Netcracker

Spring Framework Обзор возможностей

Regular meeting for get knowledge and tech in EDUC

Вопросы, которые мы разберём

- Что такое Spring Framework?
- Какие модули есть в Spring?
- Как создать Spring проект?
- Как запустить Spring проект?

Что такое Spring Framework?

Spring Framework – универсальная платформа с открытым исходным кодом, которая предоставляет облегченное решение по созданию готовых корпоративных приложений.

- Spring был впервые выпущен в **июне 2003** года и получил широкое распространение.
- Текущая версия Spring – **5**
- Сайт проекта: **https://spring.io/**

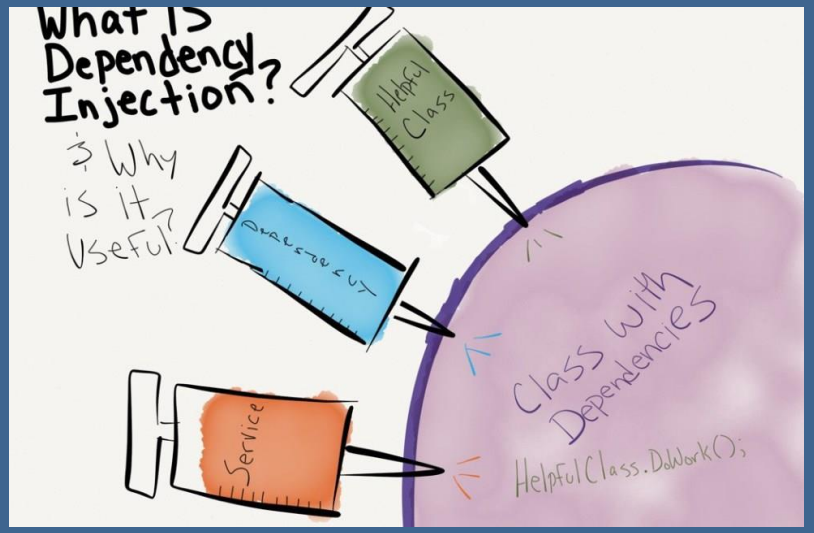
Какие модули есть в Spring?

Spring может быть рассмотрен как коллекция меньших фреймворков или фреймворков во фреймворке. Большинство этих фреймворков может работать независимо друг от друга, однако они обеспечивают большую функциональность при совместном их использовании. Эти фреймворки делятся на структурные элементы типовых комплексных приложений:

- **Inversion of Control-контейнер**: конфигурирование компонентов приложений и управление жизненным циклом Java-объектов.
- **Фреймворк доступа к данным**: работает с системами управления реляционными базами данных на Java-платформе, используя JDBC- и ORM-средства и обеспечивая решения задач, которые повторяются в большом числе Java-based environments.
- **Фреймворк управления транзакциями**: координация различных API управления транзакциями и инструментарий настраиваемого управления транзакциями для объектов Java.
- **Фреймворк MVC**: каркас, основанный на HTTP и сервлетах, предоставляющий множество возможностей для расширения и настройки (customization).
- **Фреймворк удалённого доступа**: конфигурируемая передача Java-объектов через сеть в стиле RPC, поддерживающая RMI, CORBA, HTTP-based протоколы, включая web-сервисы (SOAP).
- **Фреймворк аутентификации и авторизации**: конфигурируемый инструментарий процессов аутентификации и авторизации, поддерживающий много популярных и ставших индустриальными стандартами протоколов, инструментов, практик через дочерний проект Spring Security (ранее известный как Acegi).
- **Фреймворк удалённого управления**: конфигурируемое представление и управление Java-объектами для локальной или удалённой конфигурации с помощью JMX.
- **Фреймворк работы с сообщениями**: конфигурируемая регистрация объектов-слушателей сообщений для прозрачной обработки сообщений из очереди сообщений с помощью JMS, улучшенная отправка сообщений по стандарту JMS API.
- **Тестирование**: каркас, поддерживающий классы для написания модульных и интеграционных тестов.

Inversion of Control-контейнер

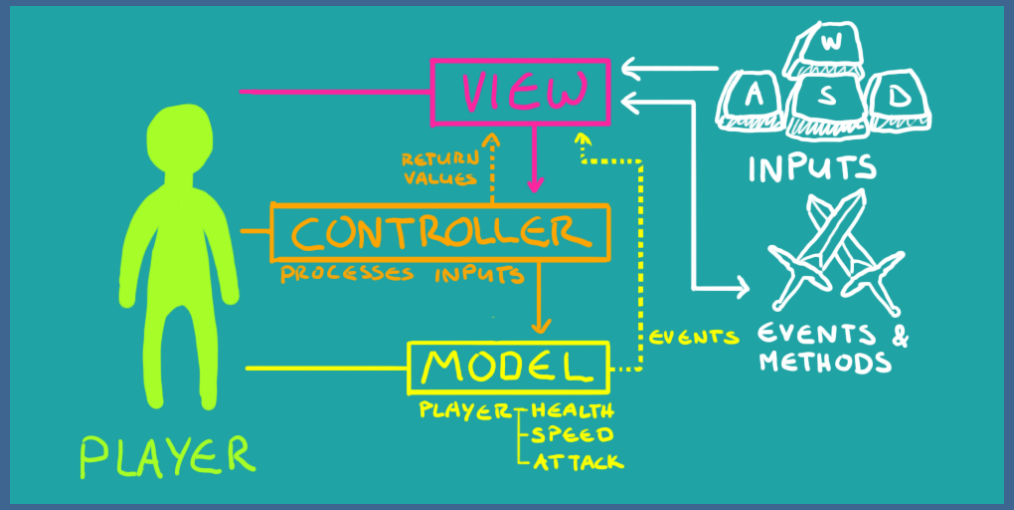
- Центральной частью Spring является контейнер **Inversion of Control**, который предоставляет средства конфигурирования и управления объектами Java. Контейнер отвечает за управление жизненным циклом объекта: создание объектов, вызов методов инициализации и конфигурирование объектов путём связывания их между собой.
- Объекты, создаваемые контейнером, также называются управляемыми объектами (**beans**). Обычно, конфигурирование контейнера, осуществляется путём внедрения **аннотаций** (начиная с 5 версии J2SE), но так же, есть возможность, по старинке, загрузить XML-файлы, содержащие определение bean'ов и предоставляющие информацию, необходимую для создания bean'ов.
- Объекты могут быть получены одним из двух способов:
- **Поиск зависимости** — шаблон проектирования, в котором вызывающий объект запрашивает у объекта-контейнера экземпляр объекта с определённым именем или определённого типа.
- **Внедрение зависимости** — шаблон проектирования, в котором контейнер передает экземпляры объектов по их имени другим объектам с помощью конструктора, свойства или фабричного метода.

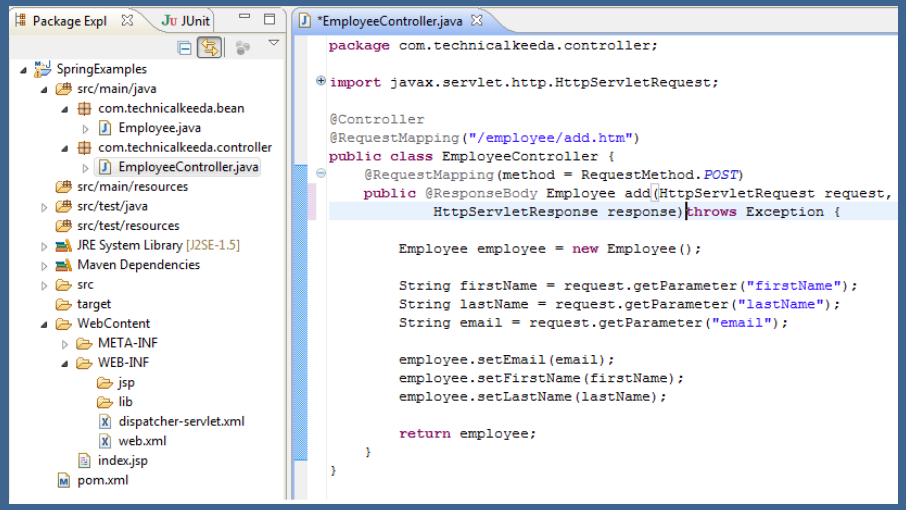



```
1  package com.in28minutes.login;
2
3  import org.springframework.beans.factory.annotation.Autowired;
9
10 @Controller
11 public class LoginController {
12
13     @Autowired
14     LoginService service;
15
16     @RequestMapping(value = "/login", method = RequestMethod.GET)
17     public String showLoginPage() {
18         return "login";
19     }
20
21     @RequestMapping(value = "/login", method = RequestMethod.POST)
```

MVC-платформа веб-приложений

- Spring MVC является фреймворком, ориентированным на запросы. В нем определены стратегические интерфейсы для всех функций современной запросно-ориентированной системы.
- Цель каждого интерфейса — быть простым и ясным, чтобы пользователям было легко его заново имплементировать, если они того пожелают. MVC прокладывает путь к более чистому front-end-коду. Все интерфейсы тесно связаны с Servlet API. Эта связь оставляет особенности Servlet API доступными для разработчиков, облегчая все же работу с ним.





Фреймворк доступа к данным

- Spring предоставляет свой слой доступа к базам данных посредством JDBC. Кроме того, он поддерживает все популярные ORM: Hibernate, JPA, JDO, EclipseLink, iBatis, Apache OJB, Apache Cayenne и т. п..
- Для всех этих фреймворков, Spring предоставляет такие особенности:
- **Управление ресурсами** — автоматическое получение и освобождение ресурсов базы данных
- **Обработка исключений** — перевод исключений при доступе к данным в исключения Spring-а
- **Транзакционность** — прозрачные транзакции в операциях с данными
- **Распаковка ресурсов** — получение объектов базы данных из пула соединений

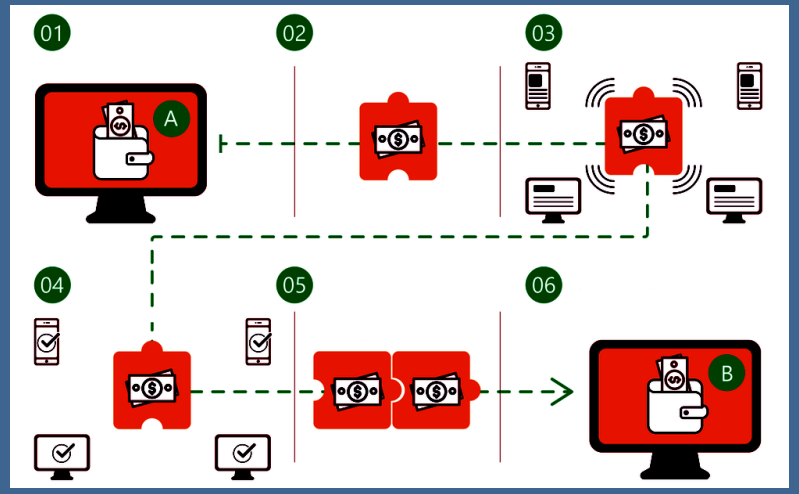
```
public void setDataSource(DataSource dataSource) {
    this.dataSource = dataSource;
    jdbcTemplate = new JdbcTemplate(this.dataSource);
}

@Override
public void insert(Book book) {
    String sql = "INSERT INTO BOOK (TITLE, DATE_RELEASE) VALUES (?,
    ?)";
    jdbcTemplate.update(sql, new Object[] { book.getTitle(),
                new java.sql.Date(book.getDateRelease().getTime())
    });
}
```

Управление транзакциями

- Фреймворк управления транзакциями в Spring привносит механизм абстракций для платформы Java. Основные возможности этих абстракций:

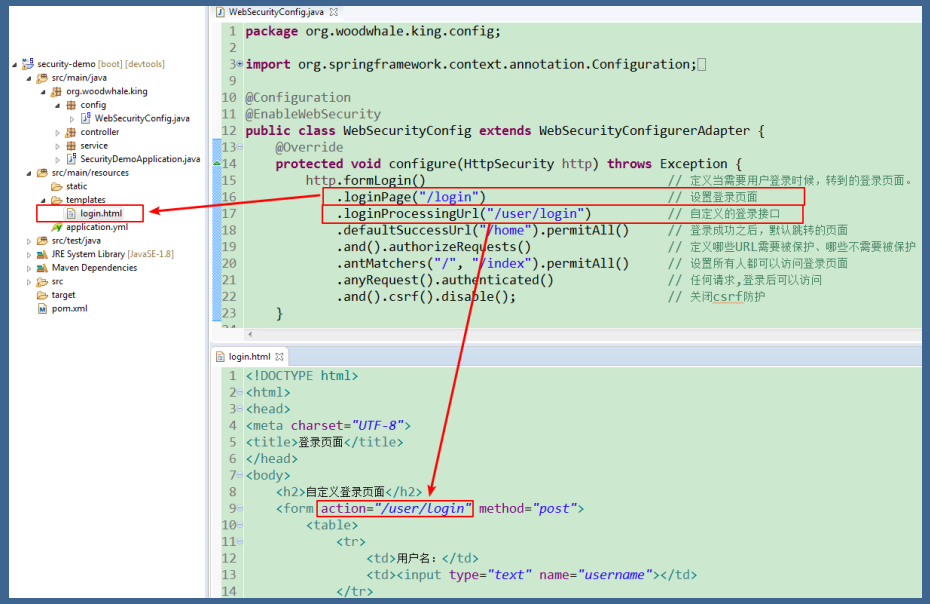
- Работа с локальными и глобальными транзакциями
- Работа с вложенными транзакциями
- Работа с точками сохранения в транзакциях




```
public void doSomething() {
  final String METHODNAME = "doSomething";
  logger.trace("entering " + CLASSNAME + "." + METHODNAME);
  TransactionStatus tx = transactionManager.getTransaction(
    new DefaultTransactionDefinition());
  try {
    // Business Logic
  } catch (RuntimeException ex) {
    logger.error("exception in "+CLASSNAME+"."+METHODNAME, ex);
    tx.setRollbackOnly();
    throw ex;
  } finally {
    transactionManager.commit(tx);
    logger.trace("exiting " + CLASSNAME + "." + METHODNAME);
  }
}
```

Аутентификация и авторизация

- Конфигурируемый инструментарий процессов аутентификации и авторизации, поддерживающий много популярных и ставших индустриальными стандартами протоколов, инструментов, практик через дочерний проект Spring Security.



Интеграция

- Поддерживает известные шаблоны Enterprise Integration через легкие сообщения и декларативные адаптеры:
- Endpoint; Channel (Point-to-point and Publish/Subscribe); Aggregator; Filter; Transformer; Control Bus;

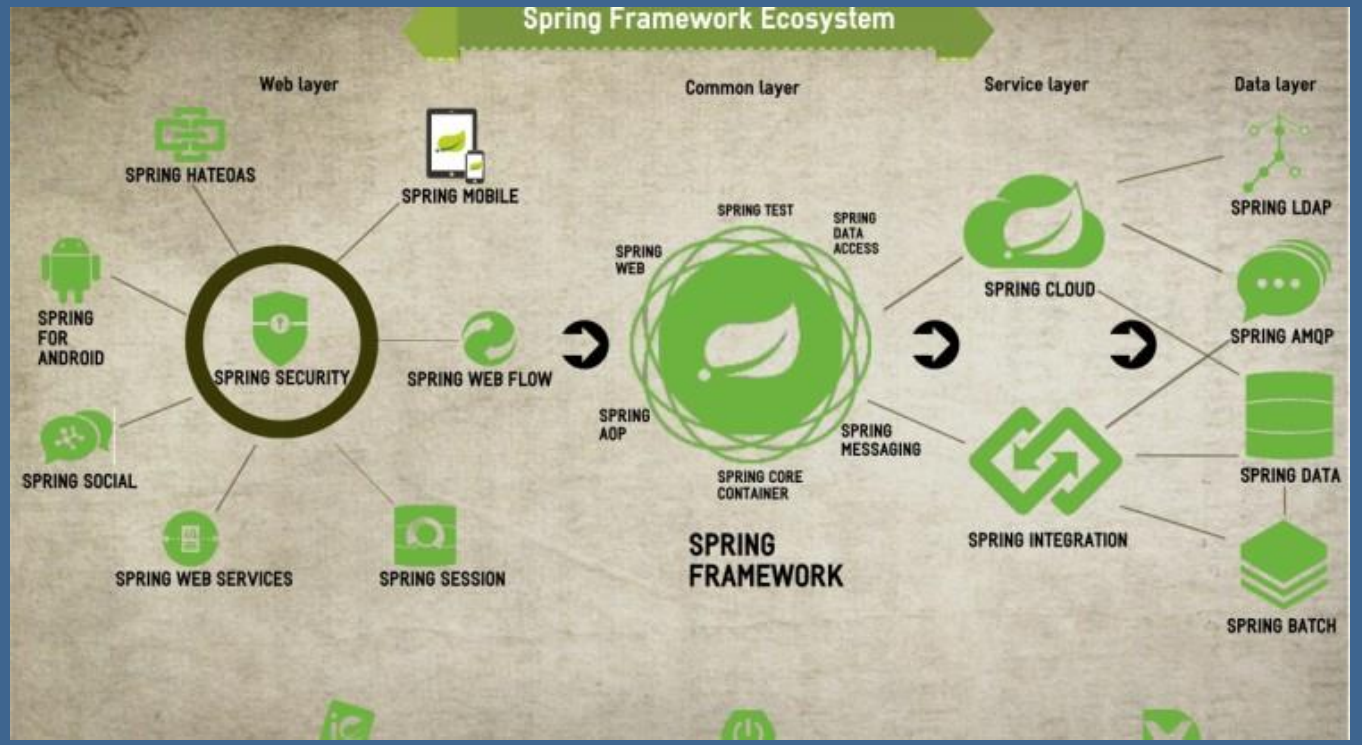
- Предоставляет функции, необходимые для отправки сообщений или для построения событийно-ориентированной архитектур:
- ReST/HTTP; FTP/SFTP; Twitter; WebServices (SOAP and ReST); TCP/UDP; JMS; RabbitMQ; Email; JMX…

```java
public class Main {

    public static void main(String... args) throws Exception {
        ApplicationContext ctx =
            new ClassPathXmlApplicationContext("context.xml");
        // Simple Service
        TempConverter converter =
            ctx.getBean("simpleGateway", TempConverter.class);
        System.out.println(converter.fahrenheitToCelcius(68.0f));
        // Web Service
        converter  = ctx.getBean("wsGateway", TempConverter.class);
        System.out.println(converter.fahrenheitToCelcius(68.0f));
    }
}
```

Основные проекты Spring

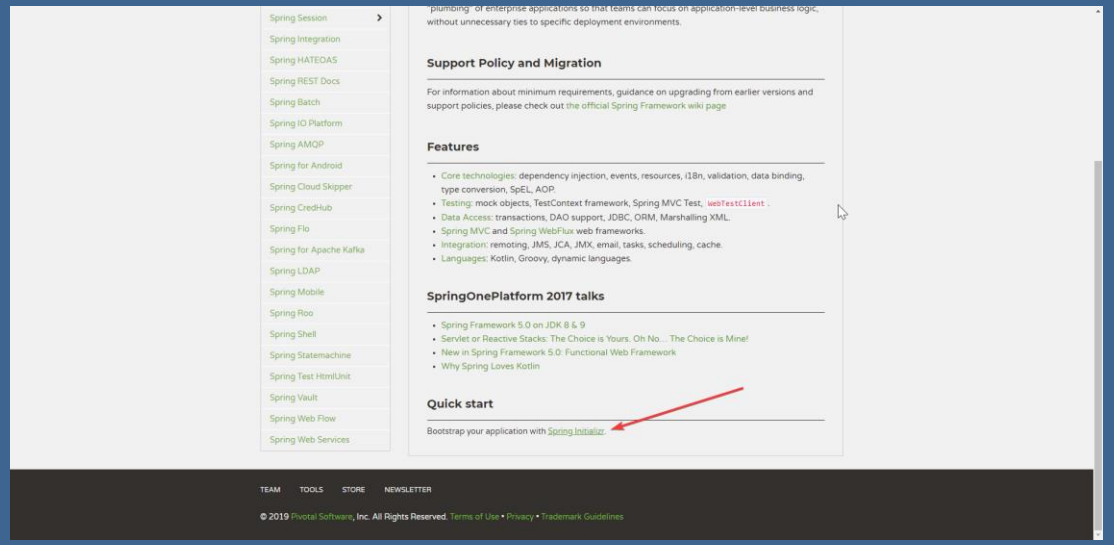
- От конфигурации до безопасности, от веб-приложений до больших данных - какими бы ни были потребности вашего приложения в инфраструктуре, есть Spring Project, который поможет вам создать его.
- Начните с малого и используйте только то, что вам нужно - Spring имеет модульную конструкцию.

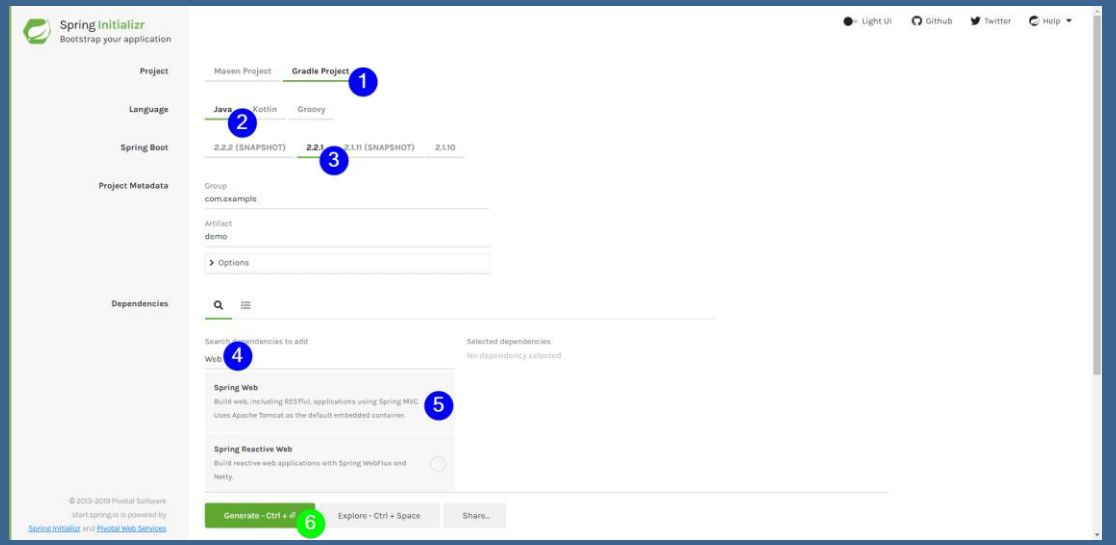


Как создать Spring проект?

Быстрый способ создать Spring проект – воспользоваться Spring Boot.

- Переходим по ссылке: https://spring.io/projects/spring-framework
- Выбираем начальную конфигурацию
- Нажимаем кнопку Generate

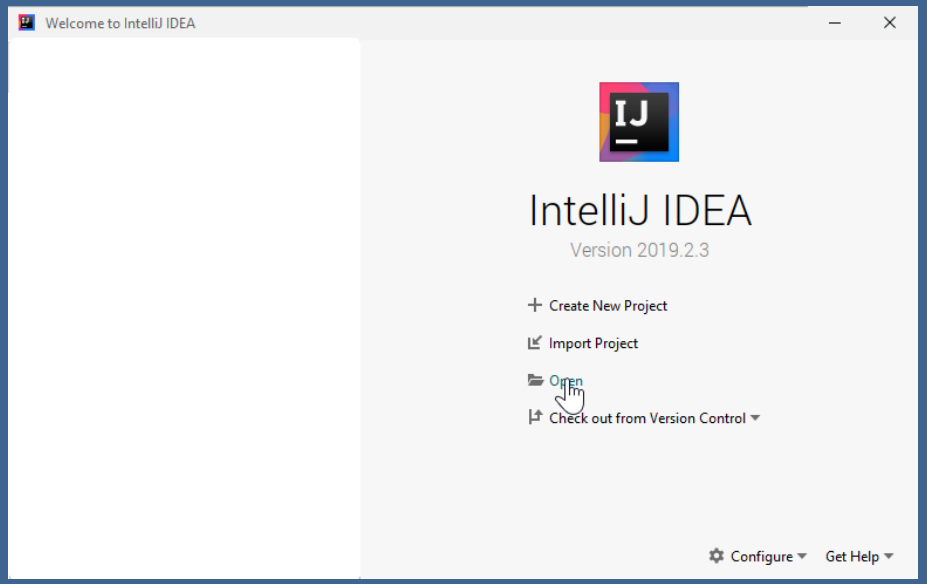


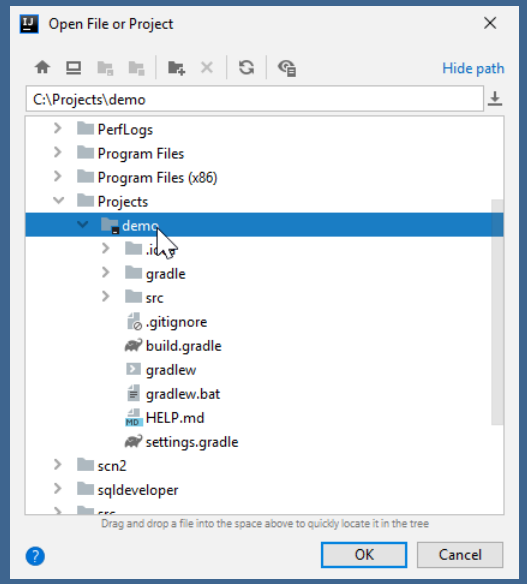


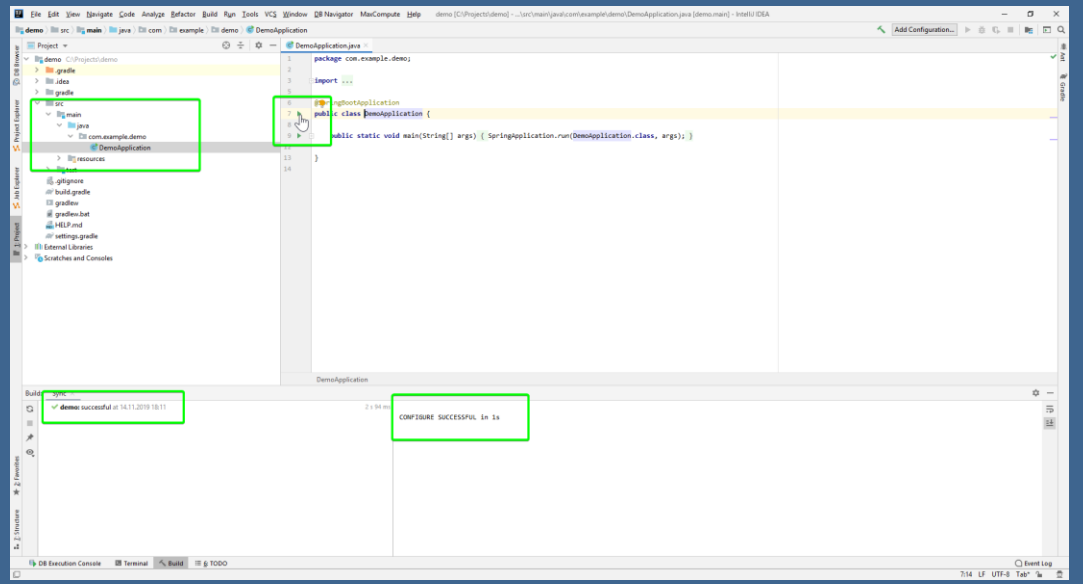
Как запустить Spring проект?

Это очень просто! Распаковываем проект и добавляем его в IntelliJ Idea.

Наше Spring приложение готово к работе!







Спасибо!

Дополнительную информацию можно почитать в интернете:

- https://spring.io/
- https://spring.io/projects
- https://spring-projects.ru/projects/spring-framework/
- https://javarush.ru/groups/posts/spring-framework-java-1
- https://ru.wikipedia.org/wiki/Spring_Framework