Spring Boot Roadmap

Before learning Spring Boot, it is recommended to have a solid understanding of the following Java web concepts:

1. Java programming language:

Ensure you are familiar with the fundamentals of the Java programming language, including object-oriented programming (OOP) concepts, data types, variables, control structures, and basic syntax.

2. Servlets:

Servlets are the foundation of Java web development. They provide a way to handle HTTP requests and generate HTTP responses. Learn about the Servlet API, request handling, response generation, and how to work with Servlet containers like Tomcat or Jetty.

3. JavaServer Pages (JSP):

JSPs are server-side templating technologies that allow embedding Java code within HTML files. Familiarize yourself with JSP syntax, scriptlets, expression language, implicit objects, and integrating Java code with JSP views.

4. JavaServer Faces (JSF):

JSF is a component-based web framework for building user interfaces. Learn about the JSF lifecycle, managed beans, navigation handling, form processing, and view templating using JSF tags.

5. Java Database Connectivity (JDBC):

Since Spring Boot integrates seamlessly with databases, it's essential to have a good understanding of JDBC, which is the standard Java API for interacting with relational databases. Learn about establishing connections, executing SQL queries, and handling result sets.

6. Object-Relational Mapping (ORM):
Familiarize yourself with ORM frameworks like Hibernate or JPA (Java Persistence API). These frameworks simplify database operations by mapping Java objects to database tables, enabling you to work with entities, relationships, and perform CRUD operations.

7. Dependency Injection (DI):
Spring Boot uses dependency injection extensively. Learn about the principles and patterns related to DI, as well as understanding how to inject dependencies using frameworks like Spring.

8. Inversion of Control (IoC):
Understanding IoC is closely related to DI. Learn about the concept of IoC, where the control of application flow is handed over to a container/framework, such as Spring Boot.

9. RESTful Web Services:
REST (Representational State Transfer) is a popular architectural style for building scalable web services. Familiarize yourself with REST principles, HTTP methods, request/response formats, and API design best practices.

10. Maven or Gradle:
Spring Boot projects typically use build tools like Maven or Gradle. Learn to configure and manage dependencies, handle project builds, and manage configurations using these build tools.

The above concepts provide a solid foundation for learning Spring Boot and will help you understand the underlying technologies and principles that Spring Boot builds upon