



# LẬP TRÌNH JAVA NÂNG CAO

## CHƯƠNG 2: Spring Framework



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

Framework là gì

2

Spring Framework

3

SpringBoot Framework

4

So sánh Spring vs Springboot

5

Q & A

# 1. Giới thiệu chung

What is  
**Framework?**



**Định nghĩa của Framework:**

- Trong lập trình, Framework là **một tập hợp các thư viện, công cụ, mô hình thiết kế và quy tắc** mà lập trình viên có thể sử dụng để xây dựng và phát triển các ứng dụng, hệ thống hoặc phần mềm một cách nhanh chóng và hiệu quả.

# Ví dụ

Làm sao để sắp xếp một mảng số nguyên tăng dần/ giảm dần?

```
public static void sortAscending(int[] array) {  
    int n = array.length;  
    for (int i = 0; i < n-1; i++) {  
        for (int j = 0; j < n-i-1; j++) {  
            if (array[j] > array[j+1]) {  
                // Hoán đổi các phần tử nếu thứ tự sai  
                int temp = array[j];  
                array[j] = array[j+1];  
                array[j+1] = temp;  
            }  
        }  
    }  
}
```

```
public static void sortAscending(Integer[] array) {  
    List<Integer> list = Arrays.asList(array);  
    Collections.sort(list);  
}  
  
public static void sortDescending(Integer[] array) {  
    List<Integer> list = Arrays.asList(array);  
    Collections.sort(list, Collections.reverseOrder());  
}
```

# Ví dụ

Làm sao để lưu một đối tượng xuống cơ sở dữ liệu?

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;

public class StudentDAO {
    private static final String DB_URL = "jdbc:mysql://localhost:3306/mydatabase";
    private static final String DB_USERNAME = "username";
    private static final String DB_PASSWORD = "password";

    public void saveStudent(Student student) {
        Connection conn = null;
        PreparedStatement stmt = null;

        try {
            conn = DriverManager.getConnection(DB_URL, DB_USERNAME, DB_PASSWORD);
            String query = "INSERT INTO students (student_id, full_name, score) VALUES (?, ?, ?)";
            stmt = conn.prepareStatement(query);
            stmt.setString(1, student.getStudentId());
            stmt.setString(2, student.getFullName());
            stmt.setDouble(3, student.getScore());
            stmt.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            try {
                if (stmt != null)
                    stmt.close();
                if (conn != null)
                    conn.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }
}
```



# Ví dụ

Làm sao để lưu một đối tượng xuống cơ sở dữ liệu?

```
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.cfg.Configuration;

public class StudentDAO {
    private static SessionFactory sessionFactory;

    public void saveStudent(Student student) {
        Transaction transaction = null;
        try (Session session = getSessionFactory().openSession()) {
            transaction = session.beginTransaction();
            session.save(student);
            transaction.commit();
        } catch (Exception e) {
            if (transaction != null) {
                transaction.rollback();
            }
            e.printStackTrace();
        }
    }

    private SessionFactory getSessionFactory() {
        if (sessionFactory == null) {
            try {
                Configuration configuration = new Configuration().configure();
                sessionFactory = configuration.buildSessionFactory();
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
        return sessionFactory;
    }
}
```


# Ví dụ

Làm sao để lưu một đối tượng xuống cơ sở dữ liệu?

```
public void saveStudent(Student student) {
    Connection conn = null;
    PreparedStatement stmt = null;

    try {
        conn = DriverManager.getConnection(DB_URL, DB_USERNAME, DB_PASSWORD);
        String query = "INSERT INTO students (student_id, full_name, score) VALUES (?, ?, ?)";
        stmt = conn.prepareStatement(query);
        stmt.setString(1, student.getStudentId());
        stmt.setString(2, student.getFullName());
        stmt.setDouble(3, student.getScore());
        stmt.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        try {
            if (stmt != null)
                stmt.close();
            if (conn != null)
                conn.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

```
public void saveStudent(Student student) {
    Transaction transaction = null;
    try (Session session = sessionFactory().openSession()) {
        transaction = session.beginTransaction();
        session.save(student);
        transaction.commit();
    } catch (Exception e) {
        if (transaction != null) {
            transaction.rollback();
        }
        e.printStackTrace();
    }
}
```



# Vai trò và lợi ích

## **Cung cấp cấu trúc và khung làm việc:**

- ✓ Framework cung cấp một cấu trúc và khung làm việc sẵn có cho việc phát triển phần mềm.
- ✓ Nó định nghĩa các quy tắc, mô hình và công cụ giúp người phát triển xây dựng ứng dụng một cách có hệ thống và có hiệu suất cao





# Vai trò và lợi ích

<https://stratoflow.com/how-to-speed-up-software-development/>

## What does software development speed depend on?

Complexity



Code Quality



Framework




People



Resources



STRATO  
FLOW



# Vai trò và lợi ích


## **Tăng năng suất phát triển:**

- ✓ Framework cung cấp các thư viện, công cụ và tiện ích để giảm bớt công việc lặp lại và tăng năng suất phát triển.
- ✓ Người phát triển có thể tận dụng các chức năng đã được xây dựng sẵn, giảm thời gian và công sức cần thiết để triển khai các tính năng phổ biến.



# Vai trò và lợi ích





# Vai trò và lợi ích

## **Chuẩn hóa và tái sử dụng mã nguồn:**

- ✓ Framework thường xác định các quy tắc và chuẩn mà người phát triển phải tuân thủ. Điều này giúp tạo ra mã nguồn dễ đọc, dễ bảo trì và dễ tái sử dụng.
- ✓ Người phát triển có thể sử dụng lại các thành phần, module và thư viện được xây dựng sẵn trong Framework để giảm thiểu việc viết mã từ đầu.

# Vai trò và lợi ích



## Bảo mật:

Framework thường có các cơ chế bảo mật tích hợp và kiểm tra tự động giúp ngăn chặn các lỗ hổng bảo mật và giảm rủi ro.

‘Nó cung cấp các công cụ và thư viện cho việc kiểm thử, gỡ lỗi và kiểm tra hiệu năng để đảm bảo chất lượng ứng dụng.



# Vai trò và lợi ích



## Hỗ trợ từ cộng đồng và tài liệu:

- ✓ Framework thường có một cộng đồng lớn và nhiệt tình xung quanh nó.
- ✓ Người phát triển có thể tìm kiếm hỗ trợ, chia sẻ kinh nghiệm và nhận được các tài liệu, ví dụ và hướng dẫn từ cộng đồng.
- ✓ Điều này giúp giải quyết các vấn đề phát triển một cách nhanh chóng và hiệu quả.

# Vai trò và lợi ích



## **Cải thiện hiệu suất và khả năng mở rộng:**

- ✓ Framework cung cấp các công cụ và kỹ thuật tối ưu hóa để cải thiện hiệu suất ứng dụng.
- ✓ Nó cung cấp khả năng mở rộng để xử lý



# Các thành phần cơ bản của Framework

- ✓ Thư viện (Library)
- ✓ Mô hình thiết kế (Design pattern)
- ✓ Cấu trúc thư mục (Directory structure)
- ✓ Các công cụ hỗ trợ (Tooling)
- ✓ Cách thức hoạt động (Workflow)

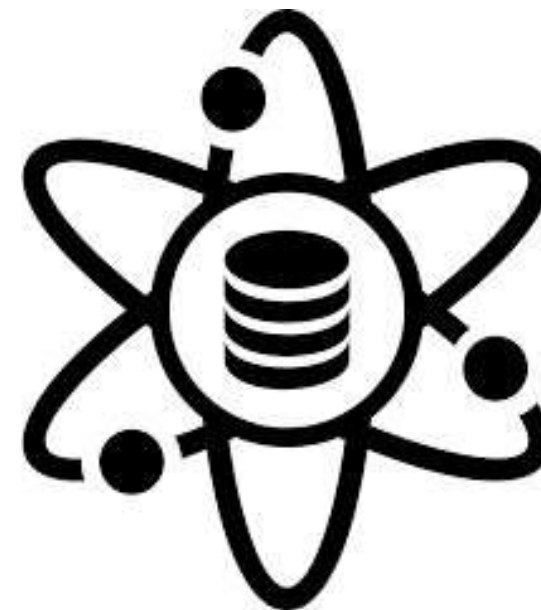


# Các loại Framework phổ biến trong lập trình

- ✓ Web Frameworks
- ✓ Desktop Frameworks

Mobile Frameworks

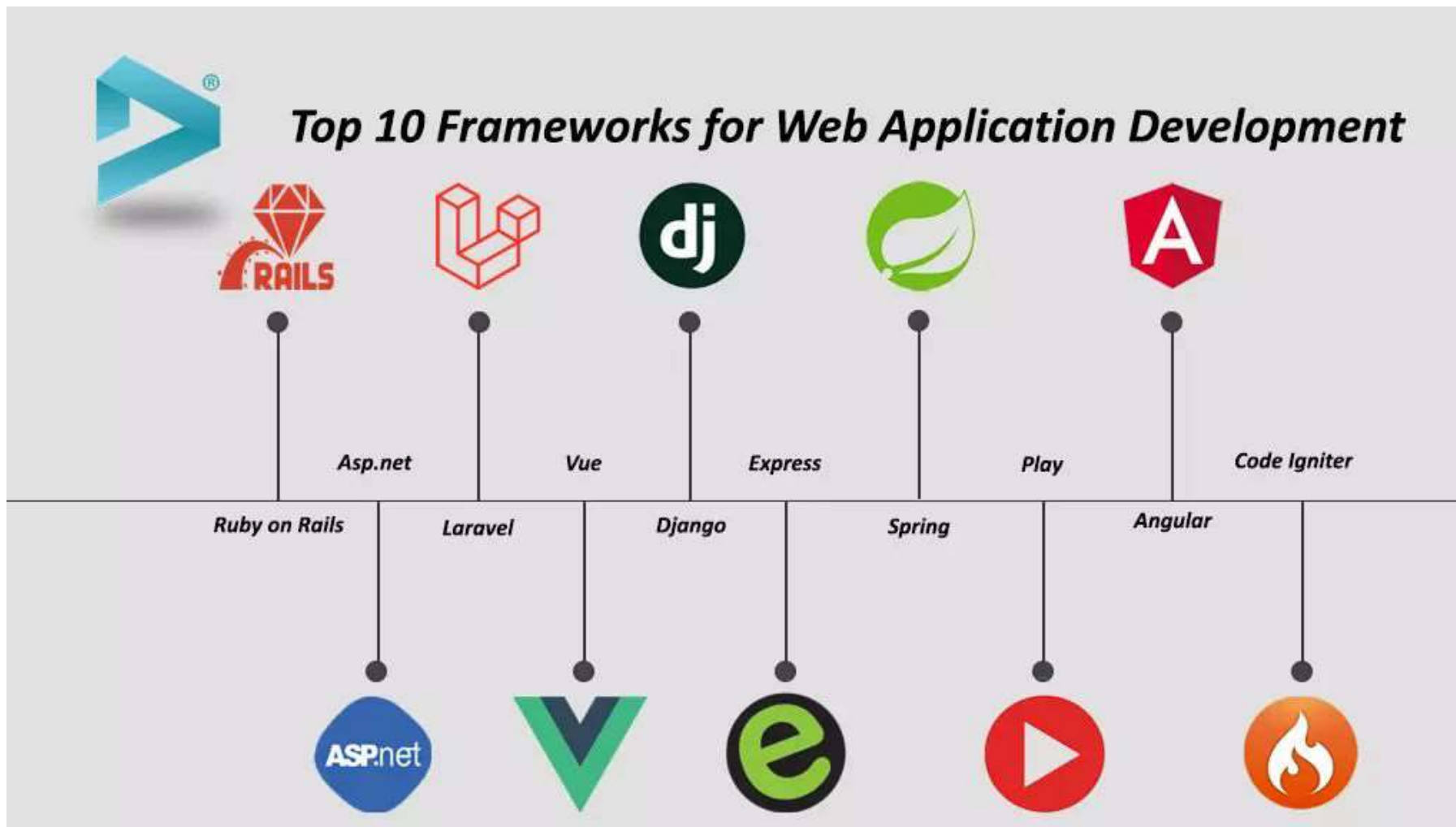
Data Science Frameworks





# Web Frameworks

<https://www.decipherzone.com/blog-detail/top-web-frameworks>

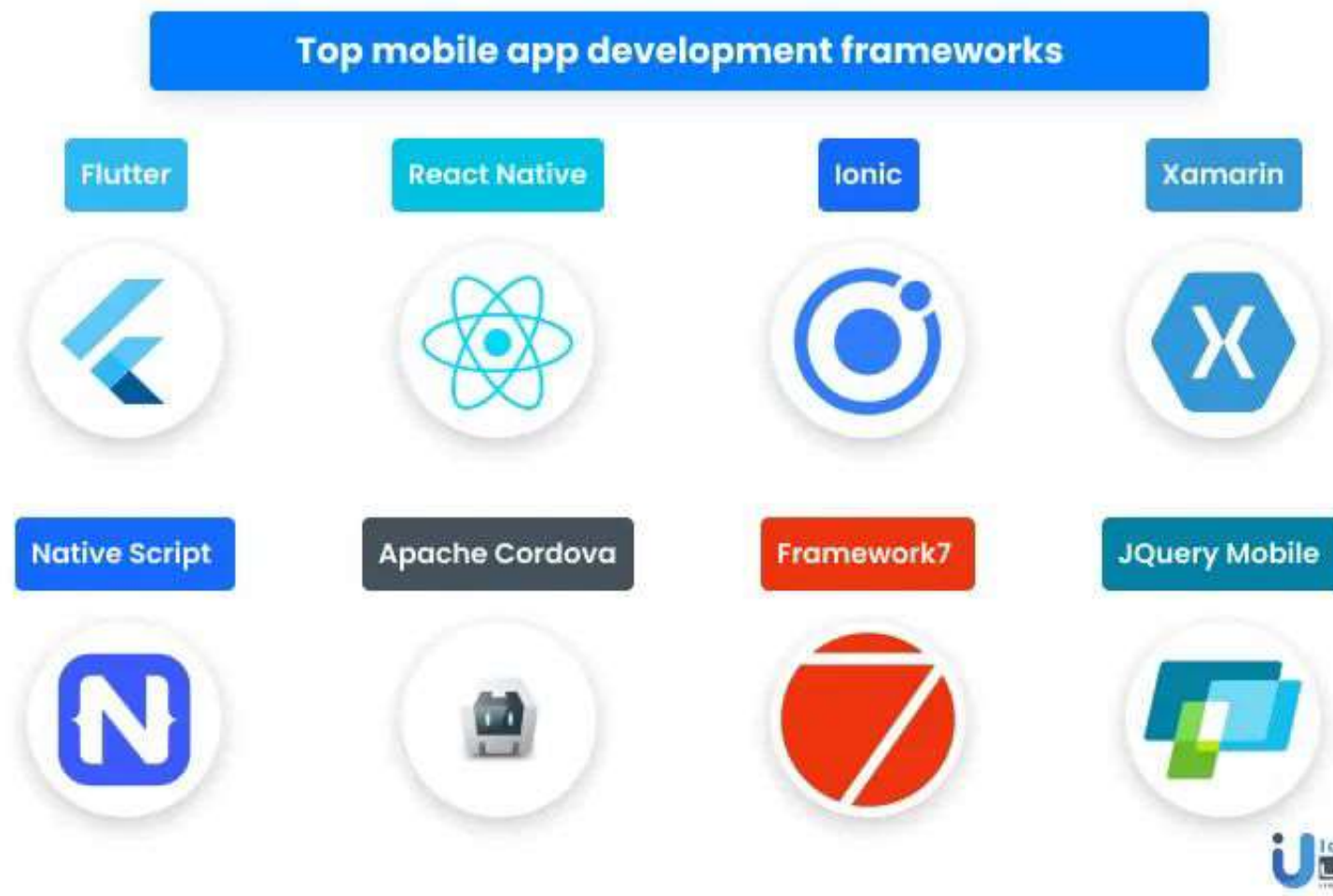






# Mobile Frameworks

<https://ideausher.com/blog/top-mobile-app-developmentframeworks/>



# Desktop Frameworks

<https://medium.com/geekculture/10-tools-for-developing-desktop-applications-with-user-interfaces-f3806ef0d841>



Lazarus



XOJO



# Data Science Frameworks

<https://www.kdnuggets.com/2017/05/frameworks-offerdata-scientists-programming-languages-lack.html>

**django**





# Cách lựa chọn framework phù hợp

A. Yêu cầu dự án

B. Các yếu tố cần xem xét khi lựa chọn

1. Hỗ trợ cộng đồng
2. Hiệu năng và khả năng mở rộng
3. Tính di động và đa nền tảng
4. Thời gian và nguồn lực



# Phương pháp học tập hiệu quả / framework

## 1. Xác định mục tiêu

Giúp bạn tập trung vào những kiến thức và kỹ năng cần thiết để đạt được mục tiêu của mình.

## 2. Tham gia lớp học, nghiên cứu tài liệu

Tìm hiểu thông qua tài liệu hướng dẫn, tài liệu chính thức, sách, video hoặc các nguồn tài liệu trực tuyến khác; hiểu cách Framework hoạt động, các khái niệm, cú pháp và các tính năng quan trọng.

## 3. Thực hành

Tạo các ví dụ đơn giản, theo các hướng dẫn và ví dụ trong tài liệu. Thử các tính năng và chức năng khác nhau của Framework để hiểu





# Phương pháp học tập hiệu quả / framework

## 4. Dự án thực tế

Hãy thử xây dựng một dự án thực tế sử dụng Framework. Dự án có thể là một ứng dụng đơn giản, Trong quá trình xây dựng, bạn sẽ gặp phải các thách thức và học hỏi từ những vấn đề thực tế

## 5. Tham gia cộng đồng và tìm kiếm sự giúp đỡ

Tham gia diễn đàn, nhóm người dùng, trang web chia sẻ kiến thức hoặc các cộng đồng trên mạng xã hội.

## 6. Tiếp tục học tập và nâng cao kỹ năng

Cập nhật kiến thức của mình về Framework, theo dõi các bản cập nhật mới và học các phương pháp tiếp cận mới

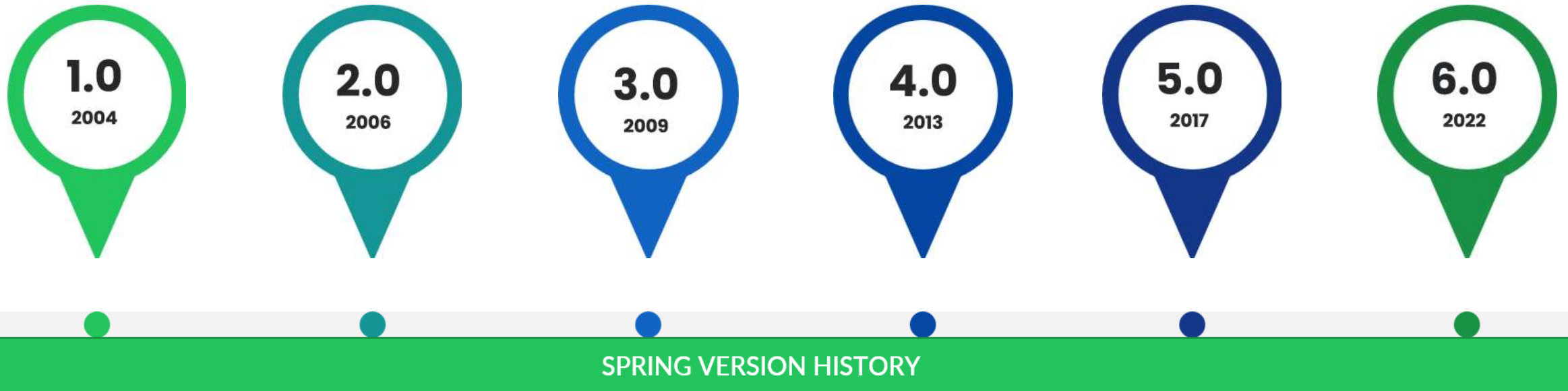


# Spring Framework

- ✓ Spring Framework là một Framework phát triển ứng dụng Java mạnh mẽ và linh hoạt.
- ✓ Nó vô cùng phổ biến.
- ✓ Cung cấp một cơ sở hạ tầng mạnh mẽ để xây dựng các ứng dụng doanh nghiệp, từ các ứng dụng web đơn giản đến các hệ thống phức tạp

# SPRING RELEASE TIMELINE

<https://spring.io/blog/category/releases>



- The first version of Spring was written by Rod Johnson, who released the framework with the publication of his book *Expert One-on-One J2EE Design and Development* in October 2002
- Spring came into being in 2003 as a response to the complexity of the early J2EE specifications. While some consider Java EE and Spring to be in competition, Spring is, in fact, complementary to Java EE. The Spring programming model does not embrace the Java EE platform specification; rather, it integrates with carefully selected individual specifications from the EE umbrella
- Spring continues to innovate and to evolve. Beyond the Spring Framework, there are other projects, such as Spring Boot, Spring Security, Spring Data, Spring Cloud, Spring Batch, among others.



# Tại sao sử dụng Spring Framework

<https://spring.io/why-spring>

## Why Spring?

Spring makes programming Java quicker, easier, and safer for everybody. Spring's focus on speed, simplicity, and productivity has made it the world's most popular Java framework.



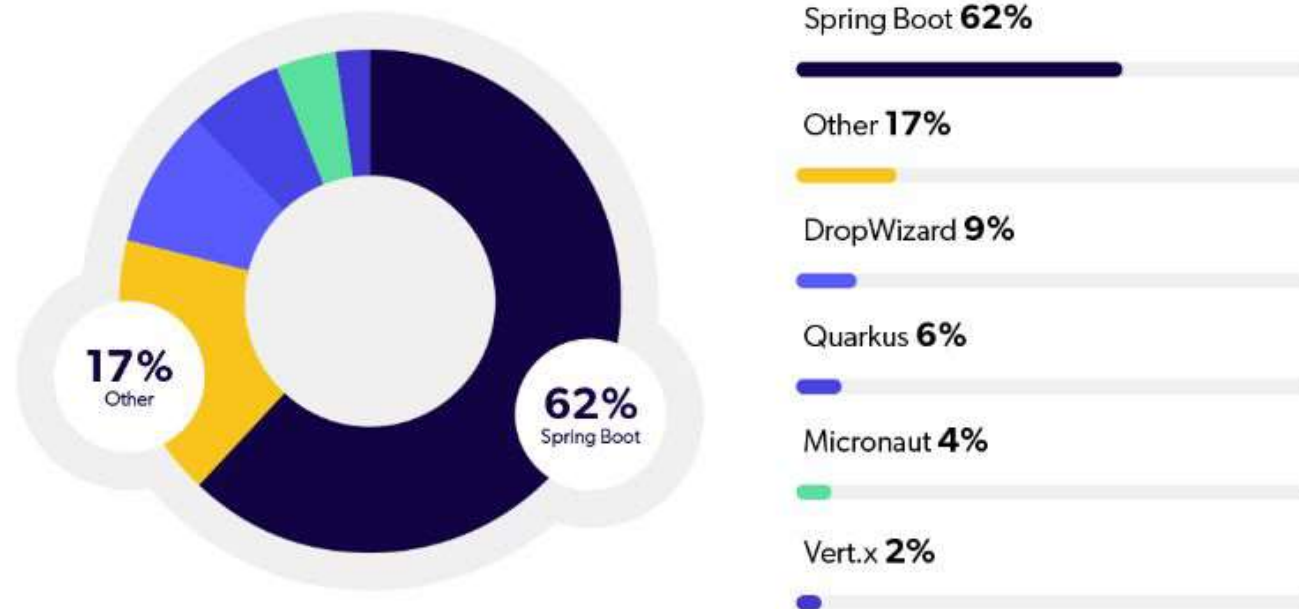
# Spring Boot

- ✓ Bắt đầu với Spring Boot một cách dễ dàng
- ✓ Tối thiểu hóa việc phải hiểu rõ cách cấu hình các tập tin
- ✓ Sử dụng dễ dàng với Maven hoặc Gradle
- ✓ Tránh xung đột giữa các dependency
- ✓ Tích hợp sẵn web server cụ hỗ trợ như Web server ...



# Spring Boot

<https://www.linkedin.com/pulse/spring-vs-world-comparing-boot-alternatives-omar-ismail/>



# Template Spring Boot

<https://start.spring.io/>

The current version of Spring Boot changes regularly. Just choose the latest release (but not snapshot).

Click on 'Add dependencies', type 'Web' in the search box, then click on the dependency 'Spring Web' to select it.

The screenshot shows the Spring Initializr web application interface. The header includes the Spring logo and the text 'spring initializr'. The main content area is divided into several sections:

- Project:** Includes checkboxes for 'Maven Project' (selected) and 'Gradle Project'.
- Language:** Includes checkboxes for 'Java' (selected), 'Kotlin', and 'Groovy'.
- Spring Boot:** Includes checkboxes for various versions: '2.3.0 M4', '2.3.0 (SNAPSHOT)', '2.2.7 (SNAPSHOT)', '2.2.6' (selected), '2.1.14 (SNAPSHOT)', and '2.1.13'.
- Project Metadata:** Includes input fields for 'Group' (com.example), 'Artifact' (demo), 'Name' (demo), 'Description' (Demo project for Spring Boot), and 'Package name' (com.example.demo).
- Packaging:** Includes checkboxes for '.jar' (selected) and 'War'.
- Java:** Includes checkboxes for '14', '11', and '8' (selected).
- Dependencies:** Includes a button 'ADD DEPENDENCIES...' and a list of dependencies. The 'Spring Web' dependency is selected, with a description: 'Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.'

At the bottom, there are three buttons: 'GENERATE', 'EXPLORE', and 'SHARE...'. The 'GENERATE' button has a keyboard shortcut '⌘ + ↵'.



# Ưu điểm của Spring Boot

Nhúng trực tiếp Tomcat, Jetty hoặc Undertow (không cần triển khai tệp WAR)

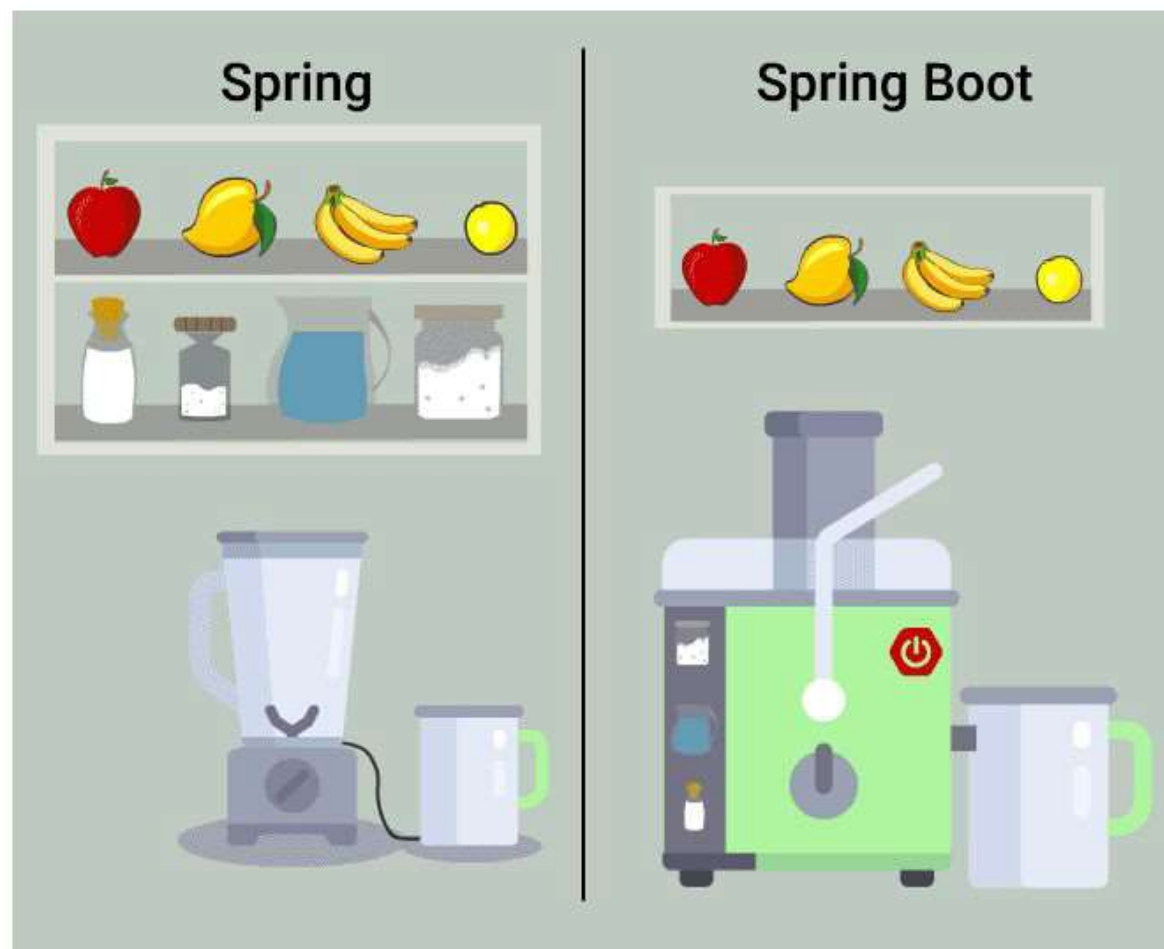




# Ưu điểm của Spring Boot

Spring Boot = Spring Framework + Application Container + Configuration

Source: <https://devrant.com>





TRƯỜNG ĐẠI HỌC  
**VĂN LANG**

KHOA CÔNG NGHỆ THÔNG TIN

