multiverse

Contents

Name:
Date:

Spring Boot

> It is recommended that you do this Learning Plan in combination with the Java Learning Plan as knowledge of Java is essential for using Spring Boot.

Spring Boot is a powerful framework built on top of the Java programming language, designed to simplify and accelerate the development of Java-based applications. It provides a comprehensive set of tools, libraries, and conventions that aim to streamline the process of building enterprise-level applications. Spring Boot embraces the “convention over configuration” principle, reducing the need for manual setup and configuration, and allowing developers to focus on writing business logic.

One of the key features of Spring Boot is its embedded server, which eliminates the need for deploying applications to a separate application server. With this approach, developers can create standalone, self-contained applications that can be easily deployed and executed with minimal effort. Spring Boot also offers a powerful dependency management system, which automatically manages the dependencies required by the application, ensuring compatibility and simplifying the integration of various components.

Additionally, Spring Boot promotes a modular and flexible architecture by providing a wide range of pre-built modules, known as “starters.” These starters encapsulate common functionalities and integrate seamlessly with Spring Boot, enabling developers to quickly add features like security, database access, messaging, and more to their applications. Furthermore, Spring Boot supports the creation of microservices, allowing developers to build scalable and distributed systems by leveraging its robust ecosystem and integrations with cloud platforms.

Spring Boot is a developer-friendly framework that simplifies Java application development by providing an opinionated approach, auto-configuration, and a comprehensive set of tools and libraries. It allows developers to focus on writing application logic rather than dealing with boilerplate code and complex configurations, making it an excellent choice for building robust and scalable Java applications.

Learning Plan Tasks

1. Introduction to Spring Boot
2. Spring Boot Basics
3. Working with Spring Boot
4. Data Persistence with Spring Boot
5. Spring Boot Advanced Topics
6. Build Something with Spring Boot!

#checkoutTheDocs

- **Spring**: Documentation
- **Spring**: Quick Start
- **Spring**: Spring Boot Guides
- **Spring**: Courses
- **Spring Boot Reference Guide**
- **Spring Boot in Action**

- **Java Brains**: Spring Boot Tutorial
- **Tech Primers**: Spring Boot Primers

  Many of the tasks below are covered in the extensive Spring Boot Guides. Feel free to use these to complete many of the objectives outlined below!

1. Introduction to Spring Boot

This section covers the following topics:

- What is Spring Boot and why it is important
- Benefits and features of Spring Boot
- Spring Boot architecture and its components
- Setting up the development environment

Tasks

1. What is Spring Boot and why it is important
   - Research and read articles on what Spring Boot is and its advantages.
   - Watch video tutorials that explain what Spring Boot is and its use cases.
   - Compare Spring Boot with other similar frameworks like Dropwizard or Micronaut to understand the differences and similarities.
2. Benefits and features of Spring Boot
   - Research and read articles on the benefits and features of Spring Boot.
   - Try to identify the most useful features of Spring Boot in your own project or use case.
   - Write a short code snippet using one of the features you find most useful.
3. Spring Boot architecture and its components
   - Research and read articles on the Spring Boot architecture and its components.
   - Draw a diagram or flowchart that shows the architecture of a Spring Boot application.
   - Identify the different components of a Spring Boot application, and explain how they interact with each other.
4. Setting up the development environment
   - Download and install the latest version of Spring Boot.
   - Install and configure a Java IDE such as Eclipse or IntelliJ IDEA.
   - Create a simple “Hello, World!” application using Spring Boot and run it on your local machine using the Spring Boot Guides.

2. Spring Boot Basics

This section covers the following topics:

- Creating a simple Spring Boot application
- Understanding the basic structure of a Spring Boot project
- Building and running Spring Boot application
- Using the Spring Initializr to bootstrap your application

Tasks

1. Creating a simple Spring Boot application
   - Create a new Spring Boot project using the Spring Initializr.
   - Create a simple "Hello, World!" application using Spring Boot, and run it on your local machine.
   - Modify the application to include a basic REST endpoint that returns a simple JSON response.
2. Understanding the basic structure of a Spring Boot project
   - Explore the different files and directories in a Spring Boot project.
   - Understand the role of each file, such as `application.properties`, `pom.xml`, and `Application.java`.
   - Experiment with modifying the contents of each file to see how it affects the application.
3. Building and running Spring Boot application
   - Build the Spring Boot project using Maven or Gradle.
   - Run the application using the command line or from within your IDE.
   - Experiment with different command-line arguments to modify the behavior of the application.
4. Using the Spring Initializr to bootstrap your application
   - Use the Spring Initializr to create a new Spring Boot project.
   - Customize the project settings to include specific dependencies and features.
   - Understand the difference between using the Spring Initializr and manually creating a Spring Boot project.
5. Build Something!
   - **Idea #1 - Simple REST API**: Create a simple domain object class, such as a `Book` class. Implement a REST controller class with endpoints to perform basic CRUD operations on the `Book` objects.
   - **Idea #2 - File Upload and Download API**: Implement a REST controller with endpoints to handle file upload and download operations.

3. Working with Spring Boot

This section covers the following topics:

- Introduction to Spring Boot configuration and properties
- Dependency Injection in Spring Boot
- Working with controllers and RESTful web services
- Using Spring Boot DevTools for faster development

Tasks

1. Introduction to Spring Boot configuration and properties
   - Understand how to use `application.properties` or `application.yml` to configure a Spring Boot application.

  - Experiment with different settings such as the server port or logging level, and see how they affect the application.
  - Create a custom configuration file and use it to configure a component of the application.

2. Dependency Injection in Spring Boot
   - Understand the basics of dependency injection and inversion of control.
   - Use the `@Autowired` annotation to inject dependencies into a Spring Boot application.
   - Experiment with different types of dependency injection such as constructor injection and setter injection.
3. Working with controllers and RESTful web services
   - Create a new controller and define multiple endpoints for it.
   - Use the `@RestController` annotation to create a RESTful web service.
   - Define different types of request mappings such as `@GetMapping`, `@PostMapping`, and `@DeleteMapping`.
4. Using Spring Boot DevTools for faster development
   - Understand what Spring Boot DevTools is and how it can speed up development.
   - Experiment with the different features of DevTools such as automatic restarts and live reload.
   - Try disabling and enabling DevTools to see how it affects the development workflow.
5. Build Something!
   - **Idea #1 - Weather Forecast Application**: Build a weather forecast application that retrieves weather data from an external API and exposes it through RESTful endpoints. Configure the API key and other settings using Spring Boot configuration properties, utilize dependency injection to manage the API client, and implement controllers to handle requests and provide weather information.
   - **Idea #2 - Simple Blogging Platform**: Create a simple blogging platform where users can create, read, update, and delete blog posts. Use Spring Boot configuration and properties to manage settings like the database connection details and the number of posts per page. Implement dependency injection to handle services like blog post management, and utilize controllers to expose RESTful endpoints for various operations.

4. Data Persistence with Spring Boot

This section covers the following topics:

- Introduction to data persistence with Spring Boot
- Working with Spring Data JPA for database access
- Connecting to a database using Spring Boot
- Using JdbcTemplate for accessing data

Name:
Date:

Tasks

1. Introduction to data persistence with Spring Boot
   - Understand how to configure database settings in a Spring Boot application.
   - Experiment with using different types of databases such as H2, MySQL, and PostgreSQL.
   - Create a simple CRUD (create, read, update, delete) application using Spring Data JPA.
2. Working with Spring Data JPA for database access
   - Understand what Spring Boot Actuator is and how it can help with monitoring and managing your application.
   - Experiment with different endpoints such as`/health`, `/metrics`, and `/info`.
   - Create custom endpoints using Spring Boot Actuator.
3. Connecting to a database using Spring Boot
   - Understand the basic principles of RESTful APIs.
   - Use Spring Boot to create a simple RESTful API that performs CRUD operations on a database.
   - Test the API using a tool such as Postman or cURL.
4. Using JdbcTemplate for accessing data
   - Understand the basic principles of Spring Security.
   - Use Spring Boot to create a secure RESTful API that requires authentication.
   - Experiment with different authentication and authorization methods such as JWT or OAuth2.

5. Spring Boot Advanced Topics

This section covers the following topics:

- Testing Spring Boot applications
- Securing Spring Boot applications with Spring Security
- Handling exceptions in Spring Boot
- Deploying Spring Boot applications to a production environment

Tasks

1. Testing Spring Boot applications
   - Understand the importance of testing in software development.
   - Learn about the different types of testing such as unit testing and integration testing.
   - Write unit and integration tests for a Spring Boot application using frameworks like JUnit and Mockito.
2. Securing Spring Boot applications with Spring Security

- Understand the basics of cloud computing and how it relates to deploying applications.
- Learn how to deploy a Spring Boot application to a cloud platform such as AWS or Google Cloud.
- Experiment with scaling your application and monitoring its performance.

3. Handling exceptions in Spring Boot
    - Understand the concept of profiles in Spring Boot and how they can be used to manage application configuration.
    - Learn how to define and activate profiles in a Spring Boot application.
    - Experiment with using different profiles to configure your application for different environments.
4. Deploying Spring Boot applications to a production environment
    - Explore more advanced topics such as Spring Boot Actuator, Spring Boot CLI, and Spring Boot Custom Starters.
    - Dive deeper into topics such as Spring Boot WebFlux, reactive programming, and asynchronous programming.

Build Something with Spring Boot!

Here are a few project ideas that you can build to practice and demonstrate what you have learned in Spring Boot:

- **TODO List Application**: Build a simple TODO list application using Spring Boot and Thymeleaf. The application should allow users to add, edit, and delete tasks, as well as mark them as complete.
- **Online Bookstore**: Build an online bookstore application using Spring Boot and a database of your choice (such as MySQL or PostgreSQL). The application should allow users to browse books, search for books, add books to a cart, and check out.
- **Blogging Platform**: Build a blogging platform using Spring Boot and a database of your choice. The application should allow users to create and publish blog posts, as well as comment on and share other users' posts.
- **Recipe Sharing Site**: Build a recipe sharing site using Spring Boot and Thymeleaf. The application should allow users to browse recipes, search for recipes by ingredient or cuisine, and upload their own recipes.
- **Social Media Platform**: Build a social media platform using Spring Boot and a database of your choice. The application should allow users to create profiles, connect with other users, share posts, and comment on and like other users' posts.