

IOC Effectiveness through Spring Framework

Prajapati Keyur P.

Assistant Professor

Department of Computer Engineering
MBICT, New V.V.Nagar 388120, India

Abstract—Today in the world less code is becoming a primary key. To keep this thing in mind, idea belonging to add some flavor to existing framework which will leverage all the authenticity with all things what developer need without touch their real code. Just sat things once, will take care for everything with less effort.

Keywords—Spring, IOC, Data Dependency, Bean Wiring

I. INTRODUCTION

The world is moving on highly tech edge. With respect to all programming languages a lot frameworks are there. Java is one of the famous and robust language right now in context of tech era. With its pure object oriented base, it won all most every area of computer informative world.

Based on this particular language competition is there, to create a framework and give full supports to its every part. I am talking specifically about spring framework. And I found some inclusive research concepts for the same.

Before that just look at its basic fundamentals through its own wiring concepts.

Spring comes with two primary key features. Dependency Injection and Inversion of control.

With its two features developers work will automatically reduce. Dependency injection give you control to define beans properties with its ancestors, while Inversion Control will set everything through what you did in objects dependencies.

After all Dependency Injection is a one great flavor of IOC. And its spring framework's heart.

Here I am going to explore all fundamental concepts of this spring framework with my research idea of object injection in the era of this framework.

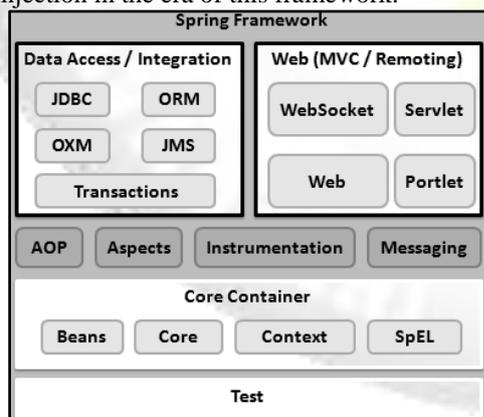


Fig. 1:

Look at its first part about database Access/Integration. It gives fully support to data access through your front end scripting language java.

II. RESEARCH CONCEPTS

Herewith I would like introduce a mix flavor of Spring Framework with ORM (Object Relation Mapping/Hibernate).

Spring gives full fledged with Data Access through JDBC Scripting. But when we are talking about web mining or any web redundancy, we have to need the small entities in a one form. I am informing you that we have to need a pocket to keep all the data inside it without any worry about the loss and safety about data. Just kick the thing and don't worry where it has to stay. Yup ORM is there as a solution.

Research is based on how it's good when we give our wiring to all the objects by default and framework will take care of it. I found there is no complete integration of your object data to spring framework, which will automatically take care about its persistence. With the help of configuration files we can just add some Meta information about objects and its dependencies. If we need to auto wire with its parent object then also we can give it.

My means of this research is give all metal information for a particular object through mapping resource and finally everything will be taken care through this framework.

In figure we can see the modeling part of messaging through beans is also resolved by these concepts.

For the same we have to need POJO class with its GET/POST (Setter / getter Methods). By annotation we also give its dependencies there.

In last core container we are responsible to give proper valid data in we configuration files and that's all.

Spring is a very good framework to create your own POJO classes and objects. In the same era we are adding an ORM informative data through our own configuration files. So at the end you will have to sit with a cup of coffee and watch your favorite entertainment.

At last but not least we can think about transaction features through either annotation or configuration for the objects what we have created for our any application. It will affect the whole application smoothly and reduce lethargic work. Initially we have to think data, classes and objects with all dependencies and need some hard work with some of files. But later on it will give you advantages at its own scalability.

REFERENCES

- [1] <https://docs.spring.io/spring/docs/4.0.x/spring-framework-reference/htmlsingle/#beans-factory-metadata>
- [2] <https://docs.spring.io/spring/docs/4.0.x/spring-framework-reference/htmlsingle/#beans-factory-collaborators>
- [3] <https://docs.spring.io/spring/docs/4.0.x/spring-framework-reference/htmlsingle/#beans-ctor-injection>
- [4] <https://docs.spring.io/spring/docs/4.0.x/spring-framework-reference/htmlsingle/#beans-factory-scopes-singleton>