**I - OVERVIEW**

1. **Tổng quan**

* Là một framework phát triển các ứng dujgn Java
* Là mã nguồn mở
* Xây dựng trên 2 nguyên tác design chính là: Dependency Injection và Aspect Oriented Programming

1. **What does Spring framework provide**

* Application context and dependency injection
* Data access
* Hỗ trợ cho chúng ta quản lý các transaction, giúp chúng ta chỉ tập trung vào phát triển các business logic mà không cần phải lo lắng nhiều về tính toàn vẹn của dữ liệu.

NOTE: Xem kĩ video sau để hiểu hơn

LINK : <https://www.youtube.com/watch?v=gq4S-ovWVlM>

**II – ARCHITECTURE**

**LINK (15p00):** <https://www.youtube.com/watch?v=210tVT2uPvI>



* 1. **Tổng quan các tầng**
  2. *Test*

Tầng này cung cấp kha năng hỗ trợ kiểm thử với jUnit và TestNG

* 1. *Spring Core Container*

Bao gồm cái module spring core, beans, context và expression languate (EL)

* Spring core, bean cung cấp tính năng IOC và Dependence Injection
* Spring Context hỗ trợ đa ngôn ngữ (internationalization), các tính năng Java EE như EJB, JMX
* Expression Languege được mở rộng từ Expresion Language trong JSP. Nó cung cấp hỗ trợ việc setting/getting các giá trị, method cải tiến cho phép truy cập collections, index, các toán tử logic,…
  1. *AOP, Aspects và Instrumentation*

Những module này hỗ trợ cài đặt lập trình hướng khía cạnh (Aspect Oriented Programming), hỗ trợ tích hợp với AspectJ

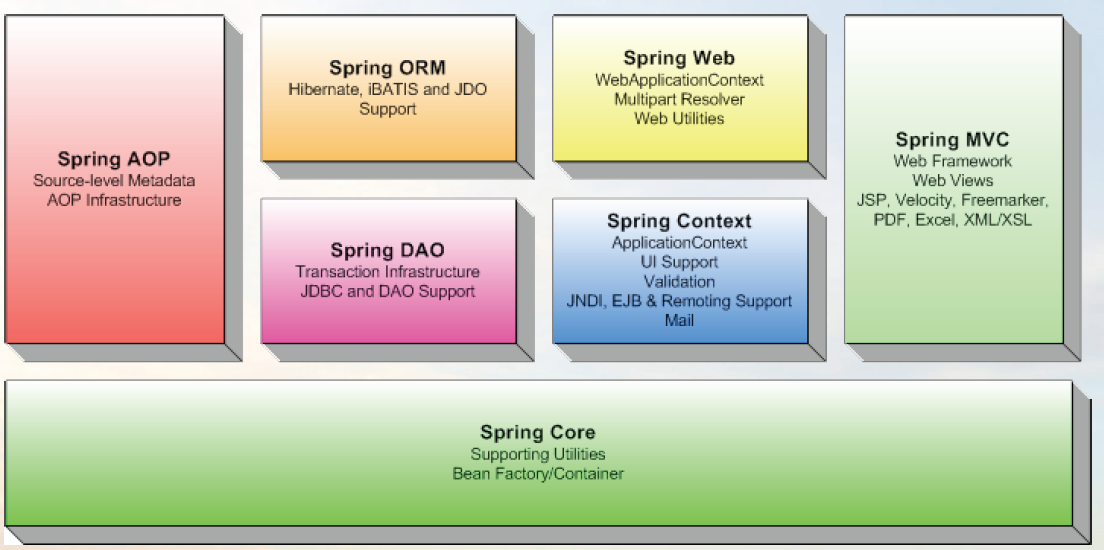
* 1. *Data Access/ Integration*

Nhóm này bao gồm JDBC, ORM, OXM, JMS và module Transaction. Nhwuxng module này cung cấp khả năng giao tiếp với database

* 1. *Web*

Đây còn gọi là Spring MVC. Nhóm này bao gồm Web, Web-Servlet,… hỗ trợ việc tạo úng dụng web.

1. **Module**



Spring framework có thể được xem như một kiến trúc chứa 7 module. Chức năng của mỗi thành phần như sau:  
 **- Core Container**  
Core container cung cấp chức năng cơ bản của Spring. Thành phần chính của nó là Bean Factory, một cài đặt của Factory pattern. BeanFactory áp dụng[IoC](https://itphutran.com/spring-framework/gioi-thieu-tong-quan-ve-spring-framework) pattern để đặc tả sự phụ thuộc từ code của ứng dụng.

* **Spring Context/Application Context**  
  Spring context là một file cấu hình để cung cấp thông tin ngữ cảnh của Spring. Spring context cung cấp các service như JNDI access, EJB integration, e-mail, internalization, validation, và scheduling functionality.

Cung cấp cách để truy cập các đối tượng

* **Spring AOP (Aspect-Oriented)**  
  Spring AOP module tích hợp chức năng lập trình hướng khía cạnh vào Spring framework thông qua cấu hình của nó. Spring AOP module cung cấp các dịch vụ quản lý giao dịch cho các đối tượng trong bất kỳ ứng dụng nào sử dụng Spring. Với Spring AOP chúng ta có thể tích hợp declarative transaction management vào trong ứng dụng mà không cần dựa vào EJB component.

Spring AOP module cũng đưa lập trình metadata vào trong Spring. Sử dụng cái này chúng ta có thể thêm annotation vào source code để hướng dẫn Spring nơi và làm thế nào để liên hệ với aspect.

**- Spring DAO**  
Tầng JDBC và DAO đưa ra một cây phân cấp exception để quản lý kết nối đến database, điều khiển exception và thông báo lỗi được ném bởi vendor của database. Tầng exception đơn giản điều khiển lỗi và giảm khối lượng code mà chúng ta cần viết như mở và đóng kết nối. Module này cũng cung cấp các dịch vụ quản lý giao dịch cho các đối tượng trong ứng dụng Spring.

**- Spring ORM**  
Spring có thể tích hợp với một vài ORM framework để cung cấp Object Relation tool bao gồm: JDO, Hibernate, OJB và iBatis SQL Maps.

**- Spring Web module**  
Nằm trên application context module, cung cấp context cho các ứng dụng web. Spring cũng hỗ trợ tích hợp với Struts, JSF và Webwork. Web module cũng làm giảm bớt các công việc điều khiển nhiều request và gắn các tham số của request vào các đối tượng domain.

**- Spring MVC Framework**  
MVC Framework thì cài đặt đầy đủ đặc tính của MVC pattern để xây dựng các ứng dụng Web. MVC framework thì cấu hình thông qua giao diện và chứa được một số kỹ thuật view bao gồm: JSP, Velocity, Tiles và generation of PDF và Excel file.

**III – SETUP AND ENVIROMENT**

*Step 1 - Setup Java Development Kit (JDK*)

## *Step 2 - Install Apache Common Logging API*

## Bạn có thể download version mới nhất của Apache Commons Logging API từ: [https://commons.apache.org/logging/](https://commons.apache.org/logging/download_logging.cgi)

## Common Logging API

## *Step 3 - Setup Eclipse IDE*

## Eclipse Home page

## *Step 4 - Setup Spring Framework Libraries*

## Bạn có thể download version mới nhất của Spring Framework Libraries từ: <https://repo.spring.io/release/org/springframework/spring>

## Spring Directories

**IV – EXAMPLE HELLO WORLD**

NOTE: Tuyến có upload demo trên Github, tự coi.

**V - KHÁC**

1. **Các khái niệm khác**
   1. Depencdency Injection

Dependency Injection là một mẫu thiết kế phần mềm mà các đối tượng phụ thuộc sẽ được inject vào một lớp nào đó. Dependency Injection là một implementation cụ thể của khái niệm Inversion of Control (đảo ngược điều khiển).

Ta sẽ có một ví dụ đơn giản về Dependency Injection như sau: Bạn có một web controller có nhiệm vụ lưu thông tin gửi từ form của người dùng. Theo nguyên lý Đơn trách nhiệm nói trên, bạn không muốn lớp controller tương tác với cơ sở dữ liệu. Thay vào đó, bạn sẽ sử dụng một lớp service để làm công việc này. Như vậy, controller của bạn sẽ chỉ phải xử lý dữ liệu của form (get form data, validate data, ...) rồi gọi một phưong thức của lớp service được inject để lưu dữ liệu. Controller không cần phải quan tâm hay lo lắng về kết nối cơ sở dữ liệu, pooling hay bảng nào sẽ được update. Cũng như service không cần phải biết request có những thông tin gì.

1. **Slider**



