



Построение диаграмм микросервисов по исходному коду в IntelliJ IDEA

Автор: Александр Андреевич Башкиров, 17.Б11-мм
Научный руководитель: к.ф.-м.н, доцент, Д.В. Луцив
Рецензент: программист ООО “ИнтеллиДжей Лабс”,
к.т.н. Н.Н. Митропольский

Санкт-Петербургский государственный университет
Кафедра системного программирования

14 мая 2021 г.

Введение. Диаграммы микросервисов

eShopOnContainers reference application (Development environment architecture)

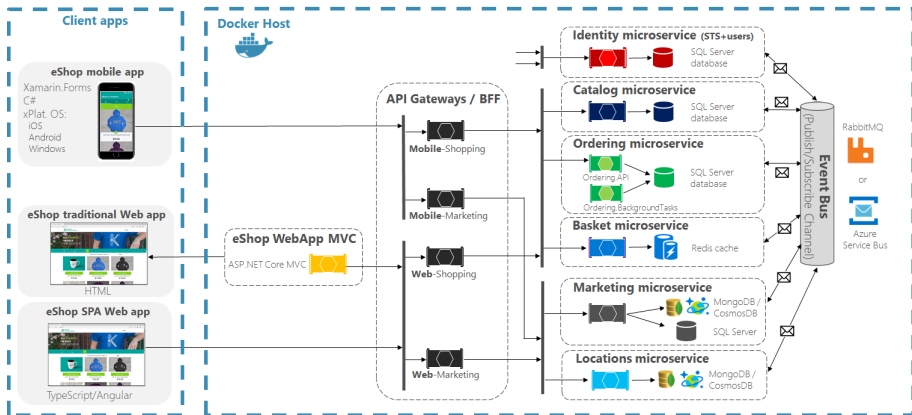


Рис. 1: Микросервисная архитектура эталонного интернет-магазина от .NET Architecture [5]

Введение. Диаграммы микросервисов

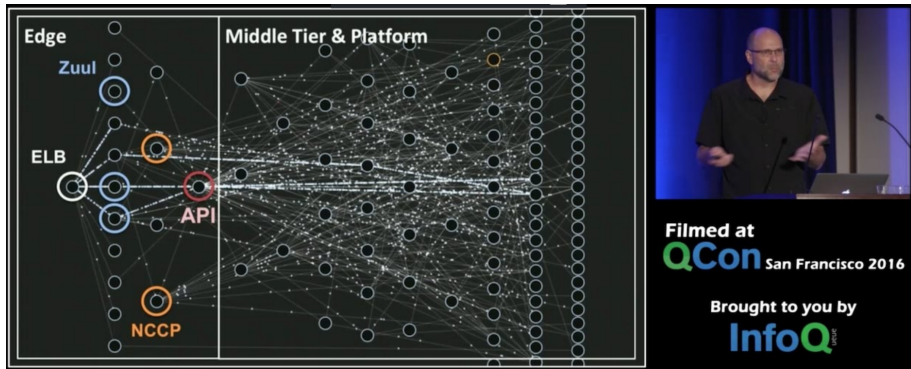


Рис. 2: Микросервисная архитектура Netflix на 2016 год [1]

Поддержка микросервисных фреймворков в IntelliJ IDEA

В IntelliJ IDEA уже реализована функциональность

- поиска в исходном коде микросервисов объявлений *эндпоинтов*;
- обращений к эндпоинтам;
- взаимодействий с очередями сообщений.

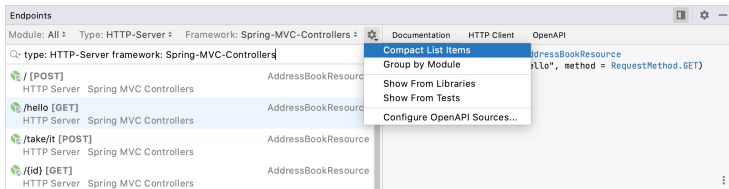


Рис. 3: Пример окна просмотра эндпоинтов



Введение. Средства визуализации в IntelliJ IDEA [4]

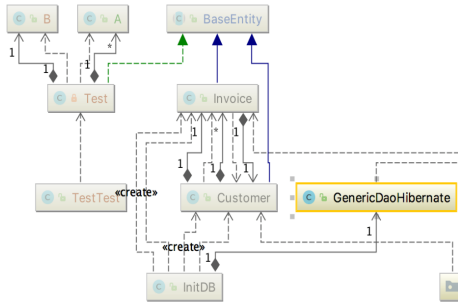


Рис. 4: UML диаграммы классов в IntelliJ IDEA [3]

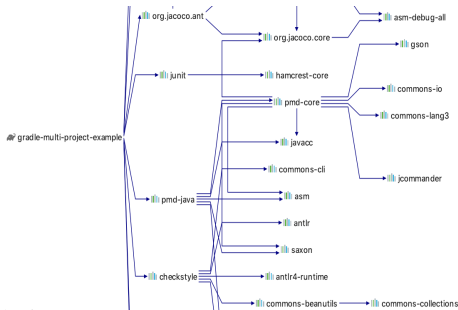


Рис. 5: Диаграммы Gradle¹ зависимостей в IntelliJ IDEA [2]

¹<https://gradle.org/>

Постановка задачи

Целью работы является расширение проприетарного плагина Endpoints для IntelliJ IDEA для предоставления функциональности генерации диаграмм микросервисов для исходных кодов на языках Java и Kotlin, использующих фреймворки [Spring Boot](#) и [Micronaut](#). Для её выполнения были поставлены следующие задачи:

- Изучить предметную область и существующие решения.
- Сформулировать требования и ограничения реализации.
- Спроектировать архитектуру реализовываемого компонента.
- Реализовать функциональность генерации диаграмм микросервисов на основании данных исходного кода, отображающих сервисы и возможное взаимодействие между ними посредством
 - ▶ HTTP запросов,
 - ▶ WebSocket протокола,
 - ▶ очередей сообщений.
- Выполнить апробацию на проектах с открытым исходным кодом, использующих микросервисную архитектуру.

Существующие решения



Рис. 6: Классификация средств визуализации микросервисов в соответствии с предъявляемыми требованиями и уровнем детализации

Существующие решения

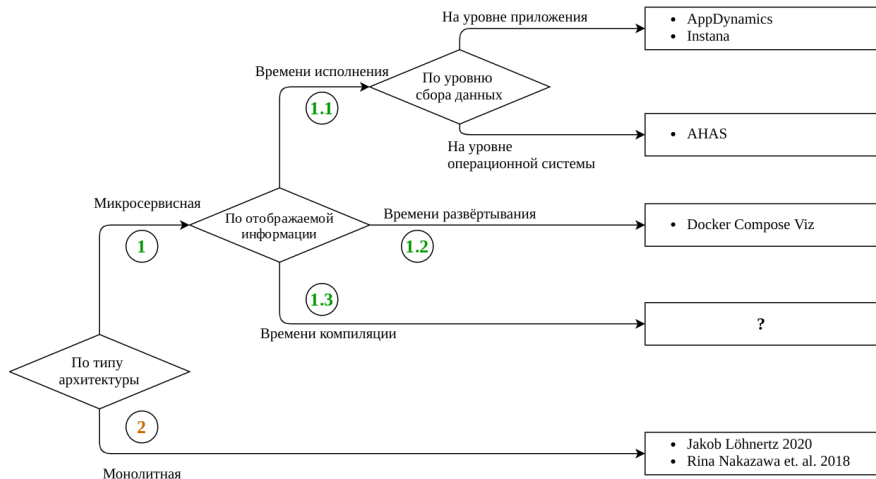


Рис. 7: Классификация средств визуализации микросервисов в соответствии со сферами применимости

Особенности данных времени компиляции

```
1  @RestController
2  @RequestMapping("/greetings")           // <-- Path declaration
3  public class SpringRestController {
4      @GetMapping(value =("/{name}"), // <-- HTTP Mapping declaration
5                  produces = MediaType.TEXT_PLAIN_VALUE)
6      public ResponseEntity<?> greeting(@PathVariable String name) {
7          return new ResponseEntity<>("Hello " + name, HttpStatus.OK);
8      }
9  }
```

Листинг 1: Пример простейшего сервиса в терминах Spring REST Controller

```
1  public class Application {
2      public static void main(String args[]) {
3          final var greeting = new RestTemplate()//----- <-- Request
4              .getForObject("http://some.domain.name/greetings/{name}",
5                           String.class, "Alexander");
6          System.out.println(greeting);
7      }
8  }
```

Листинг 2: Пример запроса к микросервису на листинге 1 в терминах Spring REST Client

Архитектура проекта

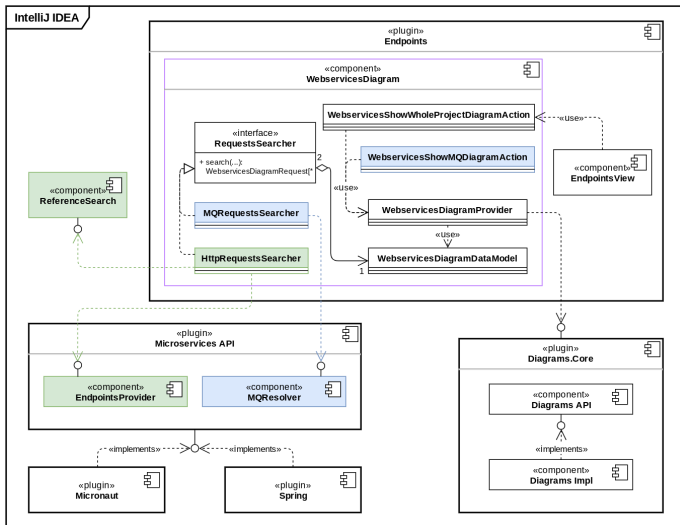


Рис. 8: UML-диаграмма компонентов реализации

Процесс генерации диаграммы

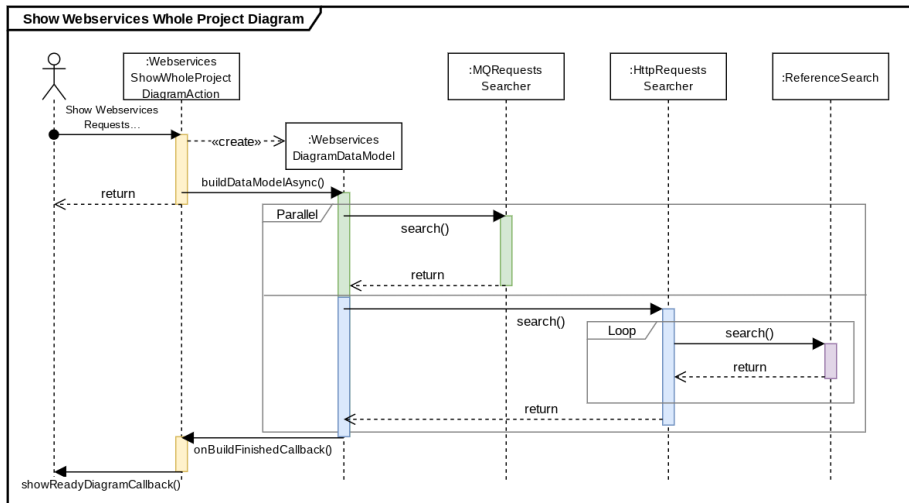
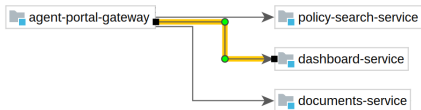


Рис. 9: Процесс генерации диаграммы взаимодействия микросервисов

Навигация по проекту с помощью диаграммы



Powered by yFiles

Find: Edge Origins x








Edge Origins 3 results

- agent-portal-gateway 3 results
 - pl.altkom.asc.lab.micronaut.poc.gateway.client.v1 3 results
 - DashboardGatewayClient 3 results
 - queryTotalSales(GetTotalSalesQuery) 1 result
 - 22 @Post("/totalsales")
 - querySalesTrends(GetSalesTrendsQuery) 1 result
 - queryAgentsSales(GetAgentsSalesQuery) 1 result

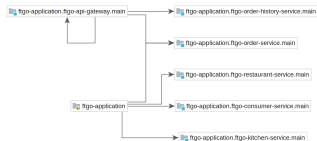
```
16 @Client(id = "dashboard-service", path = "/dashboard")
17 @Retryable(attempts = "2", delay = "2s")
18 public interface DashboardGatewayClient
19     extends DashboardOperations {
20
21     @Override
22     @Post("/totalsales")
23     GetTotalSalesQueryResult queryTotalSales(
24         @Body GetTotalSalesQuery query);
25
26     @Override
27     @Post("/trends")
28     GetSalesTrendsQueryResult querySalesTrends(
29         @Body GetSalesTrendsQuery query);
```

Рис. 10: Пример использования окна поиска для представления взаимодействий микросервисов в исходном коде

Апробация

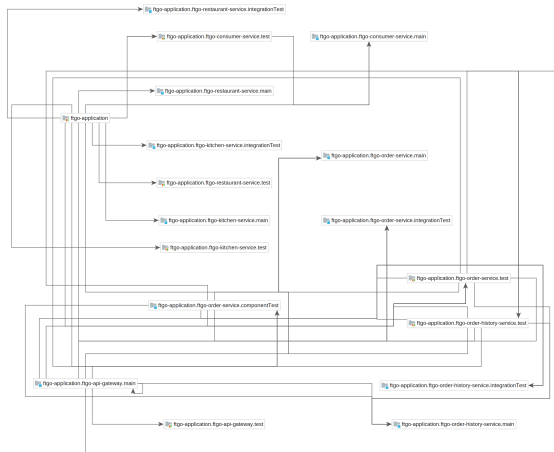
Проект	Используемые технологии	Число сервисов
Train Ticket 	Java, Spring	> 50
FTGO 	Java, JavaScript, Spring	6
Apollo 	Java, JavaScript, Spring	6
Genie 	Java, Spring	4
Asc-Lab POC 	Java, Micronaut, Kafka	10
Piomin Kafka 	Java, Micronaut, Kafka	4
Jexp Meetup 	Java, Micronaut, Kafka	4

Апробация. FTGO Application



Powered by yFiles

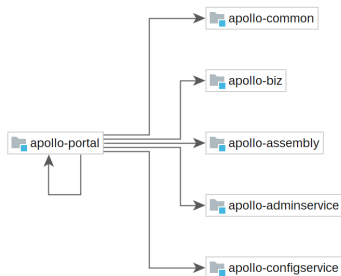
(a) Продуктовые модули



Powered by yFiles

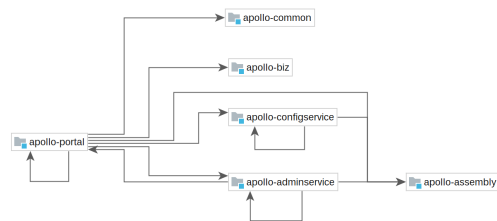
(b) С учетом модулей с тестами

Апробация. Apollo



Powered by yFiles

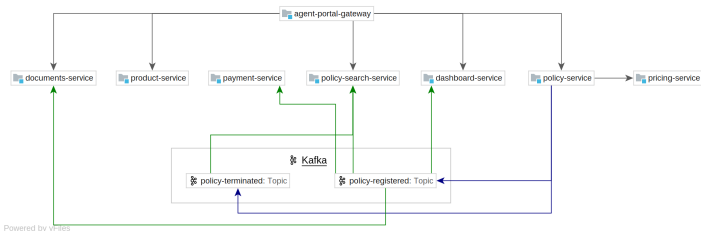
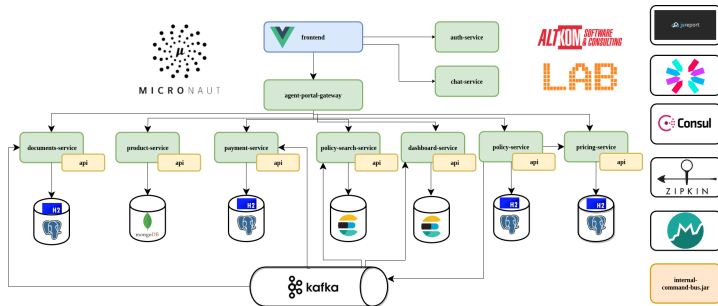
(a) Продуктовые модули



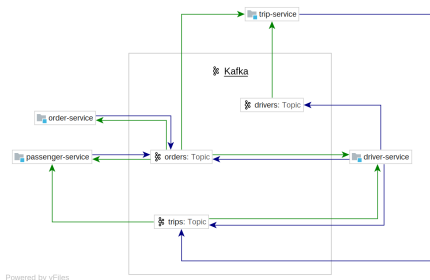
Powered by yFiles

(b) С учетом модулей с тестами

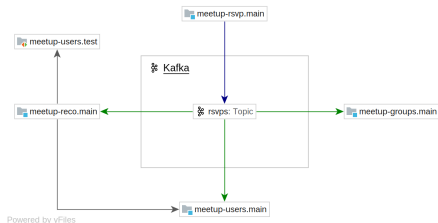
Апробация. Asc-Lab POC



Апробация. Piomin Kafka и Jexp Meetups



(a) Piomin Kafka



(b) Jexp Kafka

Возможности дальнейшего развития

- Поддержка визуализации RPC-фреймворков, в частности gRPC.
- Поддержка поиска конфигураций баз данных на основании сведений фреймворка.
- Добавление к визуализации данных времени развёртывания, к примеру, из файлов Docker Compose, Kubernetes, OpenAPI и др.
- Реализация плагинов поддержки микросервисных фреймворков для иных языков программирования
- Интеграция с UML-диаграммами классов.
- Интеграция с системой непрерывной серверной аналитики исходного кода Qodana².

²<https://www.jetbrains.com/help/qodana/getting-started.html>

- Сделан обзор предметной области.
- Сформулированы требования и ограничения к целевой функциональности.
- Спроектирована архитектура целевой функциональности.
- Реализована функциональность генерации диаграмм микросервисов как часть Endpoints плагина. Поддерживается отображение взаимодействий
 - ▶ с помощью HTTP/Websocket протоколов,
 - ▶ посредством очереди сообщений.
- Выполнена апробация на ряде крупнейших микросервисных проектов с открытым исходным кодом:
 - ▶ подсистема анализа микросервисных фреймворков справляется с поставленными задачами;
 - ▶ подсистема отображения диаграмм оказалась несостоятельной.

Дополнительные результаты

- Исправление ряда ошибок и проблем использования в плагине отрисовки диаграмм (исправления вошли в выпуск IntelliJ IDEA 2021.1³):
 - ▶ утечка памяти при открытом обзорном окне;
 - ▶ потеря качества при масштабировании иконок ([IDEA-256152](#));
 - ▶ артефакты отрисовки ([IDEA-257863](#));
 - ▶ ошибки навигации;
 - ▶ неприемлимая производительность отрисовки при включенной фоновой (background) сетке ([IDEA-98769](#));
 - ▶ артефакты и подвисания при перетаскивании узлов;
 - ▶ некорректное перекрытие узлов и рёбер;
 - ▶ некорректно отображаемая группировка рёбер;
 - ▶ не работающая система Undo/Redo ([IDEA-211984](#)).

³<https://www.jetbrains.com/idea/whatsnew/#whats-new-20211>

Дополнительные результаты

- Улучшена интеграция с библиотекой yFiles [6], в частности добавлены:
 - ▶ 18 новых алгоритмов размещения (layout);
 - ▶ отрисовка направляющих линий (snapping);
 - ▶ отрисовка мостов при перекрытии рёбер;
 - ▶ склейка (merge) рёбер одинакового типа;
 - ▶ алгоритм автоматической укладки только рёбер.
- Добавлена⁴ поддержка генерации UML-диаграмм по языку Kotlin ([IDEA-251446](https://www.jetbrains.com/idea/whatsnew/#kotlin-x)). Функциональность UML-диаграмм классов готова к интеграции с высокоуровневыми диаграммами (взаимодействия микросервисов, модулей проекта, Spring компонентов).

⁴<https://www.jetbrains.com/idea/whatsnew/#kotlin-x>

Библиография и ссылки I



Evans Josh. Mastering Chaos - A Netflix Guide to Microservices. — 2016. —

Режим доступа: <https://www.infoq.com/presentations/netflix-chaos-microservices/> (дата обращения: 13.12.2020).



IntelliJ Idea Gradle dependencies. —




Режим доступа: <https://www.jetbrains.com/help/idea/work-with-gradle-dependency-diagram.html> (дата обращения: 13.12.2020).



IntelliJ Idea UML class diagrams. —

Режим доступа:
<https://www.jetbrains.com/help/idea/class-diagram.html>
(дата обращения: 13.12.2020).

Библиография и ссылки II

-  IntelliJ Idea Функциональная и эргономичная IDE для JVM. —
Режим доступа: <https://www.jetbrains.com/ru-ru/idea/> (дата обращения: 13.12.2020).
-  .NET Microservices Sample Reference Application. —
Режим доступа:
<https://github.com/dotnet-architecture/eShopOnContainers>
(дата обращения: 13.12.2020).
-  yFiles for Java – enables you to create impressive cross-platform Java diagramming applications. —
Режим доступа:
<https://www.yworks.com/products/yfiles-for-java-2.x> (дата обращения: 13.12.2020).