В память о моем отце, Иване.

– Игорь Лозинский

Mapt – это цифровая онлайн-библиотека, которая откроет вам полный доступ к более чем 5 тыс. книг и видеороликов, а также к передовым инструментам, которые помогут вам планировать свое личное развитие и продвигаться по карьерной лестнице. За более подробной информацией обращайтесь на наш вебсайт www.mapt.io.

Что дает подписка?

- Электронные книги и видеоуроки практической направленности, созданные более чем 4 тыс. профессионалами в своем деле, помогут вам меньше времени тратить на обучение и больше на программирование.
- Воспользовавшись инструментом Skill Plans, вы сможете составить индивидуальный план своего обучения.
- Станете получать доступ к новым электронным книгам и видеоурокам каждый месяц.
- У вас будет возможность полнотекстового поиска в библиотеке Mapt.
- Вы сможете копировать и распечатывать содержимое книг, а также оставлять закладки.

www.PacktPub.com

Знаете ли вы, что издательство Packt предлагает электронные версии всех выпускаемых им книг в форматах PDF и ePub? Приобрести электронные версии можно на сайте www.PacktPub.com, а при покупке печатной книги вы получите скидку на ее электронную копию. За дополнительной информацией обращайтесь по адресу: customercare@packtpub.com.

На сайте www.PacktPub.com можно также найти множество технических статей, подписаться на бесплатные новостные письма и получить исключительные скидки на печатные и электронные книги издательства Packt.

Предисловие

Наконец реактивное программирование получило заслуженную поддержку в таких известных в мире Java продуктах, как Spring Boot и Spring Framework. Как бы вы охарактеризовали решения Spring? Многие пользователи обычно отвечают на этот вопрос коротко: прагматичность. Предлагаемая поддержка реактивного программирования не является исключением, и команда решила продолжать оказывать содействие как реактивным, так и нереактивным стекам. С выбором приходит ответственность, поэтому важно понимать, когда лучше использовать парадигму «реактивного программирования» и какие передовые методы можно применить при создании своей следующей промышленной системы.

Проект Spring позиционируется как поставщик лучших инструментов для разработки любых видов микросервисов. Поддержка реактивного стека Spring помогает разработчикам создавать невероятно эффективные, высокодоступные и надежные конечные точки. Кроме того, реактивные микросервисы на основе Spring терпимо относятся к задержкам в Сети и в меньшей степени подвержены разрушительному влиянию отказов. Только представьте – это решение можно использовать и для Edge API, и для серверной части мобильного приложения, и для тесно взаимосвязанных микросервисов! Не знали? Реактивные микросервисы изолируют медленные транзакции и вознаграждают быстрые.

Как только вы определитесь с потребностями, Project Reactor станет для вас реактивной основой, естественным образом сочетающейся с вашим реактивным проектом на основе Spring. В последних выпусках 3.x этой библиотеки было реализовано большинство реактивных расширений, описанных компанией Microsoft в 2011 году. Наряду со стандартным словарем библиотека Reactor добавляет великолепную поддержку механизма Reactive Streams управления потоком на каждом функциональном шаге и уникальные возможности, такие как передача контекста.

В своей книге Олег и Игорь на примерах пусть и искусственных, но не упрощенных, описывают фантастическое путешествие по миру реактивного программирования и реактивных систем. После краткого введения в библиотеку Project Reactor и историю ее развития вы быстро погрузитесь в исследование практических примеров на основе Spring Boot 2. Кроме того, авторы не упустили случая всерьез охватить тему тестирования и четко и ясно показать, как создается качественный реактивный код.

Авторы предлагают читателям исчерпывающее введение в шаблоны реактивного проектирования для эффективного масштабирования, отвечающего настоящим и будущим потребностям. Они не просто описывают приемы реактивного программирования, но также дают массу практических советов по использованию Spring Boot и Spring Framework. В главе, посвященной перспективам реактивного программирования, авторы разжигают любопытство читателя, знакомя его с некоторыми подробностями о реактивных взаимодействиях с использованием

RSocket – многообещающей технологии, призванной привнести преимущества реактивного программирования в транспортный уровень.

Надеюсь, вы получите от чтения книги столько же удовольствия, сколько и я, и продолжите исследовать новые способы разработки приложений.

Стефан Мальдини (Stéphane Maldini)
Ведущий разработчик, Project Reactor

Об авторах

Олег Докука (**Oleh Dokuka**) – опытный инженер-программист, обладатель награды Pivotal Champion и один из основных вкладчиков в развитие Project Reactor и Spring Framework. Он хорошо знает, как устроены оба фреймворка, и ежедневно популяризирует идеи реактивного программирования с использованием Project Reactor. Наряду с этим Олег использует Spring Framework и Project Reactor в разработке программного обеспечения, поэтому он не понаслышке знает, как создавать реактивные системы с применением этих технологий.

Игорь Лозинский (**Igor Lozynskyi**) – старший Java-разработчик, в основном создающий надежные, масштабируемые и невероятно быстрые системы. Имеет за плечами более чем семилетний опыт работы с платформой Java. Увлекается интересными и динамичными проектами как в своей жизни, так и в разработке программного обеспечения.

Об обозревателях

Николай Алименко (**Mikalai Alimenkou**) – старший менеджер по работе с корпоративными клиентами, технический лидер Java и опытный преподаватель. Эксперт в разработке на Java, в области масштабируемых архитектур, а также методов гибкой разработки, процессов обеспечения качества и управления проектами. Имеет более чем 14-летний опыт разработки программного обеспечения, специализируется на создании сложных, распределенных и масштабируемых систем и на глубоком преобразовании компаний. Активный участник многих международных конференций. Основатель, тренер и независимый консультант тренинг-центра XP Injection. Организатор и идеолог международных конференций Selenium Camp, JEEConf и XP Days Ukraine, а также клуба анонимных разработчиков Anonymous Developers Club (UADEVCLUB).

Назарий Черкас (**Nazarii Cherkas**) трудится архитектором решений в Hazelcast – в компании, разрабатывающей проекты с открытым исходным кодом, такие как Hazelcast IMDG и Hazelcast Jet. Имеет многолетний опыт деятельности на разных должностях – от инженера-программиста на Java до руководителя группы разработчиков. Принимал участие в различных проектах для множества отраслей – от телекоммуникации и здравоохранения до критических систем, обслуживающих инфраструктуру одной из крупнейших в мире авиакомпаний. Имеет степень магистра информатики Черновицкого национального университета имени Юрия Федьковича.

Томаш Нуркевич (**Tomasz Nurkiewicz**) – обладатель титула Java Champion. Потратил на программирование половину жизни. Трудится в сфере электронной коммерции. Занимается разработкой свободного программного обеспечения, является самым ценным блогером DZone и очень активно участвует в работе сайта Stack Overflow. Автор, тренер, докладчик, технический обозреватель и легкоатлет. Считает, что не бывает кода, который не тестируется автоматически. Написал книгу о RxJava.

Предисловие от издательства

Отзывы и пожелания

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны. Вы можете написать отзыв прямо на нашем сайте www.dmkpress.com, зайдя на страницу книги, и оставить комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу **dmkpress@gmail.com**, при этом напишите название книги в теме письма. Если есть тема, в которой вы квалифицированы, и вы заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу **http://dmkpress.com/authors/publish_book/** или напишите в издательство по адресу **dmkpress@gmail.com**.

Список опечаток

Хотя мы приняли все возможные меры для того, чтобы удостовериться в качестве наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг — возможно, ошибку в тексте или в коде, — мы будем очень благодарны, если вы сообщите нам о ней. Сделав это, вы избавите других читателей от расстройств и поможете нам улучшить последующие версии этой книги. Если вы найдете какие-либо ошибки в коде, пожалуйста, сообщите о них главному редактору по адресу **dmkpress@gmail.com**, и мы исправим это в следующих тиражах.

Нарушение авторских прав

Пиратство в интернете по-прежнему остается насущной проблемой. Издательства «ДМК Пресс» и Packt очень серьезно относятся к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконно выполненной копией любой нашей книги, пожалуйста, сообщите нам адрес копии или веб-сайта, чтобы мы могли применить санкции. Пожалуйста, свяжитесь с нами по адресу электронной почты **dmkpress@gmail.com** со ссылкой на подозрительные материалы. Мы высоко ценим любую помощь по защите наших авторов, помогающую нам предоставлять вам качественные материалы.

Оглавление

Вступление

Реактивные системы всегда откликаются, что и требуется большинству предприятий. Разработка таких систем – сложная задача, требующая глубокого понимания предмета. К счастью, разработчики Spring Framework создали новую, реактивную версию проекта.

В книге «Практика реактивного программирования в Spring 5» вы познакомитесь с увлекательным процессом разработки реактивных систем на основе фреймворка Spring Framework 5.

Книга начинается со знакомства с основами реактивного программирования в Spring. Вы получите представление о возможностях фреймворка и узнаете об основах реактивного программирования. Далее вашему вниманию будут представлены методы реактивного программирования, способы их использования для организации взаимодействия баз данных и серверов. Все эти задачи продемонстрированы на примере реального проекта, что позволит вам попрактиковаться в применении полученных навыков.

А теперь просим всех на борт реактивной революции в Spring 5!

Кому адресована эта книга

Книга адресована разработчикам на Java, использующим фреймворк Spring для своих приложений и желающим получить возможность создавать надежные и реактивные приложения, которые можно масштабировать в облаке. Предполагается базовое знание распределенных систем и асинхронного программирования.

Содержание книги

Глава 1 *«Причины выбора Spring»* описывает случаи, для которых подходит реактивность. Здесь вы узнаете, чем реактивное решение лучше проактивного, познакомитесь с несколькими примерами кода, демонстрирующими разные способы связи между серверами, а также с потребностями и требованиями бизнеса к современному Spring Framework.

Глава 2 *«Реактивное программирование в Spring – основные понятия»* раскрывает потенциал реактивного программирования, знакомит с основными понятиями на примерах кода. Она демонстрирует мощь реактивного, асинхронного, неблокирующего программирования в Spring Framework на примерах кода и как оно применяется на практике. Здесь вы познакомитесь с моделью «издатель-подписчик», с реактивными событиями Flow и особенностями применения этих методов на практике.

Глава 3 *«Reactive Streams – новый стандарт потоков»* описывает проблемы реактивных расширений Reactive Extensions. На примерах кода раскрывается природа проблем, исследуются разные подходы к их решению. Глава также знакомит со спецификацией Reactive Streams, которая вводит новые компоненты в хорошо известную модель «издатель-подписчик».

Глава 4 *«Project Reactor – основа реактивных приложений»* рассматривает реализацию реактивной библиотеки, полностью осуществляющей спецификацию Reactive Streams. Сначала описываются преимущества Reactor, а затем рассматриваются причины, побудившие разработчиков Spring заняться созданием нового решения. Кроме того, глава знакомит с основами использования данной впечатляющей библиотеки – здесь вы получите представление о Mono и Flux, а также об особенностях применения реактивных типов.

Глава 5 *«Добавление реактивности с помощью Spring Boot 2»* знакомит с реактивными модулями, имеющимися в Spring, которые пригодятся при разработке реактивных приложений. Здесь вы узнаете, как начать использовать модули и как Spring Boot 2 помогает разработчикам быстро настраивать приложения.

Глава 6 *«Неблокирующи и асинхронные взаимодействия с WebFlux»* описывает один из основных модулей – Spring WebFlux, являющийся важным инструментом для организации асинхронного и неблокирующего взаимодействия пользователя и внешних служб. В ней дается обзор преимуществ этого модуля и сравнение его с Spring MVC.

Глава 7 *«Реактивный доступ к базам данных»* описывает модель реактивного программирования доступа к данным на основе Spring 5. Основное внимание уделяется поддержке реактивности в модулях Spring Data, исследуются средства, входящие в состав Spring 5, Reactive Streams и Project Reactor. Здесь представлен код, демонстрирующий реактивный подход к взаимодействию разных баз данных, таких как SQL и NoSQL.

Глава 8 *«Масштабирование с Cloud Streams»* познакомит с реактивными возможностями Spring Cloud Streams. Для начала вашему вниманию будет предложен обзор проблем и недостатков, с которыми можно столкнуться при организации масштабирования на разных серверах. Глава раскроет возможности решения Spring Cloud и продемонстрирует его реализацию на примерах кода с соответствующей конфигурацией Spring Boot 2.

Глава 9 *«Тестирование реактивных приложений»* описывает основы тестирования реактивных приложений. Вы познакомитесь с модулями Spring 5 Test и Project Reactor Test, узнаете, как управлять частотой следования событий, перемещать временные интервалы, расширять пулы потоков выполнения, проверять результаты и переданные сообщения.

Глава 10 *«И наконец, выпуск!»* содержит подробное пошаговое руководство по развертыванию и мониторингу решения. Наглядно показано, как осуществляется мониторинг реактивных микросервисов с использованием модулей Spring 5. Также в этой главе рассматриваются инструменты, которые пригодятся для сбора данных мониторинга и их отображения.

Что потребуется для работы с книгой

Разработка реактивных систем – сложная задача, требующая глубокого понимания предметной области, поэтому вам потребуется познакомиться с распределенными системами и асинхронным программированием.

Загрузка исходного кода примеров

Загрузить файлы с дополнительной информацией для книг издательства «ДМК Пресс» можно на сайте www.dmkpress.com или www.дмк.рф в разделе *«Читателям – Файлы к книгам»*.

Кроме того, примеры кода к книге доступны на сайте GitHub: https://github.com/PacktPublishing/Hands-On-Reactive-Programming-in-Spring-5.

Загрузка цветных иллюстраций

Мы также подготовили документ PDF, содержащий цветные иллюстрации со скриншотами и диаграммами, используемыми в книге. Его можно взять здесь: https://www.packtpub.com/sites/default/files/downloads/9781787284951_ColorImages.pdf.

Типографские соглашения

В книге используется несколько разных стилей оформления текста с целью обеспечить визуальное отличие информации разных типов. Ниже приводится несколько примеров стилей оформления и краткое описание их назначения.

Программный код в тексте, имена таблиц баз данных, имена папок, имена файлов, расширения файлов, пути в файловой системе, адреса URL, ввод пользователя и ссылки в Twitter оформляются так, как показано в следующем предложении: «При первом обращении будет вызван обработчик onSubscribe(), который сохранит подписку Subscription локально и известит издателя Publisher о готовности подписчика принимать события через метод request()».

Блоки программного кода оформляются следующим образом:

```
@Override
public long maxElementsFromPublisher() {
   return 1;
}
```

Ввод или вывод в командной строке оформляются так:

```
./gradlew clean build
```

Новые термины и важные определения выделяются в тексте полужирным.

Так оформляются предупреждения и важные примечания.

Так оформляются советы и рекомендации.

Отзывы и пожелания

Мы всегда рады отзывам наших читателей. Расскажите, что вы думаете об этой книге – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезными.

Вы можете написать отзыв прямо на нашем сайте `www.dmkpress.com`, зайдя на страницу книги и оставив комментарий в разделе *«Отзывы и рецензии»*. Также можно послать письмо главному редактору по адресу: `dmkpress@gmail.com`, при этом укажите название книги в теме письма.

Если есть тема, в которой у вас высокая квалификация, и вы заинтересованы в написании новой книги, заполните форму на нашем сайте `http://dmkpress.com/authors/publish_book/` или напишите в издательство по адресу: `dmkpress@gmail.com`.

Список опечаток

Хотя мы приняли все возможные меры для того, чтобы удостовериться в качестве наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг (в тексте или в коде), мы будем благодарны, если вы сообщите нам о ней. Этим вы поможете улучшить последующие версии книги.

Если найдете ошибки в коде, пожалуйста, сообщите о них главному редактору по адресу: `dmkpress@gmail.com`, и мы исправим их в следующих изданиях.

Нарушение авторских прав

Пиратство в интернете по-прежнему насущная проблема. Издательства «ДМК Пресс» и Packt очень серьезно относятся к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконно выполненной копией любой нашей книги, пожалуйста, сообщите нам адрес копии или веб-сайта, чтобы мы могли принять меры.

Пожалуйста, свяжитесь с нами по адресу `dmkpress@gmail.com` и пришлите ссылки на подозрительные материалы.

Мы высоко ценим любую помощь по защите наших авторов, способствующую предоставлению качественных материалов.

Глава 1

Причины выбора Spring

В этой главе мы объясним понятие **реактивности** и расскажем, почему реактивные подходы лучше традиционных, для чего рассмотрим примеры, когда традиционные подходы терпят неудачу. Затем исследуем фундаментальные принципы построения надежных систем, которые в большинстве своем являются **реактивными системами**. Узнаем, каковы основные причины, объясняющие необходимость использования механизмов рассылки сообщений для организации взаимодействий между распределенными серверами. Мы покажем, в какие случаи реактивность вписывается как нельзя лучше, расскажем о приемах **реактивного программирования** для создания модульной реактивной системы. Кроме того, обсудим, почему команда разработчиков Spring Framework решила включить реактивный подход в ядро фреймворка **Spring Framework 5**. Прочитав главу, вы поймете важность реактивности и то, почему стоит перенести свои проекты в реактивный мир.

Здесь рассматриваются следующие темы:

- основные преимущества реактивности;
- основные принципы создания реактивных систем;
- случаи, когда реактивный дизайн подходит лучше всего;
- приемы программирования реактивных систем;
- причины включения поддержки реактивности в Spring Framework.

Основные преимущества реактивности

В наши дни стало модным использовать слово **реактивный** – оно такое волнующее и непонятное. Однако стоит ли продолжать популяризовать реактивность, даже после того как слово заняло почетное место на разнообразных международных конференциях? Если забьем в поиск слово «реактивный», то обнаружим, что чаще всего оно встречается в паре со словом «программирование», и вместе они обозначают модель программирования. Однако это не единственный смысл реак-

тивности. За данным словом стоят фундаментальные принципы проектирования, направленные на создание надежных систем. Чтобы понять ценность реактивности как важнейшего принципа проектирования, представим, что мы развиваем малое предприятие.

Допустим, наше малое предприятие – это интернет-магазин, продающий современные товары по привлекательным ценам. Как большинство владельцев подобных магазинов, мы также наняли разработчиков программного обеспечения, которые помогут нам справиться с проблемами. Мы выбрали традиционный подход к разработке программного обеспечения и создали магазин.

Каждый час наш онлайн-магазин обычно посещает 1 тыс. пользователей. Чтобы справиться с потоком покупателей, мы купили современный компьютер и запустили на нем веб-сервер Tomcat, настроив пул с 500 предварительно созданными потоками выполнения. Среднее время отклика на большинство запросов – около 250 мс. Простейшие расчеты показывают, что такая конфигурация позволит нам обслуживать до 2 тыс. запросов в секунду. Согласно статистике, вышеупомянутое число пользователей в среднем производит около 1 тыс. запросов в секунду. Следовательно, текущей производительности системы вполне достаточно для обслуживания средней нагрузки.

Итак, мы настроили приложение с неплохим запасом производительности. Более того, наш интернет-магазин работал вполне стабильно до… последней пятницы ноября, то есть до Черной пятницы.

Черная пятница – важный день и для покупателей, и для продавцов. Покупатели получают возможность купить товар со скидкой, а продавцы – получить дополнительную прибыль. Однако этот день характеризуется необычным наплывом клиентов, что может стать причиной сбоев в работе интернет-магазина.

И конечно же, мы потерпели сокрушительное фиаско! В какой-то момент нагрузка превысила все наши ожидания. В пуле не оказалось свободных потоков выполнения для обработки запросов. Сервер резервного копирования не справился с наплывом покупателей, время отклика возросло, периодически наблюдались сбои. Мы начали терять некоторые запросы, в результате наши клиенты, неудовлетворенные долгим обслуживанием, переметнулись к конкурентам.

В итоге мы потеряли большое число клиентов, остались без дополнительной прибыли, а рейтинг магазина рухнул. Все это стало результатом увеличения времени отклика в условиях возросшей нагрузки.

Но не волнуйтесь, это далеко не новая проблема. Даже такие гиганты, как Amazon и Walmart, сталкивались с ней и давно нашли способы ее решения. Но не будем спешить и пройдем тот же путь, каким следовали наши предшественники, чтобы понять основные принципы проектирования надежных систем и дать им общее определение.

Узнать больше о проблемах магазинов-гигантов можно на следующих сайтах:

- Amazon.com. Проблема с отключениями (`https://www.cnet.com/news/amazon-com-hit-with-outages/`);
- Amazon.com. Как отказы приводили к потерям до 66 240 долларов в минуту (`https://www.forbes.com/sites/kellyclay/2013/08/19/amazon-com-goes-down-loses-66240-per-minute/#3fd8db37495c`);
- Walmart. Провал в Черную пятницу: веб-сайт не справился с нагрузкой (`https://techcrunch.com/2011/11/25/walmart-black-friday/`).

Теперь главный вопрос: что с этим делать? Из примера выше следует, что приложение должно как-то реагировать на изменение нагрузки и доступности внешних служб. Иначе говоря, оно должно активно реагировать на любые изменения, которые могут ухудшить доступность системы и ее способность откликаться на запросы пользователей.

Один из путей к главной цели – увеличение **эластичности**. Под этим термином понимается способность сохранять отзывчивость при различной рабочей нагрузке, то есть пропускная способность системы должна автоматически увеличиваться с ростом числа пользователей и уменьшаться – со снижением спроса. Эта особенность улучшает отзывчивость системы, потому что в любой момент пропускная способность системы может вырасти и обеспечить приемлемое среднее время задержки.

Время задержки – важная характеристика отзывчивости. При отсутствии должной эластичности из-за роста нагрузки увеличится время задержки, которое напрямую влияет на отзывчивость системы.

Например, увеличить пропускную способность системы можно, расширяя вычислительные мощности или запуская дополнительные экземпляры. В результате возрастет отзывчивость системы. С другой стороны, если поток пользователей уменьшился, система в ответ должна снизить потребление ресурсов, сократив тем самым накладные расходы. Добиться желаемой эластичности можно путем масштабирования – горизонтального или вертикального. Однако масштабирование распределенной системы – сложная задача. Обычно ограничиваются узкими местами или точками синхронизации в системе. С теоретической и практической точек зрения эти проблемы объясняются законом Амдала и универсальной моделью масштабирования Гюнтера Нейла. Мы обсудим их позже – в `главе 6` «*Неблокирующие и асинхронные взаимодействия с WebFlux*».

Здесь под накладными расходами понимается стоимость развертывания новых экземпляров в облаке или дополнительное потребление электроэнергии в случае установки добавочных компьютеров.

Конец ознакомительного фрагмента.
Приобрести книгу можно
в интернет-магазине
«Электронный универс»
e-Univers.ru