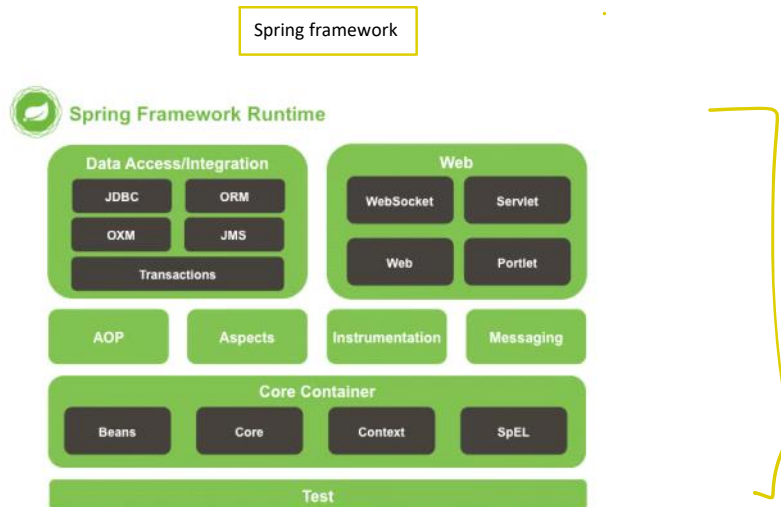


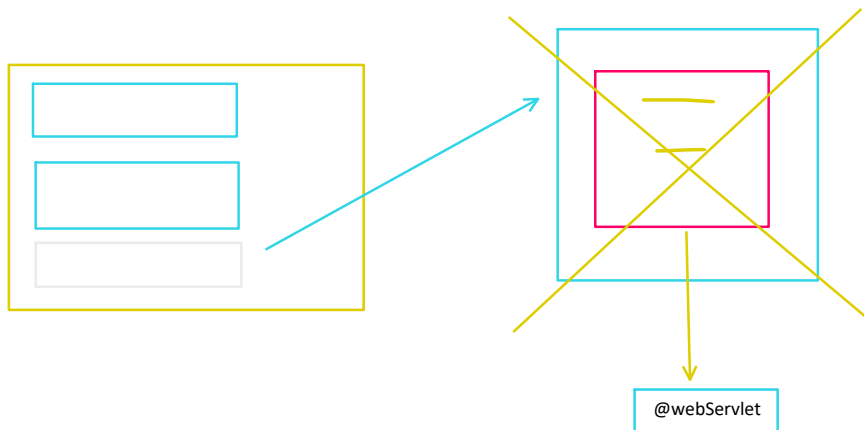
Spring core

20 July 2025 09:03



Framework:

1. semi developed software which provides common logic required for project development
2. Framework will help developers to implement more functionality in less time
3. When we use framework to develop project, we can focus only on **business logic**



Type of frameworks:

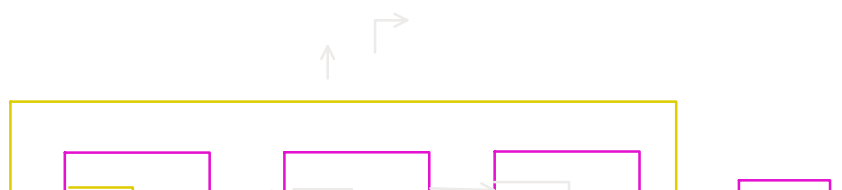
1. Frontend framework : Angular
2. Web framework : Struts (Outdated)
3. ORM framework : Hibernate -> Spring data JPA

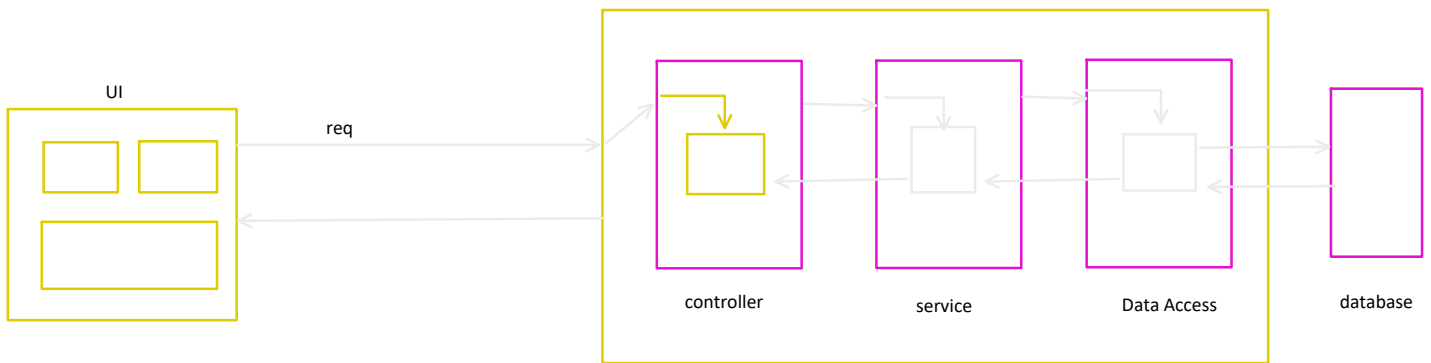
By using struts we can develop only **web layer** in the project (Controllers)

By using JPA we develop only **Data Access layer** (persistence layer)

Architecture Application

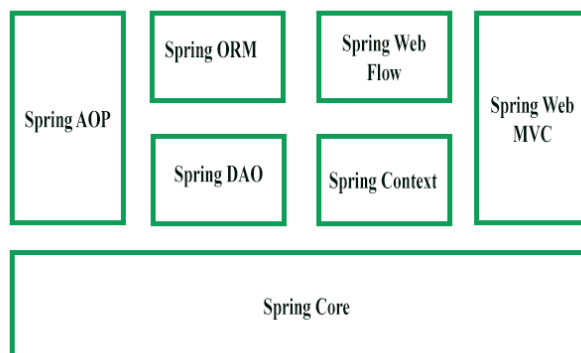
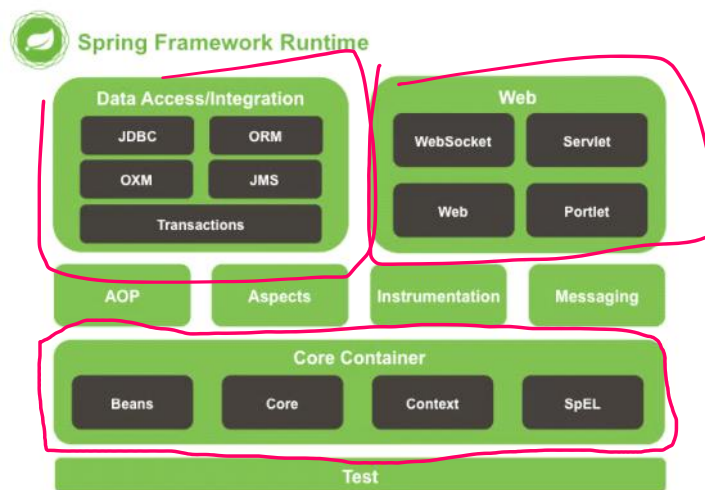
UI



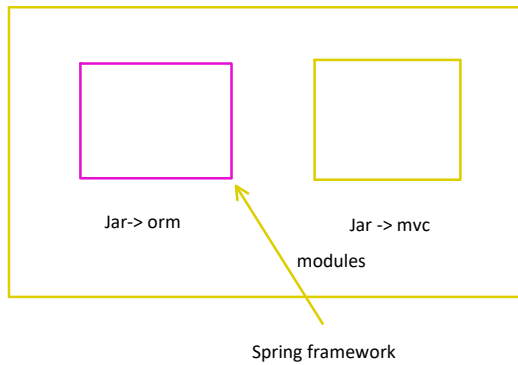


Note: to overcome the problem of struct framework, spring framework came into market

1. Spring f/w is called as application development framework
2. By using spring f/w we can develop end to end application
3. Spring f/w is free and open source
4. Spring f/w developed in modular fashion because this modular development it is loosely coupled.



Note : we just need spring core knowledge to work on spring f/w



Spring Modules

1. Spring core
2. Spring Context
3. Spring JDBC
4. Spring ORM
5. Spring AOP
6. Spring Web MVC
7. Spring Security
8. Spring Batch
9. Spring Data JPA
10. Spring Rest
11. Spring Cloud
12. Spring AI

Note : Spring is very flexible framework. It will not force to use all modules. Based on requirement we can pick up particular module and we can use it.

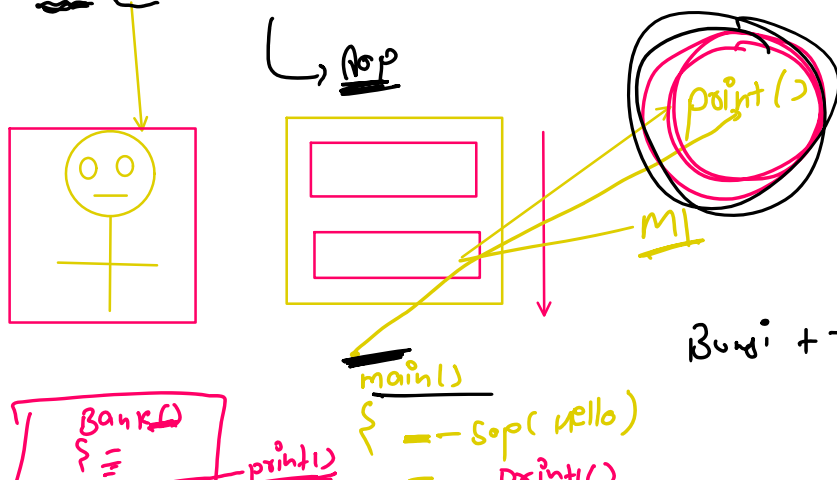
1. Spring is versatile framework (easily it can be integrated with other framework)
2. The current version of spring framework is 6.x
3. Spring f/w is under licences o VM ware Tanza
 - a. URL : www.spring.io

1. Spring Core: It is base module in the spring f/w

- a. Spring core module providing fundamental concepts of spring f/w
 - i. IOC container (Inversion of control)
 - ii. Dependency injection
 - iii. Bean life cycle
 - iv. Bean Scope
 - v. Autowiring etc....

2. Spring Context : it will deal with configuration required for our spring application.

3. Spring AOP : Aspect Oriented programming





Note: if we combine business logic & secondary logic then will face Maintenance issue of our project. So we use AOP to separate business logic and secondary logic.

4. Spring JDBC : Spring jdbc is used to simplify Database communication logic

In java jdbc we need to write boiler plate code (repeated code) like below in several classes

1. LOAD DRIVER
2. GET CONNECTION
3. CRETAE CONNECTION
4. EXECUTE QUERY
5. CLOSE CONNECTION

=> Using Spring JDBC we can directly execute query the remaining part spring JDBC will take care

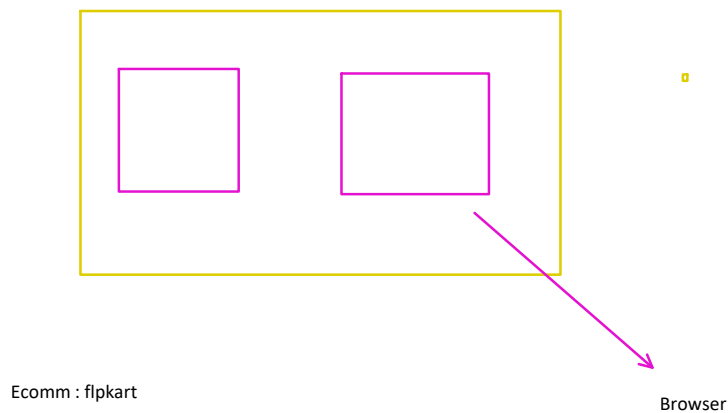
5. Spring Web MVC : it is used to developer both web Application & Distributed App

=> Web application: (C 2 B)

Ex: Gmail.com, facebook.com

=> Distributed Application (B 2 B) / web services or restful Services

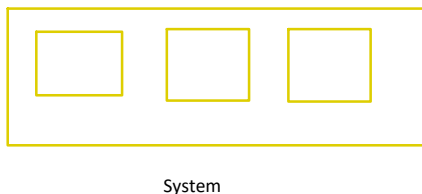
q



Distributed app:

A distributed app is a system where components are spread across multiple nodes but work together as single system.

Zomato:



Teck stack :
Microservices
Kafka
Load balancer

Web annlication

Distributed ann

Web application

1 server
End user uses this via UI

Distributed app

Multiple server
Services talk to each other



6. Spring ORM : Object relational mapping
=> spring f/w having integration with ORM frameworks
Eg: Spring ORM, Spring Data JPA

Note: JDBC will represent data in text format where as hibernate ORM will represent data in Objects format

```
executeQuery(query)
```

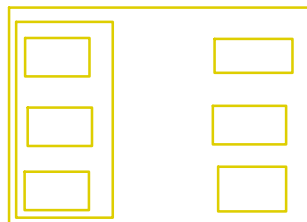
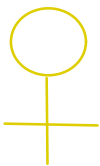
```
While(rs.hasNext()){
```

```
Rs.getString(name)  
Rs.getInt(id)
```

```
-----  
Student s = JPA.get(); => Object
```

7. Spring Security:
- a. Security is very important for every application
 - b. Using Spring security we can implement Authentication & Authorization
 - c. Spring security with Oauth 2.0
 - d. Spring Security with JWT (JSON web token)

8. Spring Batch : Batch means bulk operation
- a. Reading data from excel and store it into database table
 - b. Sending Monthly statement to customer in email
 - c. Sending Reminders to customers as Bulk SMS.



10. Spring Test : It provides unit test framework.

Note: Spring core -> it is all about managing dependencies among the classes with loose coupling

In project we will develop several classes. All those classes we can categorize into 3 types

- 1. POJO
- 2. Java Bean
- 3. Component

What is POJO ?

=> POJO : plain old java object

=> Any java class which can be compiled by using only JDK software is called a POJO class

Ex: 1

Class Demo {

```
    Int id ;
    String name ;
}
```

Ex 2:

Class Demo2 extends Thread {

```
    Int id ;
    String name ;
}
```

Ex 3:

Class Demo3 implements Runnable {

```
    Int id;
    String name;

    //run method
    Run()
    {
    }
}
```

Ex: 4 yes

Class Demo3 extends GenericServlet {

```
    Int id;
    String name;

    //run method
    Run()
    {
    }
    Service()
    {
    }
}
```

What is java beans

=> Any java class which follows beans specification rules is called as Java Bean.

1. Class should implement serializable interface
2. Class should have private data member (variables)
3. Every private variable should have public setter and public getter method
4. Class should have zero-param constructor

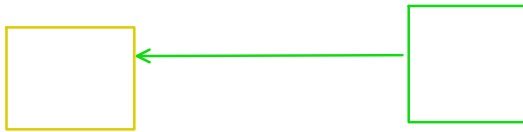
Note : Bean classes are used to write business logic and to store and retrieve data

What is component ?

The java classes which contains business logic is called as componet classes

Eg: controller, service

Tight Coupling



```

1 public class Car {
2     1 usage
3     PertolEngine engine = new PertolEngine();
4     no usages new *
5     public void drive()
6     {
7         engine.start();
8         System.out.println("car is moving...");
9     }
10 }

```

Car is tightly coupled with Engine class
We can't change engine type
We can't test the car without Engine
No flexibility

```

public class Car {
    //PertolEngine engine = new PertolEngine();

    1 usage
    Engine engine = new DieselEngine();
    no usages new *
    public void drive()
    {
        engine.start();
        System.out.println("car is moving...");
    }
}

```

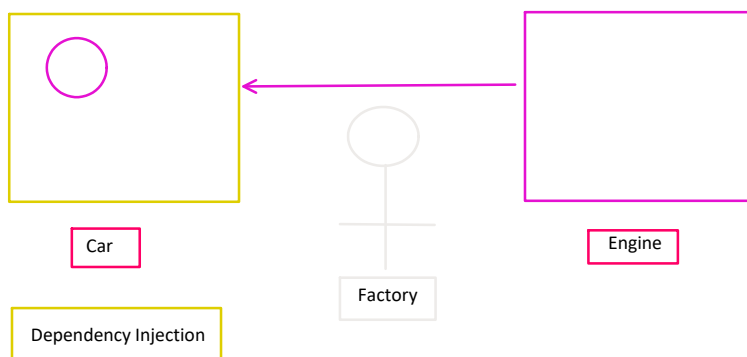
Flexible but still poor
We are using interface but still we have make object hardcoded `new DieselEngine();`

Without Factory

1. Car create engine
2. Tightly Coupled
3. Hard test
4. No Flexibility

With Factory

1. Engine is created by factory
2. Loosely Coupled
3. Easy test
4. Flexibility



1. Loosely coupled
2. Dependency Injection

