INTRODUCTION TO SPRING FRAMEWORK

INTRODUCTION :

* What is the hype around the Spring Framework?
* In this course we will be learning about developing with Spring and we will use the various configuration methods to resolve all the common problems that we face in software development.
* We will develop an partial application without Spring and then we will add XML Configuration , Java Configuration and finally Annotations based on Spring.

WHAT IS SPRING :

* The Spring framework started out as an Inversion of Control Container.
* It was brought in to replace some complex configurations that was present in the Java J2EE Framework.
* Spring was later developed to using Java without EJB’s . Its original aspect was to how to work better with EJB’s but then they kind of realized that they didn’t need that for a lot of situations.
* So it became like how do I do the same development without EJB’s.
* So now Spring Framework is used without EJB’s.
* Spring enables us to application development without using an Application server.
* They don’t know that Tomcat is not an application server , it’s a web server.
* Its one of the reasons why Tomcat has taken over the Java Development Standard Container.
* Its easy to use , its lightweight.
* Spring is completely POJO Based.
* Any code you write in Spring can be written without Spring.
* POJO is defined as a Plain Old Java Object , the Spring framework doesn’t do much behind the scenes , but it just makes us write cleaner code and is POJO based and Interface Driven.
* Spring was built out of the frustrations of J2EE.
* So its really unobtrusive , it shouldn’t be getting in your way. If it is , then you are probably using it wrong.
* Spring also uses AOP/Proxies so that you can apply transactions to your code to get those cross cutting concerns out of your code.
* So actually your code should be smaller and lightweight from using Spring.
* What are the most appealing parts of Spring is that its built around BEST PRACTICES. We end up having design patterns in our code without really knowing them. Things like Singleton , Factories , Abstract Factories , all those best practices are built in to our code inherently just by using Spring.
* Spring uses the Template method pattern a lot , although its not a pattern , it’s a design based methodology or best practice that’s annotation based configuration.

UPDATE :

* This course on pluralsight has been updated since its initial release , we have used MAVEN to do the builds in our code.

THE PROBLEM :

* What problem is Spring trying to solve?
* It increases our testability.
* It increases our maintainability or the ability to maintain our code.
* It also helps us with scalability. It decouples things and helps us from adding caching and other things to our code without having to rewrite our code base.
* It also helps us to reduce the complexity of our code.
* It puts the focus on the business and the business doesn’t care what framework Im using but just getting the code and the job done.

BUSINESS FOCUS :

* If we have used Java then we would have used JDBC.
* JDBC at first glance doesn’t look that bad.
* When we start to analyse then we get to know that its not that pretty.
* Look at all the extra stuff we have in our code , Connections , PreparedStatements , ResultSets , try-catch-finally block and DriverManager.
* When all our business cares about is just the SELECT statement and the ResultSet with the WHILE LOOP.

THE SOLUTION :

* The solution that we are trying to get out of Spring or using the Spring Framework.
* That we can remove the configuration code or the lookup code and the developers can focus of the business needs.
* The business doesn’t care whether I have a try-catch block or a series of connection statements.
* It cares for when we ask for a car we get that car and something like that.
* Our code can also focus on testing.
* Spring also enables us from doing Annotation Based or XML Based application development.

BUSINESS FOCUS REVISITED :

* Our previous code contained DriverManager , Connection , Statement , PreparedStatement and ResultSet , Try-Catch-Finally block that does nothing but catch exceptions and close connections.
* Now lets look at how the code could be using the Spring Framework. (Business Focus Revisited – I-SCREENSHOT)
* The Spring JDBC Template which is a module of Spring is enabling us to write simpler code to retrieve , update , insert and delete values from the database.
* Nowhere we talk about opening and closing a connection or the usage of a DriverManager or a try-catch-finally block.
* We have a findCar() method and a find() method from the getEntityManager() method and a return statement to find the car based on a ID.
* This Is an example of the Template Method that does a lot to help us clean our code.
* This may scare some of you that there is a BLACK-BOX with Spring , but look at how smaller and condensed our code is , it really is just doing what we want it to.

HOW IT WORKS :

* As we have seen , everything in SPRING is a simple POJO . Spring itself can be thought of as a glorified HashMap.
* Spring can also be used as a registry , we are going to use a simple main method in our application to run it.
* We really get benefits out of Spring by using its wiring constructs or the AutoWiring.
* The Spring container – Imagine a number of squares all linked up together
* Consider those squares as Beans and you can see their references where some of them are standalone , some are referencing other beans. How they are all wired up and how we are using the objects together.
* Let us consider the JDBC Example :
* One of those squares could be a Statement , One of them could be a PreparedStatement , One of them might be a connection behind the scenes or one could be a EntityManager of some sorts.
* All of our beans are stored in that container and we access them out of that container and that’s where that HashMap comes into play.

SUMMARY :

* In this first module we looked at What is Spring.
* The problems that we are trying to solve.
* We also took a look at what the business focus is and what the solution is by using Spring.