If one has to start a new spring project we have to add build path or add maven dependencies, configure application server, add spring configuration .

Using spring boot we avoid all the boilerplate code and configurations that we had to do previously.

Features of Spring boot-

* **Auto-Configuration** - No need to manually configure dispatcher servlet, static resource mappings, property source loader, message converters etc.
* **Dependency Management** - The different versions of commonly used libraries are pre-selected and grouped in different starter POMs that we can include in your project. By selecting one Spring Boot version we are implicitly selecting dozens of dependencies that we would have to otherwise select and harmonize ourself. Example-
* **Advanced Externalized Configuration** - There is a large list of bean properties that can be configured through application.properties file without touching java or xml config.
* **Production support**- We get health checking, application and jvm metrics, jmx via http and a few more things for free.
* **Runnable Jars** - We can package your application as a runnable jar with embedded tomcat included so it presents a self-contained deployment unit

What are advantages of Spring Boot?

The advantages of Spring Boot are

* Reduce Developement, Testing time and efforts.
* Use of JavaConfig helps avoid usage of XML.
* Avoid lots of maven imports and the various version conflicts.
* Provide Opinionated Development approach.
* Quick start to development by providing defaults.
* No Separate Web Server Needed.Which means that you no longer have to boot up Tomcat, Glassfish, or anything else.
* Requires less configuration-Since there is no web.xml file. Simply add classes annotated with@Configuration and then you can add methods annotated with@Bean, and Spring will automagically load up the object and manage it like it always has. You can even add @Autowired to the bean method to have Spring autowire in dependencies needed for the bean.
* Environment Based Configuration-Using these properties, you can pass into the application which environment you are using with:-Dspring.profiles.active={enviornment}. Spring will then load up the subsequent application properties file at (application-{environment}.properties) after loading up the main application properties file.