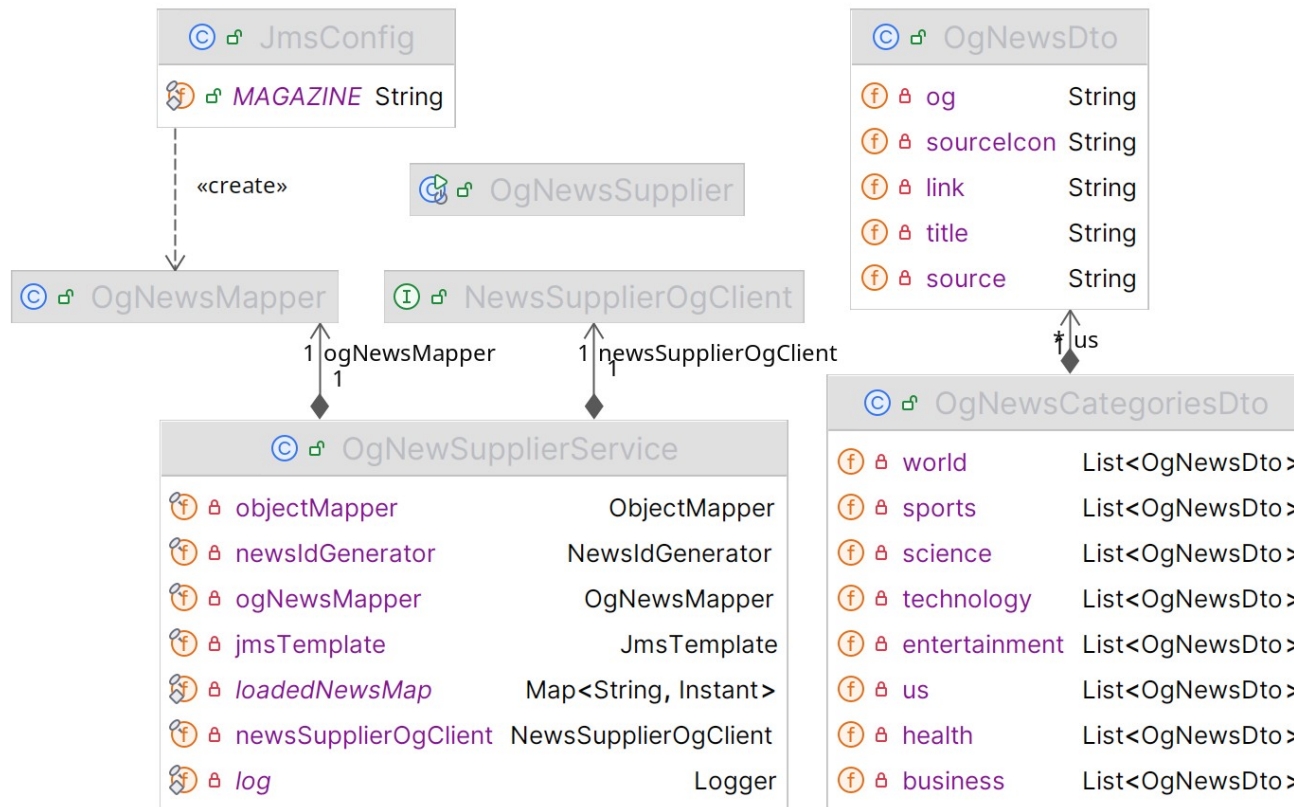
- Между новостными сервисами и агрегатором осуществляется через брокер сообщений ActiveMQ.
- Между поставщиком курсов Binance и фронтэндом осуществляется через STOMP (Simple Text Oriented Messaging Protocol).
- Между агрегатором и фронтендом осуществляется с использованием REST.

# Структура проекта

Состоит из следующих модулей:

- **og-news-supplier**
- **news-aggregator-server**
- **news-aggregator-frontend**
- **common-news**
- **binance-cryptocurrency-rate**

# Структура модуля og-news-supplier



# Структура модуля news-aggregator-server





# Структура модуля binance-cryptocurrency-rate

BinanceCryptocurrencyRunner
<code>main(String[])</code> void

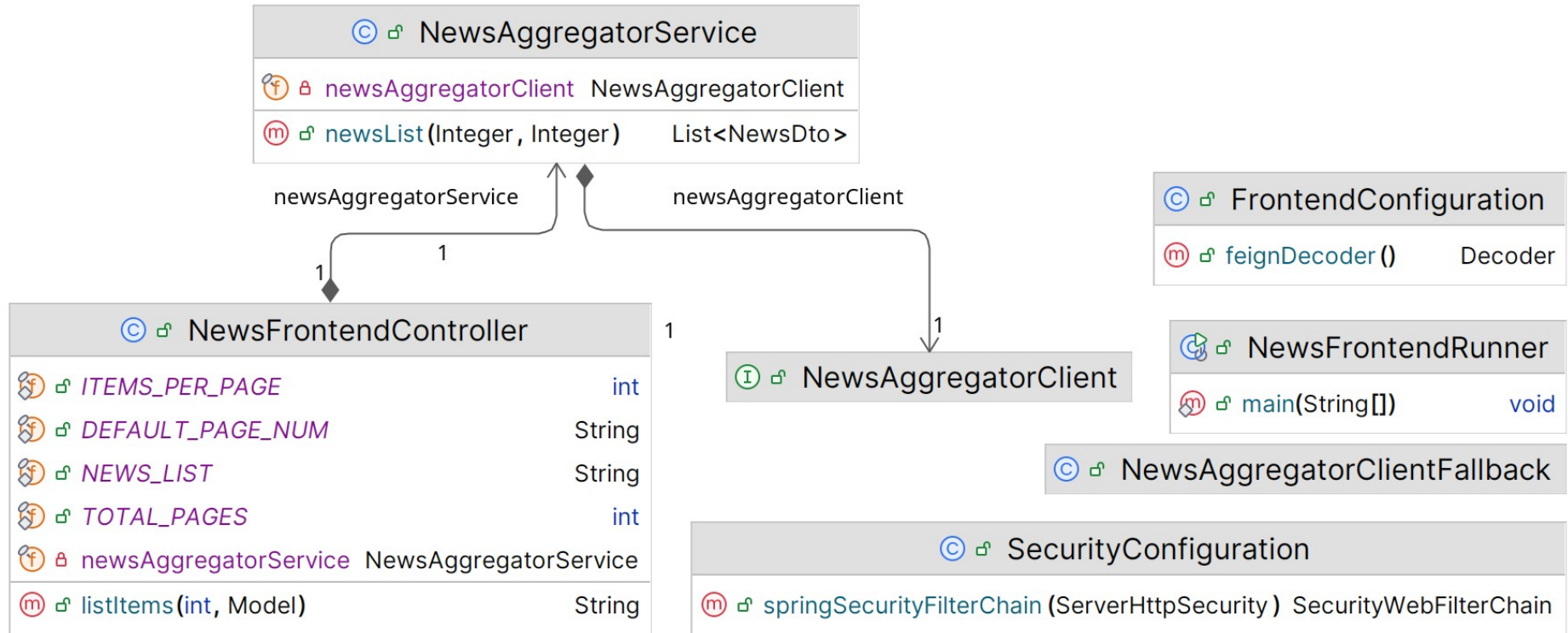
BinanceConfig
<code>reactorNettyWebSocketClient ()</code> ReactorNettyWebSocketClient
<code>objectMapper ()</code> ObjectMapper

WebSocketConfig
<code>registerStompEndpoints (StompEndpointRegistry)</code> void
<code>configureMessageBroker (MessageBrokerRegistry)</code> void

BinanceService
<code>client</code> ReactorNettyWebSocketClient
<code>BINANCE_WS_URL</code> String
<code>objectMapper</code> ObjectMapper
<code>connected</code> AtomicBoolean
<code>btcToUsdtBits</code> AtomicLong
<code>log</code> Logger
<code>messagingTemplate</code> SimpMessagingTemplate
<code>reconnectIfNeeded ()</code> void
<code>handleMessage (String)</code> void
<code>sendPriceUpdate ()</code> void
<code>init ()</code> void
<code>connect ()</code> void

CryptoRateDto
<code>symbol</code> String
<code>price</code> double
<code>timestamp</code> long
<code>canEqual (Object)</code> boolean
<code>getSymbol ()</code> String
<code>getPrice ()</code> double
<code>getTimestamp ()</code> long
<code>setSymbol (String)</code> void
<code>setPrice (double)</code> void
<code>setTimestamp (long)</code> void
<code>equals (Object)</code> boolean
<code>hashCode ()</code> int
<code>toString ()</code> String

# Структура модуля news-aggregator-frontend



# Вывод

**Проект демонстрирует реализацию масштабируемого микросервисного приложения с использованием современных технологий и инструментов.**