Section 3- Part 2: What is @EnableAutoConfiguration?

This annotation automatically configures the your application based on the dependencies present in your project.

🌿Spring boot looks at what are the dependencies that we have in pom.xml file such as spring-boot-starter-web, spring-boot-starter-jpa etc.

🌿This annotation sets up the automatic configuration for all the dependencies services which comes as the part of these dependencies.

🌿For example web dependency consists of the web server, spring mvc etc. so all these dependencies are automatically managed and let the respective application classes to use it as per the need.

🌿In previous section we have seen that we define the manual configuration by defining the manual configuration class. That class is annotated by the @oConfiguration. This class has beans in it which are nothing but the objects. And spring helps us to inject this objects of manual configuration wherever it is needed. So everything was for the manual configuration basis.

🌿Here in @EnableAutoConfiguration what happens is that all the default configuration is scanned first. The beans of all these default dependencies are injected where ever it is necessary. Yes, now you are getting it, Beans are the most necessary part of this whole spring boot framework. Even it is the manual configuration it needs beans and even if it the automatic configuration it still needs beans to implement.

🌿Now that we know that @EnableAutoConfiguration configures the dependencies and manages the beans for the default settings let see how it actually works from the closer perceptive.

💡 1️⃣Scans the Class Path first

2️⃣Check which libraries are present in the class path (Jar Files).

3️⃣Automatically configures the setting based on the detected dependencies.

4️⃣Registers the default beans for all the dependencies without any need of extra configuration.