CYDEO

SPRING CORE

Framework

What is Framework ?

An application framework is a set of common software functionalities that provides a foundation structure for developing an application.

An application framework eases the effort of writing an application by taking out the effort of writing all the program code from scratch.

All the possible components you can use to assemble any piece of wooden construction

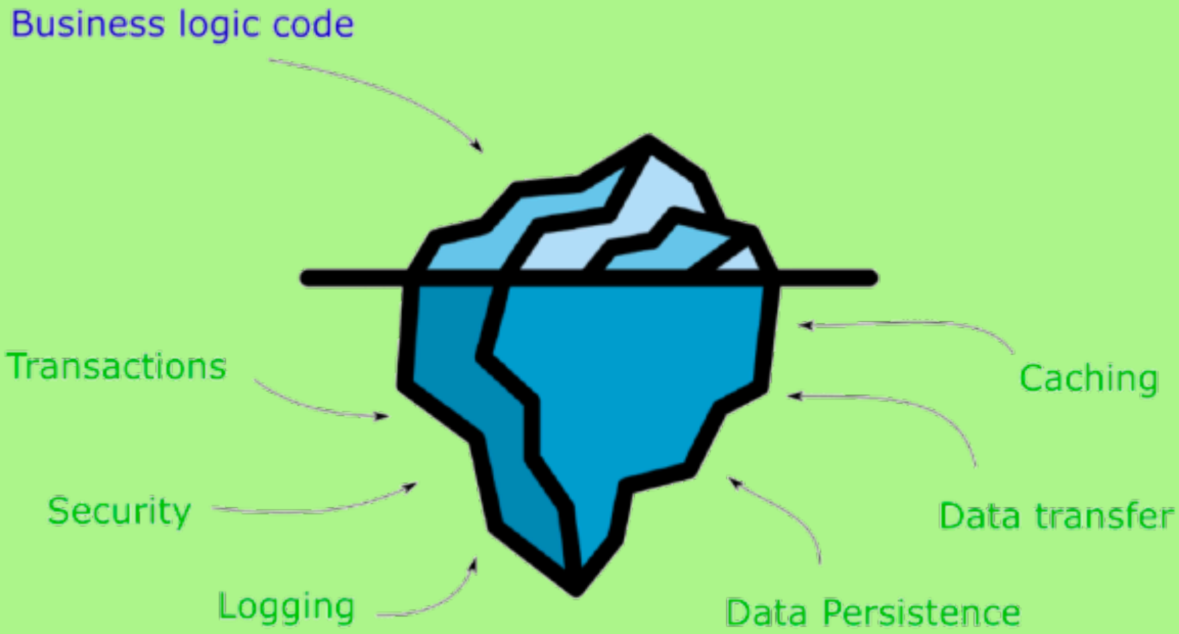
Choose the perfect components to assemble to get the right result

Ferris Wheel

Nightlight

Carousel Model

Common App Requirements


Business logic code
Transactions
Caching
Security
Data transfer
Logging
Data Persistence

Business Logic Code

Business logic code is what makes an application different from another from the functionality point of view.

Spring Ecosystem

We refer to Spring as a framework, but actually, Spring is much more complex.

Spring is an ecosystem of frameworks and not just one framework.

When developers refer to the Spring framework, they refer to a part of the software capabilities that include : Spring Core, Spring Data Access, Spring Model - View - Controller(MVC), Spring Testing

Spring Ecosystem

Spring ecosystem includes a big collection of other frameworks that integrate well one with the other and form together a larger universe.

Here we have projects like Spring Data, Spring Security, Spring Cloud, Spring Batch, Spring Boot, and so on.

When you develop an app, you can use more of these projects together.

These projects are being independently developed. They have a separate team that works on extending its capabilities.

Summary

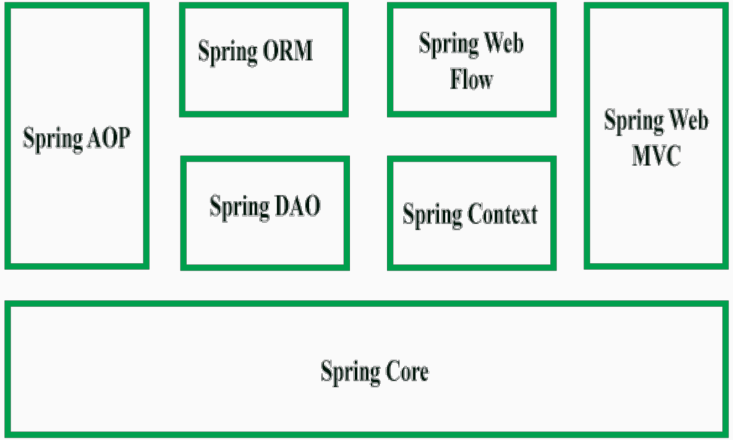
A framework is a set of common functionalities that provides a foundation structure for developing an application. It helps you build an app more efficiently by providing functionality that you just have to assemble to your implementation instead of developing yourself.

Spring framework opens a door to a large community. Having a large community using a specific piece of software makes it also more probable that others might face a similar problem with yours.

Spring is not just a framework. Spring framework indicates the core functionalities. In general, Spring offers an entire ecosystem formed of many projects which we use in application development.

Spring Core

Spring core is the part of the Spring framework providing the foundational mechanism used by Spring to integrate into apps.



Coupling

What is coupling in Java ?

Usage of an object by another object.

It simply means that one object requires another object to complete its assigned task.

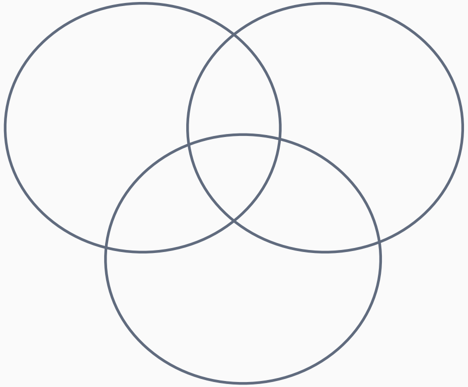
There are two types of coupling :

- Tight Coupling
- Loose Coupling

Tight Coupling

A group of classes are highly dependent on one another.

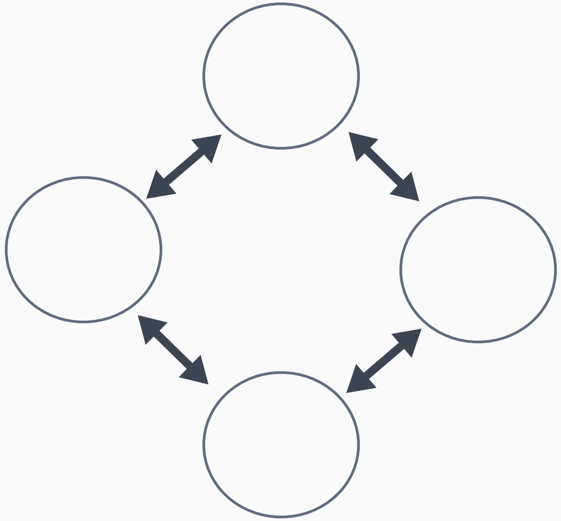
A class has a direct reference to a concrete class.



Loose Coupling

An object gets the object to be used from external sources.

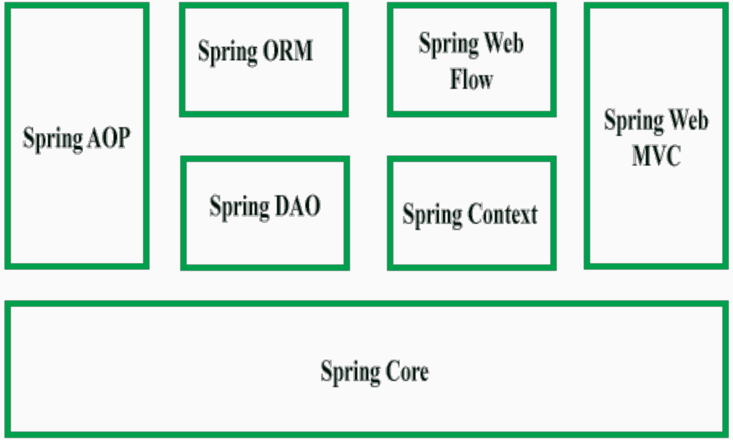
Reduces the dependencies of a class that uses the different classes directly.



Spring Core-IoC

Spring Core

Spring core is the part of the Spring framework providing the foundational mechanism used by Spring to integrate into apps.



IOC (Inversion of Control)

IoC is a principle that we do not leave the app control the execution by its own code and use dependencies. Instead, we allow the framework to control the app and its code.

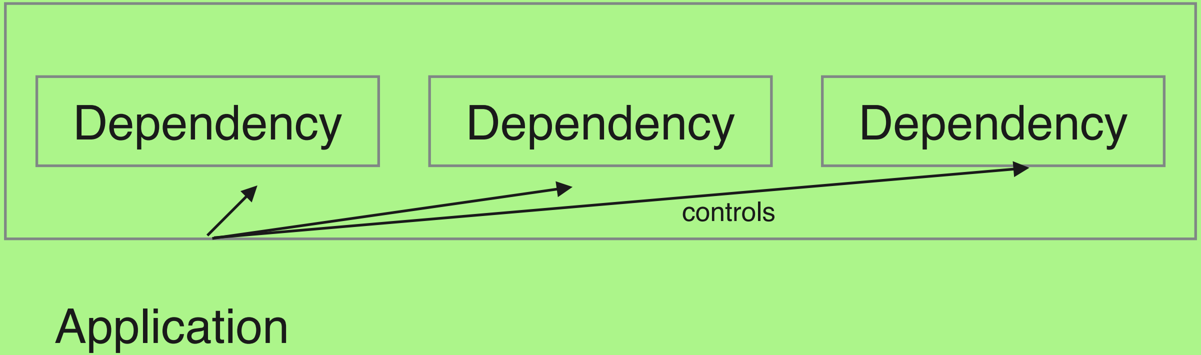
Advantages :

Decoupling the execution of a task from its implementation

Making it easier to switch between different implementation

Great ease in testing a program by isolating a component or mocking it dependencies

Without IoC



Dependency

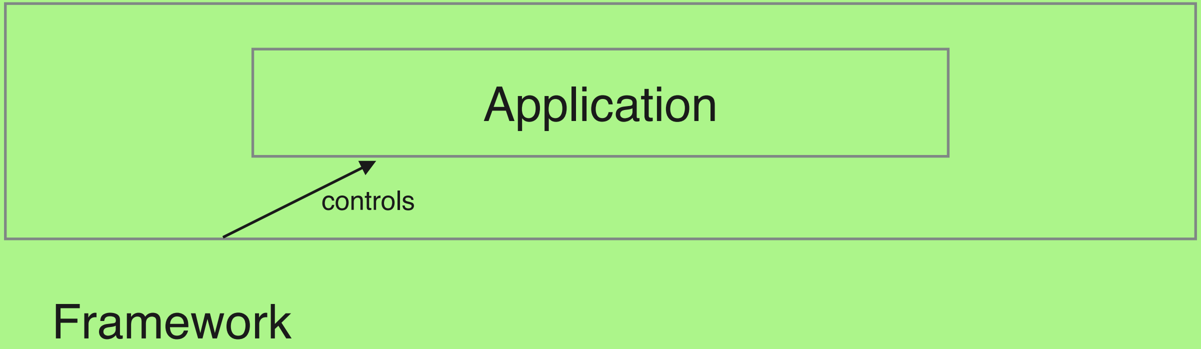
Dependency

Dependency

controls

Application

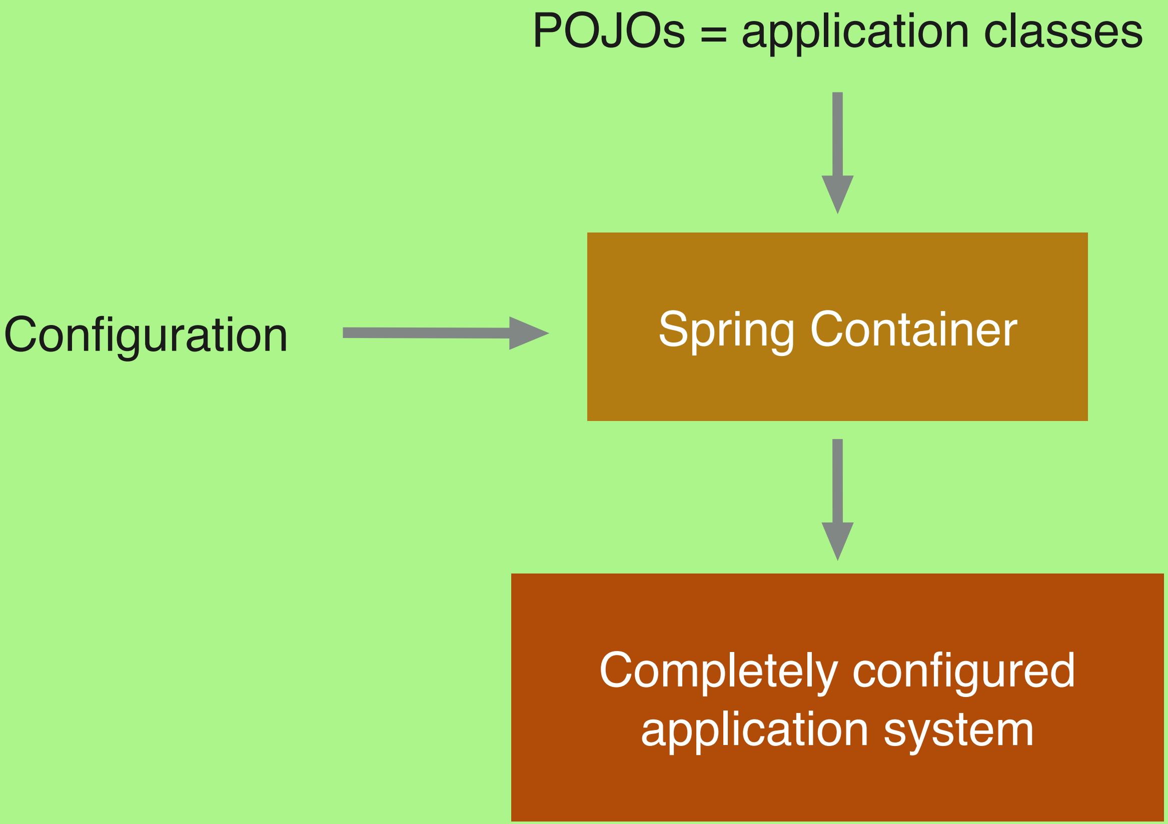
With IoC



Application

controls

Framework



POJOs = application classes

Configuration

Spring Container

Completely configured application system

Spring Container

IoC Container is tasked with the responsibility of connecting beans together to build a working application and it does so by reading a configuration provided by developer.

IoC container is external authority that passes a dependency to a dependent object that will use.

Having an external responsible for injecting dependencies allows very loosely coupled applications to built.

Spring Bean

Spring Bean is simply Java Object.

A bean is an object that is created an managed by Spring Container.

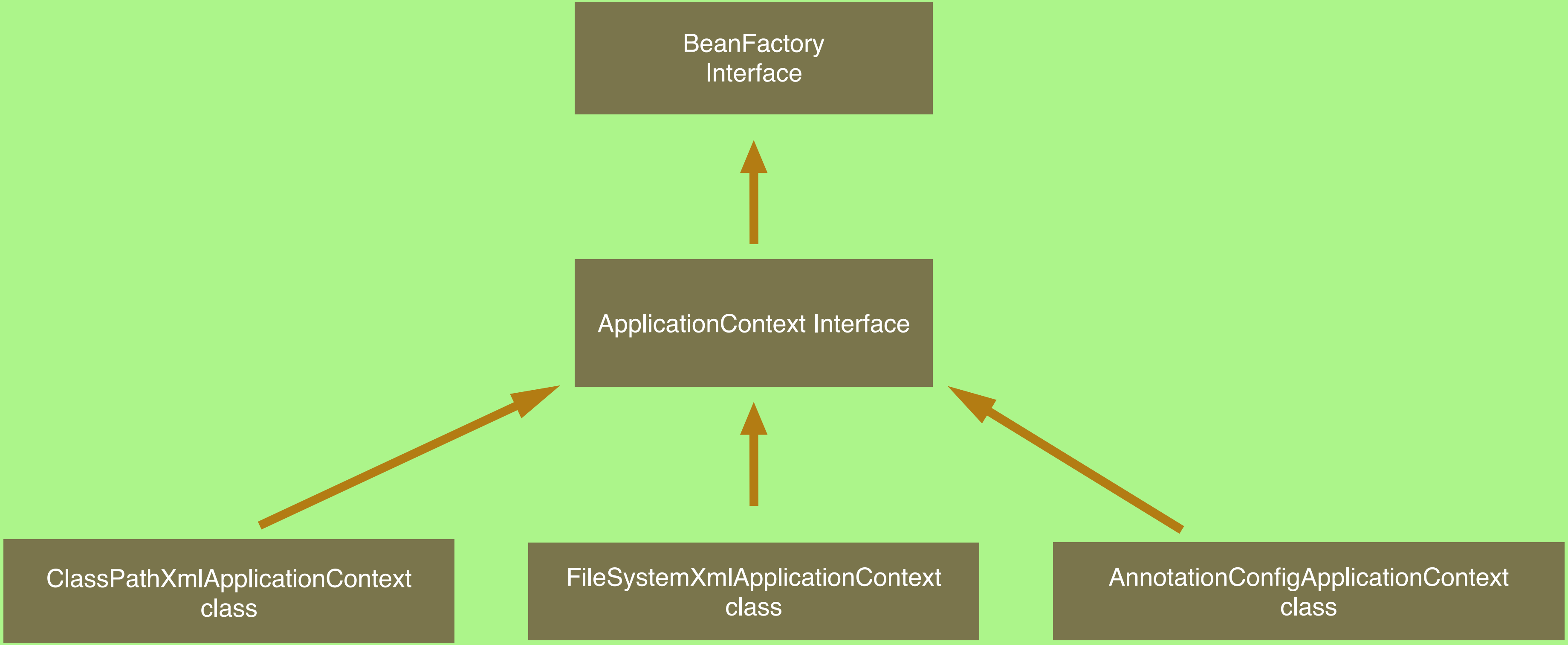
Spring beans are created from POJOs.

Types of Spring Container

There are 2 types of IoC containers :

Spring BeanFactory Container

Spring ApplicationContext Container



BeanFactory Interface

ApplicationContext Interface

ClassPathXmlApplicationContext class

FileSystemXmlApplicationContext class

AnnotationConfigApplicationContext class

```xml
<dependency>
   <groupId>org.springframework</groupId>
   <artifactId>spring-context</artifactId>
   <version>5.2.8.RELEASE</version>
</dependency>
```

Add spring-context dependency

```java
ApplicationContext container = new AnnotationConfigApplicationContext();
```

Creating an instance of the Spring context

What we did

```
@Getter
@Setter
public class Car {
    private String make;
}
```

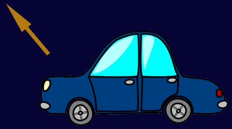
```
public static void main( String[] args ){

  Car c = new Car();

  ApplicationContext container =
       new AnnotationConfigApplicationContext();

}
```

We created a car instance,
but it is not in the Spring Context



Spring Context

We created the container but
It is now empty

Adding new beans to the Spring Context

There are multiple ways to add beans in the Spring context

- Using the @Bean annotation
- Using stereotype annotations

Using the @Bean Annotation

Define a configuration class (annotated with @Configuration).

Add to the configuration class a method that returns the object instance that you want to add to the context and annotate the method with the @Bean annotation.

Make Spring use the configuration class defined in the first step.

@Primary

A primary bean is the one that Spring will choose if it has multiple options, and you do not specify a name.

The primary bean is simply the default choice of Spring.

Dependency Injection

Dependency Injection

Dependency Injection is a fundamental aspect of the Spring framework, through which the Spring container "injects" objects into other objects or “dependencies”.

Spring achieves DI by wiring the beans together.

Spring Context
Add person and car beans

Spring Context
Make the person own the car

Implementing Relationships Among Beans

There are 2 ways in which we can establish the relationship among beans:

Wiring

Autowiring