

LẬP TRÌNH JAVA NÂNG CAO

CHƯƠNG 3: Spring Core



Giảng Viên Giảng Dạy:
ThS. Nguyễn Minh Tân
ThS. Đặng Đình Hòa
ThS. Trần Công Thanh
HỌC KỲ III – NĂM HỌC 2023-2024



KHÓA 27T-IT



Nội dung

- 1 Maven
- 2 Tổng quan Spring Framework
- 3 Kiến trúc Spring Framework
- 4 Spring Project
- 5 Q & A



MAVEN

Quản lý dự án với
MAVEN



Maven là gì



- MAVEN là một công cụ quản lý dự án phổ biến trong phát triển phần mềm.
- Nó giúp đơn giản hóa quá trình xây dựng, quản lý dependencies, testing và deployment của dự án. MAVEN sử dụng cấu trúc dự án chuẩn và tập trung vào việc tự động hóa các tác vụ phát triển.
- Bằng cách sử dụng MAVEN, người phát triển có thể dễ dàng quản lý các thư viện, kiểm soát phiên bản và tạo ra các sản phẩm phần mềm đáng tin cậy và nhất quán.



Lợi ích của Maven trong phát triển phần mềm

- Quản lý Dependencies dễ dàng
- Tự động hóa quá trình Build
- Quản lý phiên bản và Release
- Tích hợp dễ dàng với IDE
- Chuẩn hóa cấu trúc dự án

Quản lý Dependencies dễ dàng



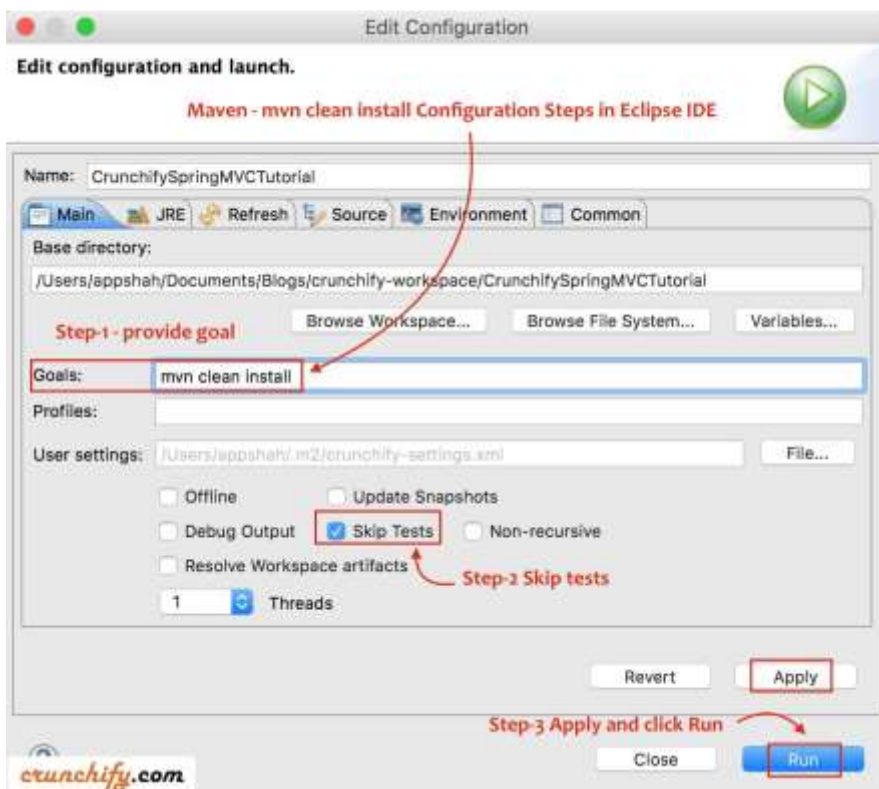
Maven™

Managing
Maven
dependencies

```
<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>org.junit</groupId>
      <artifactId>junit-bom</artifactId>
      <version>5.5.1</version>
      <type>pom</type>
      <scope>import</scope>
    </dependency>
    <dependency>
      <groupId>com.google.guava</groupId>
      <artifactId>guava</artifactId>
      <version>28.1-jre</version>
    </dependency>
  </dependencies>
</dependencyManagement>
```

MAVEN giúp tự động quản lý các dependencies của dự án. Bằng cách định nghĩa dependencies trong file pom.xml, MAVEN sẽ tự động tải về và quản lý các thư viện và phiên bản tương thích cho dự án

Tự động hóa quá trình Build



MAVEN cung cấp khả năng tự động hóa quá trình build dự án. Bằng cách định nghĩa các mục tiêu (goals) trong file pom.xml, người phát triển có thể thực hiện các tác vụ như biên dịch mã nguồn, đóng gói ứng dụng và tạo ra các phiên bản phần mềm một cách dễ dàng.

Quản lý phiên bản và Release



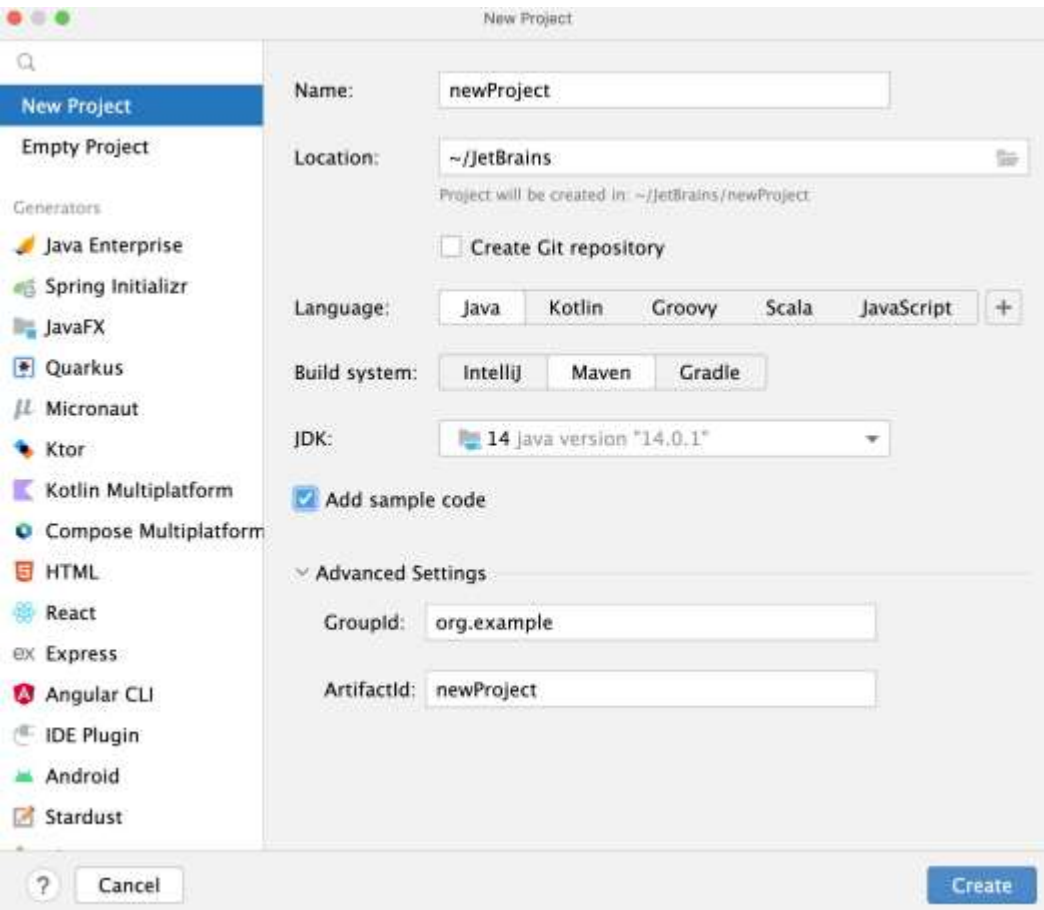
```
<dependency>  
  <groupId>software.amazon.awssdk</groupId>  
  <artifactId>s3control</artifactId>  
  <version>2.17.148</version>  
</dependency>
```

Released on: 12 Mar 2022

<https://www.roseindia.net>

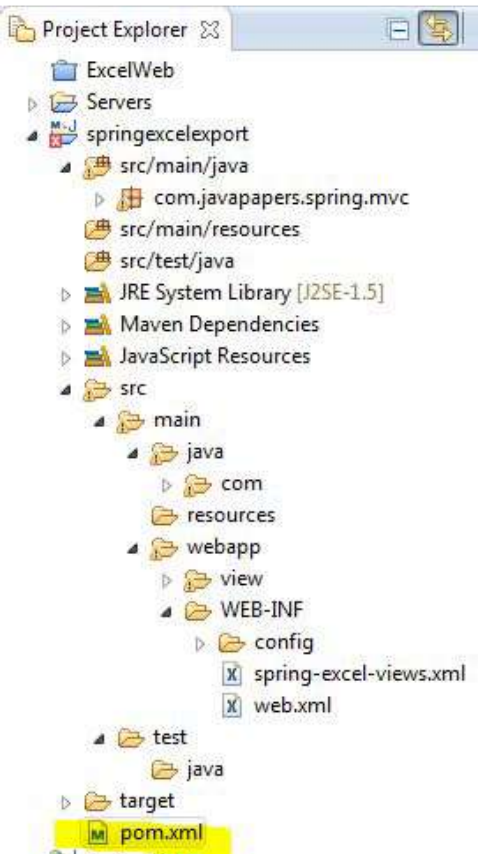
MAVEN hỗ trợ quản lý phiên bản của dự án một cách chặt chẽ. Người phát triển có thể dễ dàng xác định và cập nhật phiên bản của dự án thông qua file pom.xml. Ngoài ra, MAVEN cung cấp các plugin hữu ích để thực hiện quá trình release và deployment của dự án

Tích hợp dễ dàng với IDE



MAVEN tích hợp tốt với các môi trường phát triển tích hợp (IDE) phổ biến như Eclipse, IntelliJ IDEA và NetBeans. Điều này giúp người phát triển dễ dàng thực hiện các tác vụ MAVEN từ giao diện IDE mà không cần phải sử dụng dòng lệnh.

Chuẩn hóa cấu trúc dự án



Directory	Description
src/main/java	Your Java source code
src/main/resources	Properties / config files used by your app
src/main/webapp	JSP files and web config files other web assets (images, css, js, etc)
src/test	Unit testing code and properties
target	Destination directory for compiled code. Automatically created by Maven

MAVEN thúc đẩy việc sử dụng cấu trúc dự án chuẩn, giúp tổ chức dự án một cách rõ ràng và dễ quản lý. Sự chuẩn hóa này giúp tăng khả năng mở rộng, tái sử dụng mã nguồn và tích hợp dễ dàng với các công cụ và quy trình phát triển khác.



File POM.XML



<https://maven.apache.org/guides/introduction/introduction-to-the-pom.html>

Minimal POM

The minimum requirement for a POM are the following:

- **project** root
- **modelVersion** - should be set to 4.0.0
- **groupId** - the id of the project's group.
- **artifactId** - the id of the artifact (project)
- **version** - the version of the artifact under the specified group

Here's an example:

```
1. <project>
2.   <modelVersion>4.0.0</modelVersion>
3.
4.   <groupId>com.mycompany.app</groupId>
5.   <artifactId>my-app</artifactId>
6.   <version>1</version>
7. </project>
```

```
1. <project>
2.   ...
3.   <properties>
4.     <mavenVersion>3.0</mavenVersion>
5.   </properties>
6.
7.   <dependencies>
8.     <dependency>
9.       <groupId>org.apache.maven</groupId>
10.      <artifactId>maven-artifact</artifactId>
11.      <version>${mavenVersion}</version>
12.    </dependency>
13.    <dependency>
14.      <groupId>org.apache.maven</groupId>
15.      <artifactId>maven-core</artifactId>
16.      <version>${mavenVersion}</version>
17.    </dependency>
18.  </dependencies>
19.  ...
20. </project>
```

× pom.xml (Book)

```
<groupId>com.example.maven</groupId>
<version>1.0-SNAPSHOT</version>
</parent>
<modelVersion>4.0.0</modelVersion>
<artifactId>Book</artifactId>
<dependencies>
  <dependency...>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit-dep</artifactId>
    <version>4.10</version>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot</artifactId>
    <version>2.1.7.RELEASE</version>
  </dependency>
</dependencies>
</project>
```



MAVEN REPOSITORY

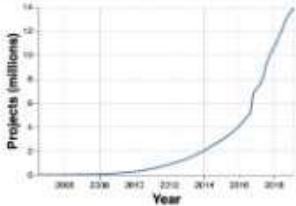
<https://mvnrepository.com/>

MVN REPOSITORY

Search for groups, artifacts, categories

Search

Indexed Artifacts (34.3M)



Popular Categories

[Testing Frameworks & Tools](#)

[Android Packages](#)

[Logging Frameworks](#)

[Java Specifications](#)

[JSON Libraries](#)


[JVM Languages](#)

[Core Utilities](#)

[Language Runtime](#)


[Mocking](#)

What's New in Maven




Scala3 Library Bootstrapped
[org.scala-lang » scala3-library » 3.3.2-RC1-bin-20230623-17...](#)
Standard library for the Scala Programming Language Version 3
Last Release on Jun 24, 2023

6,004 usages
Apache



Scala3 Library Bootstrapped
[org.scala-lang » scala3-library » 3.3.2-RC1-bin-20230623-17...](#)
Standard library for the Scala Programming Language Version 3
Last Release on Jun 24, 2023

6,004 usages
Apache



JMonkeyEngine
[org.jmonkeyengine » jme3-core » 3.6.1-stable](#)
jMonkeyEngine is a 3-D game engine for adventurous Java developers
Last Release on Jun 24, 2023

82 usages
BSD



TỔNG QUAN VỀ SPRING FRAMEWORK

Đơn giản hóa phát triển ứng dụng:

Spring Framework tập trung vào việc giảm bớt sự phức tạp của phát triển ứng dụng Java bằng cách cung cấp các khái niệm và công cụ giúp làm giảm boilerplate code, tạo điều kiện thuận lợi cho việc xây dựng các ứng dụng mạnh mẽ mà



TỔNG QUAN VỀ SPRING FRAMEWORK

Hỗ trợ các nguyên tắc của lập trình hướng đối tượng (OOP)

Spring Framework khuyến khích việc sử dụng các nguyên tắc của lập trình hướng đối tượng như dependency injection (DI), inversion of control (IoC) và hợp thành (composition) để tạo ra mã nguồn dễ đọc, linh hoạt và dễ bảo trì, không cần đầu tư quá nhiều công sức.



TỔNG QUAN VỀ SPRING FRAMEWORK

Tích hợp dễ dàng:

Spring Framework cung cấp các công cụ và tính năng để tương tác và tích hợp với các công nghệ và framework khác. Nó hỗ trợ việc tích hợp với các công nghệ như Java Enterprise Edition (Java EE), Hibernate, JPA, JDBC, RESTful Web Services, Thymeleaf, và nhiều hơn nữa.

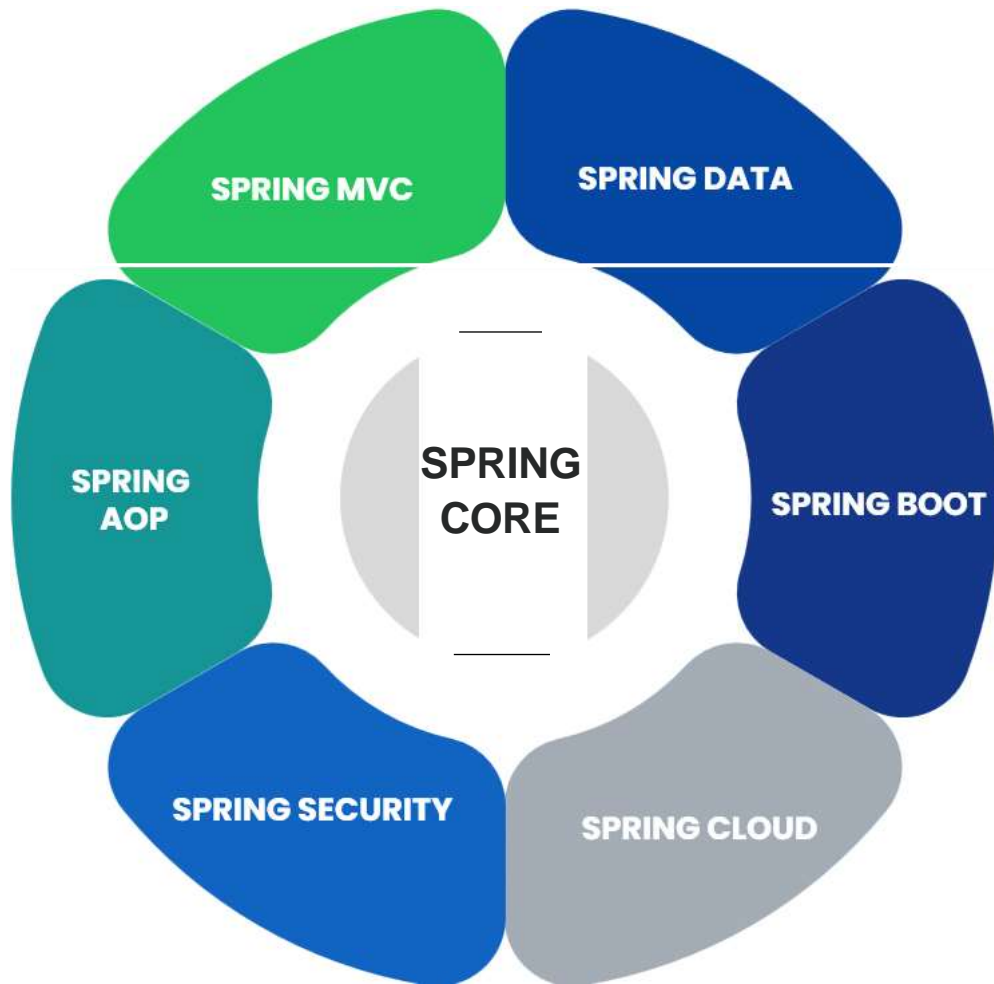


TỔNG QUAN VỀ SPRING FRAMEWORK

Hỗ trợ cho các khía cạnh phát triển ứng dụng:

Spring Framework cung cấp nhiều module và tính năng để hỗ trợ các khía cạnh phát triển ứng dụng như xử lý yêu cầu web, quản lý giao dịch, xác thực và phân quyền, xử lý dữ liệu, và nhiều khía cạnh khác.

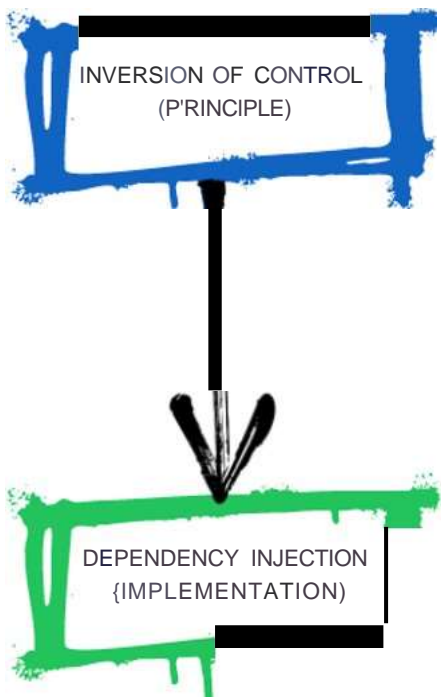
SPRING CORE



Spring Core - Một trong những phần cơ bản của Spring. Toàn bộ Spring Framework và các dự án khác của Spring đều được phát triển dựa trên Spring Core. Spring Core chứa các thành phần quan trọng sau

- IoC (Inversion of Control)
- DI (Dependency Injection)
- Beans
- Context
- SpEL (Spring Expression Language)
- IoC Container

INVERSION OF CONTROL & DEPENDENCY INJECTION



- Inversion of Control (IoC) is a Software Design Principle, independent of language, which does not actually create the objects but describes the way in which object is being created.
- IoC is the principle, where the control flow of a program is inverted: instead of the programmer controlling the flow of a program, the framework or service takes control of the program flow.
- Dependency Injection is the pattern through which Inversion of Control achieved.
- Through Dependency Injection, the responsibility of creating objects is shifted from the application to the Spring IoC container. It reduces coupling between multiple objects as it is dynamically injected by the framework.



Module và Kiến trúc Spring Framework

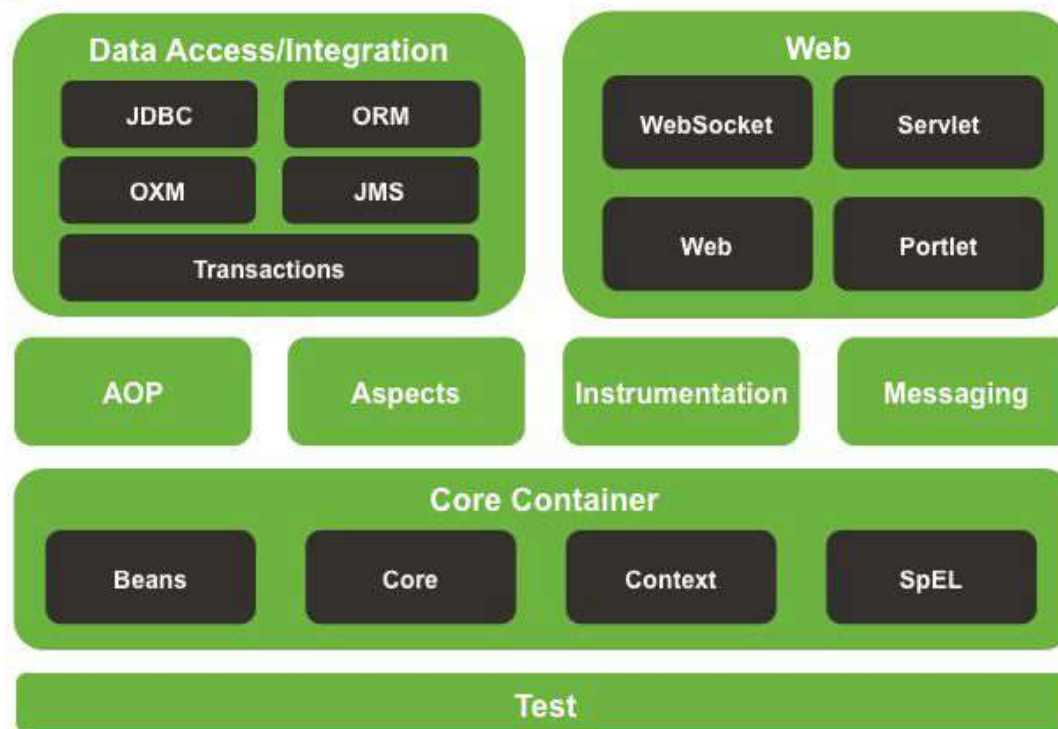


Các Module và kiến trúc của Spring Framework

<https://docs.spring.io/spring-framework/docs/3.0.x/spring-framework-reference/html/overview.html>



Spring Framework Runtime



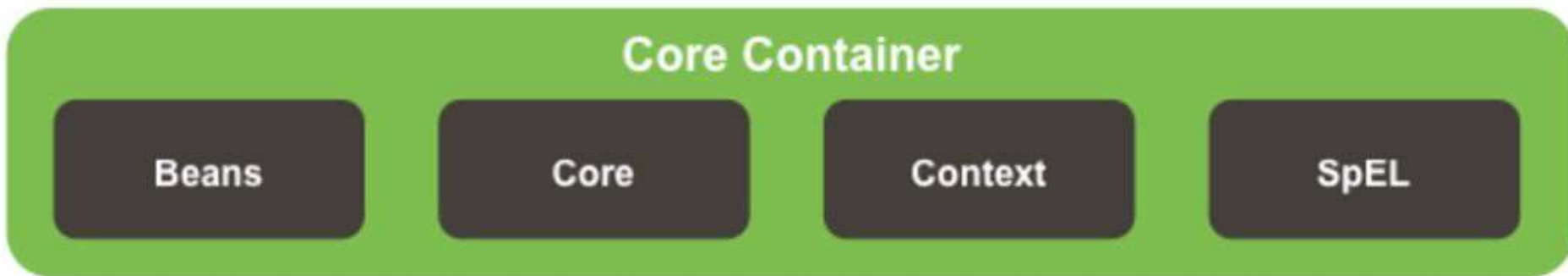
Kiến trúc của Spring Framework



Bean: Bean là thành phần cốt lõi trong Core Container của Spring và là trách nhiệm chính của Spring Container. Bean đại diện cho các đối tượng trong ứng dụng được quản lý bởi Spring. Spring Container sẽ tạo, cấu hình và quản lý các Bean. Các Bean được khai báo và cấu hình thông qua các tệp cấu hình XML, Annotations hoặc Java-based configuratio



Kiến trúc của Spring Framework



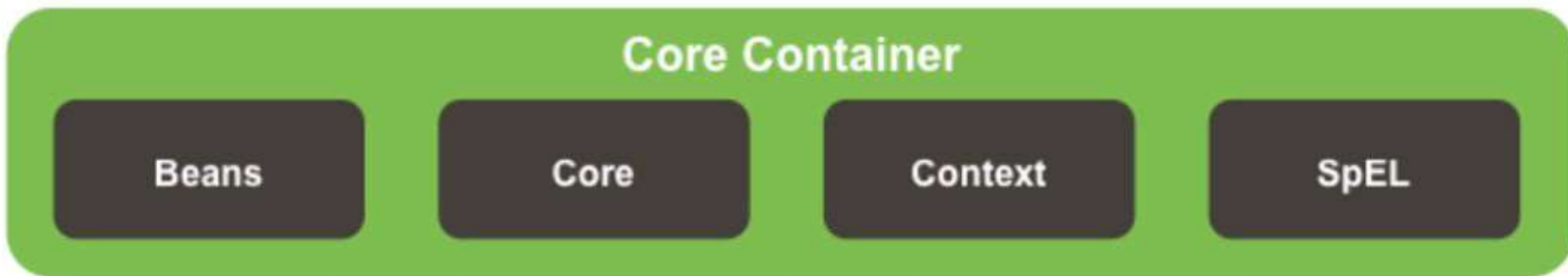
Core: Core là một thành phần quan trọng khác trong Core Container. Nó chứa các lớp và giao diện cung cấp các tính năng cốt lõi của Spring như dependency injection (DI) và inversion of control (IoC). Core cung cấp các lớp và giao diện như `ApplicationContext`, `BeanFactory`, `ResourceLoader`, và các lớp tiện ích khác để hỗ trợ việc quản lý và tạo ra các Bean

Kiến trúc của Spring Framework



Context: Context là thành phần thứ ba trong Core Container của Spring. Nó cung cấp các tính năng phục vụ cho việc tạo và quản lý các Bean, bao gồm cả xử lý sự kiện, quản lý giao dịch và quản lý chuỗi thông báo. Context được xây dựng dựa trên Core và cung cấp các chức năng mở rộng để giúp quản lý các Bean trong quá trình thực thi ứng dụng.

Kiến trúc của Spring Framework



SpEL (Spring Expression Language) là một ngôn ngữ biểu thức mạnh trong Core Container của Spring. SpEL được sử dụng trong các tệp cấu hình XML, Annotations và Java-based configuration để thực hiện các tác vụ như:

- Truy cập và đặt giá trị thuộc tính của đối tượng.
- Gọi các phương thức của đối tượng.
- Thực hiện các phép tính và toán tử.
- Xử lý biểu thức điều kiện và vòng lặp.
- Tương tác với các tài nguyên bên ngoài như properties file, environment variables, hoặc các bean khác trong Spring Container.



Kiến trúc của Spring Framework

AOP

Aspects

Instrumentation

Messaging

AOP (Aspect-Oriented Programming) là một mô hình lập trình được sử dụng trong Spring Boot để tách biệt các khía cạnh (concerns) khác nhau trong ứng dụng. Nó cho phép bạn tách riêng các mục đích của mã như **logging, giao dịch, bảo mật, caching và xử lý ngoại lệ**, và áp dụng chúng vào mã của bạn mà không cần thay đổi trực tiếp mã đó.



Kiến trúc của Spring Framework

AOP

Aspects

Instrumentation

Messaging

Aspect: Aspect là một đối tượng chứa mã thực hiện các khía cạnh chung như logging, giao dịch, bảo mật, và nhiều hơn nữa. Aspect được xác định bằng cách sử dụng annotations hoặc XML configuration.



Kiến trúc của Spring Framework

AOP

Aspects

Instrumentation

Messaging

"Instrumentation" có thể được hiểu như việc mở rộng hoặc tăng cường chức năng của một ứng dụng, đặc biệt là các khía cạnh liên quan đến giám sát (monitoring), phân tích (analytics) và ghi lại (logging).



Kiến trúc của Spring Framework

AOP

Aspects

Instrumentation

Messaging

"**Message**" thường liên quan đến việc truyền thông tin hoặc dữ liệu giữa các thành phần của hệ thống. Nó có thể ám chỉ việc gửi và nhận các thông điệp (message) qua các giao thức như HTTP, JMS (Java Message Service), AMQP (Advanced Message Queuing Protocol), Kafka, v.v. Spring Boot cung cấp nhiều cơ chế để hỗ trợ giao tiếp qua tin nhắn, bao gồm Spring Integration, Spring Cloud Stream và Spring AMQP.

Kiến trúc của Spring Framework



```
Statement statement = connection.createStatement();
ResultSet resultSet = statement.executeQuery("SELECT * FROM employees");

while (resultSet.next()) {
    int id = resultSet.getInt("id");
    String name = resultSet.getString("name");
    int age = resultSet.getInt("age");

    System.out.println("ID: " + id + ", Name: " + name + ", Age: " + age);
}
```

JDBC (Java Database Connectivity) là một API (Application Programming Interface) trong Java cho phép các ứng dụng Java tương tác với các cơ sở dữ liệu quan hệ.

- Nó cung cấp một tập hợp các lớp và giao diện để thực hiện các thao tác như kết nối đến cơ sở dữ liệu, truy vấn dữ liệu, thêm, sửa, xóa dữ liệu, và quản lý các tài nguyên liên quan đến cơ sở dữ liệu.
- JDBC là một tiêu chuẩn trong ngôn ngữ Java để giao tiếp với các cơ sở dữ liệu quan hệ phổ biến như MySQL, Oracle, SQL Server, v.v. Điều này cho phép phát triển ứng dụng Java kết nối và làm việc với dữ liệu trong cơ sở dữ liệu cách dễ dàng và linh hoạt.

Kiến trúc của Spring Framework



```
import javax.persistence.*;

@Entity
@Table(name = "employees")
public class Employee {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @Column(name = "name")
    private String name;

    // Getters and setters

}
```

ORM (Object-Relational Mapping) là một kỹ thuật trong lập trình phần mềm để ánh xạ dữ liệu giữa các đối tượng trong mã lập trình và cơ sở dữ liệu quan hệ. Nó giúp đơn giản hóa và tự động hóa quá trình truy xuất, lưu trữ và tương tác với dữ liệu trong cơ sở dữ liệu từ mã Java (hoặc các ngôn ngữ lập trình khác) mà không cần viết các truy vấn SQL chi tiết. một cách dễ dàng

Kiến trúc của Spring Framework



```
import javax.xml.bind.*;

public class XmlToJavaExample {
    public static void main(String[] args) throws JAXBException {
        String xml = "<person><name>John</name><age>30</age></person>";

        JAXBContext jaxbContext = JAXBContext.newInstance(Person.class);
        Unmarshaller unmarshaller = jaxbContext.createUnmarshaller();

        Person person = (Person) unmarshaller.unmarshal(new StringReader(xml));
        System.out.println("Name: " + person.getName());
        System.out.println("Age: " + person.getAge());
    }
}
```

OXM (Object-XML Mapping) là một kỹ thuật trong lập trình phần mềm để ánh xạ dữ liệu giữa các đối tượng trong mã lập trình và dữ liệu XML. Nó cho phép chuyển đổi (mapping) dữ liệu giữa các đối tượng Java và các tài liệu XML một cách dễ dàng và tự động.

Kiến trúc của Spring Framework



```
// Chờ và nhận một tin nhắn
Message receivedMessage = consumer.receive();
if (receivedMessage instanceof TextMessage) {
    TextMessage textMessage = (TextMessage) receivedMessage;
    String messageText = textMessage.getText();
    System.out.println("Received message: " + messageText);
}
```

JMS is a standard Java API that allows a Java application to send messages to another application. It is highly scalable and allows us to loosely couple applications using asynchronous messaging. Using JMS we can read, send and read messages.

Kiến trúc của Spring Framework



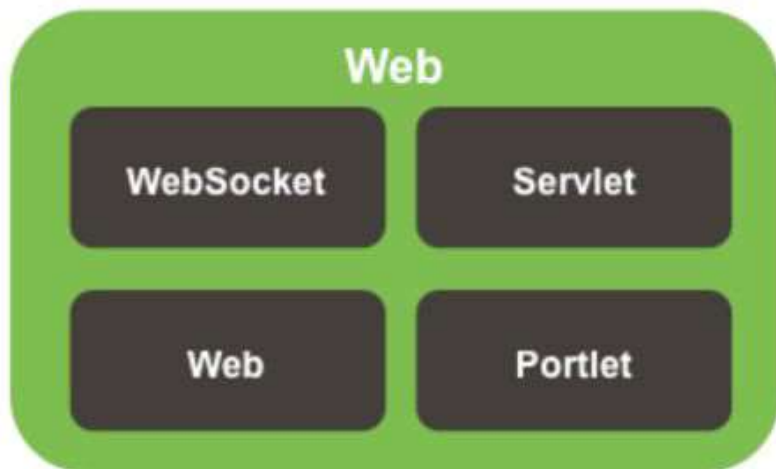
```
@Service
@Transactional
public class UserServiceImpl implements UserService {
    @Autowired
    private UserRepository userRepository;

    public void saveUser(User user) {
        userRepository.save(user);
    }

    public User getUserById(int id) {
        return userRepository.findById(id);
    }
}
```

Transactions (giao dịch) là một cơ chế cho phép thực hiện các hoạt động cơ bản của cơ sở dữ liệu một cách an toàn và đáng tin cậy. Giao dịch giúp đảm bảo tính toàn vẹn của dữ liệu bằng cách đảm bảo rằng các thao tác cơ sở dữ liệu được thực hiện hoàn toàn hoặc không thực hiện hoàn toàn.

Kiến trúc của Spring Framework



```
const socket = new WebSocket('ws://localhost:8080');

socket.onopen = function() {
  console.log('Kết nối WebSocket đã được thiết lập.');
```



```
};

socket.onmessage = function(event) {
  const message = event.data;
  console.log('Nhận dữ liệu từ máy chủ: ' + message);
};

socket.onclose = function() {
  console.log('Kết nối WebSocket đã bị đóng.');
```

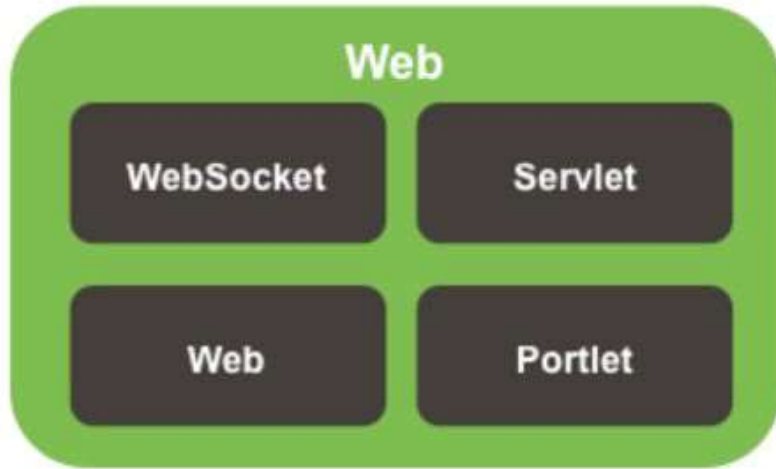


```
};

const data = 'Dữ liệu cần gửi';
socket.send(data);
```

WebSocket là một giao thức truyền thông hai chiều (full-duplex) dựa trên TCP, cho phép thiết lập kết nối liên tục và thời gian thực giữa một máy khách (client) và một máy chủ (server) trên web. Nó cho phép truyền thông tin hai chiều đồng thời, tức là cả client và server có thể gửi và nhận dữ liệu một cách đồng thời trong một kết nối duy nhất.

Kiến trúc của Spring Framework



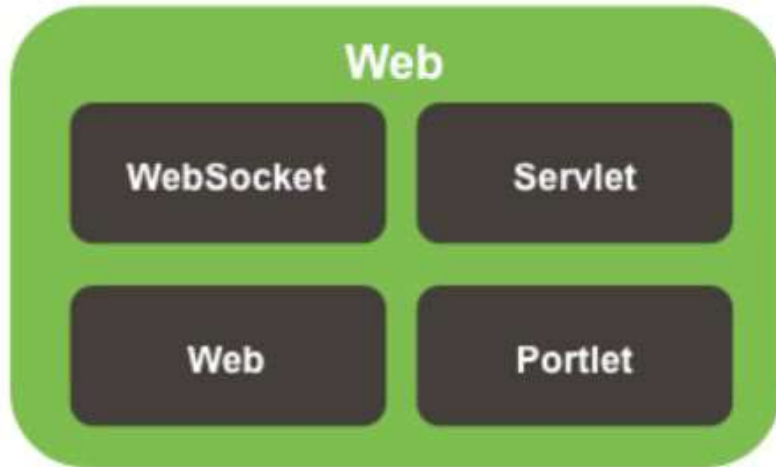
```
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.ResponseBody;

@Controller
@RequestMapping("/hello")
public class HelloController {

    @GetMapping
    @ResponseBody
    public String sayHello() {
        return "Xin chào từ Spring MVC!";
    }
}
```

Web Trong Spring Framework, để xây dựng ứng dụng web, bạn có thể sử dụng module Spring Web MVC

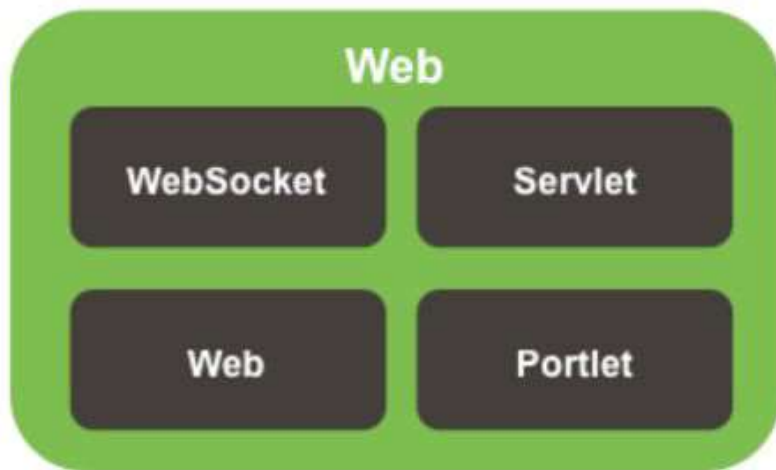
Kiến trúc của Spring Framework



```
@Override  
protected void doGet(HttpServletRequest request, HttpServletResponse  
    throws ServletException, IOException {  
    response.setContentType("text/html");  
    PrintWriter out = response.getWriter();  
    out.println("<html><body>");  
    out.println("<h1>Hello, World!</h1>");  
    out.println("</body></html>");  
    out.close();  
}
```

Servlet là một thành phần của Java Enterprise Edition (Java EE) được sử dụng để xây dựng ứng dụng web. Nó là một lớp Java được viết để xử lý yêu cầu (request) từ client và trả về phản hồi (response) tương ứng

Kiến trúc của Spring Framework



```
import org.springframework.context.annotation.*;
import org.springframework.web.portlet.config.*;

@Configuration
@EnableWebMvc
public class MyPortletConfig implements WebMvcConfigurer {

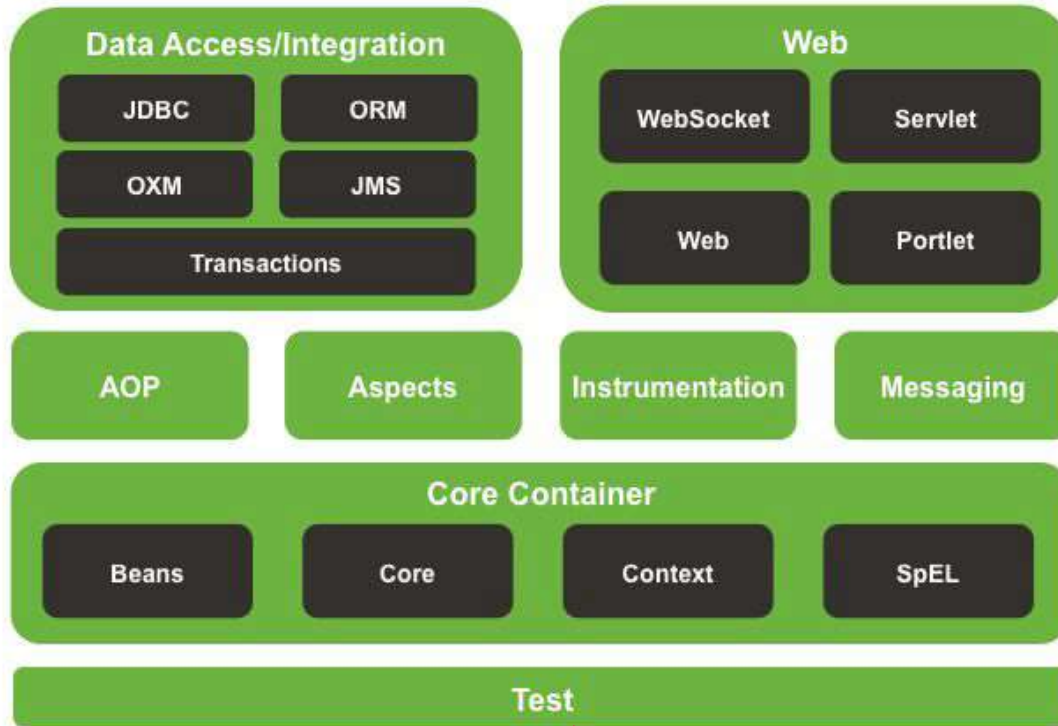
    // Cấu hình khác (nếu có)
}
```

Portlet là một khái niệm trong Java Portlet Specification (JSR-286) và được sử dụng để xây dựng các ứng dụng web phù hợp với mô hình portal.

Kiến trúc của Spring Framework



Spring Framework Runtime





Spring Project



Spring Project

The screenshot shows the Spring by VMware Tanzu website. The header includes the Spring logo and navigation links: Why Spring, Learn, Projects, Academy, and Support. The main banner features the text "Meet the Spring team this August at SpringOne" and "Las Vegas • August 22-24". A "LEARN MORE" button is present. The "Projects" dropdown menu is open, listing various Spring projects and development tools.

spring[®] by VMware Tanzu

Why Spring ▾ Learn ▾ Projects ▾ Academy ▾ Support ▾

Meet the Spring team this August at SpringOne

Las Vegas • August 22-24

LEARN MORE

Overview

Spring Boot

Spring Framework

Spring Cloud

Spring Cloud Data Flow

Spring Data

Spring Integration

Spring Batch

Spring Security

[View all projects](#)

DEVELOPMENT TOOLS

Spring Tools 4

Spring Initializr [↗](#)

NEWS | State of Spring Summit



Spring Framework

Provides core support for dependency injection, transaction management, web apps, data access, messaging, and more.

- Dự án cốt lõi của Spring Framework
- Tập trung vào việc cung cấp một nền tảng phát triển ứng dụng Java
- Mục tiêu chính là cung cấp các tính năng và mô hình để xây dựng các ứng dụng Java linh hoạt, dễ bảo trì và dễ mở rộng



Spring Project



Spring Boot

Takes an opinionated view of building Spring applications and gets you up and running as quickly as possible.

- Giúp người dùng tạo các ứng dụng Spring nhanh chóng
- Cung cấp cấu hình mặc định và tích hợp sẵn cho một số công nghệ phổ biến
- Giúp giảm thiểu sự phức tạp trong việc thiết lập và triển khai ứng dụng



Spring Project



Spring Data

Provides a consistent approach to data access – relational, non-relational, map-reduce, and beyond.

- Cung cấp các công cụ dễ dùng và mạnh để làm việc với cơ sở dữ liệu trong ứng dụng Java.
- Cung cấp một tập hợp các API và công cụ để giảm thiểu việc lặp lại khi làm việc với cơ sở dữ liệu, cho phép bạn tập trung vào việc phát triển logic ứng dụng chính.



Spring Project



Spring Cloud

Provides a set of tools for common patterns in distributed systems. Useful for building and deploying microservices.

- Cung cấp một bộ công cụ và thư viện để xây dựng và triển khai các ứng dụng cloud.
- Tăng tính mạnh mẽ và linh hoạt của ứng dụng bằng cách cung cấp các tính năng như cấu hình phân tán, quản lý dịch vụ, điều hướng tải và theo dõi.



Spring Project



Spring Security

Protects your application with comprehensive and extensible authentication and authorization support.

- Cung cấp các cơ chế bảo mật mạnh cho ứng dụng Java.
- Mục tiêu của Spring Security là giúp phát triển ứng dụng an toàn và bảo vệ khỏi các cuộc tấn công như xâm nhập, giả mạo danh tính và tấn công từ chối dịch vụ.



Spring Project



Spring Web Services

Facilitates the development of contract-first SOAP web services.

- Phát triển các dịch vụ web dựa trên tiêu chuẩn SOAP (Simple Object Access Protocol)
- Cung cấp cách tiếp cận linh hoạt và dễ sử dụng để phát triển các ứng dụng dịch vụ web



Spring Project



Spring Cloud Data Flow

Provides an orchestration service for composable data microservice applications on modern runtimes.



Spring for GraphQL

Spring for GraphQL provides support for Spring applications built on GraphQL Java.



Spring Authorization Server

Provides a secure, light-weight, and customizable foundation for building OpenID Connect 1.0 Identity Providers and OAuth2 Authorization Server products.



Spring Session

Provides an API and implementations for managing a user's session information.



Spring Project



Spring LDAP

Simplifies the development of applications that use LDAP by using Spring's familiar template-based approach.



Spring Integration

Supports the well-known Enterprise Integration Patterns through lightweight messaging and declarative adapters.



Spring HATEOAS

Simplifies creating REST representations that follow the HATEOAS principle.



Spring Batch

Simplifies and optimizes the work of processing high-volume batch operations.



Spring Project



Spring REST Docs

Lets you document RESTful services by combining hand-written documentation with auto-generated snippets produced with Spring MVC Test or REST Assured.



Spring Flo

Provides a JavaScript library that offers a basic embeddable HTML5 visual builder for pipelines and simple graphs.



Spring for Apache Kafka

Provides Familiar Spring Abstractions for Apache Kafka.



Spring Shell

Makes writing and testing RESTful applications easier with CLI-based resource discovery and interaction.



Spring Project



Spring StateMachine

Provides a framework for application developers to use state machine concepts with Spring applications.



Spring Web Flow

Supports building web applications that feature controlled navigation, such as checking in for a flight or applying for a loan.



TRƯỜNG ĐẠI HỌC
VĂN LANG

KHOA CÔNG NGHỆ THÔNG TIN

