**6. Spring Boot - 2**

1. Checkout to the branch “spring-boot-app-v1” you created in spring boot - 1
2. Create a new branch “spring-boot-app-v2”
3. Propose a proper directory structure for your project created in spring boot part 1
4. Use API naming conventions to your endpoints
5. Use separate request and response types
6. Use exception handling where applicable
7. Return proper status codes
8. Use optionals where possible
9. Why do we need logging in our applications?

Logging in spring boot application record information, actions and events within an application. Also logging can be used to monitor the performance, understand the behavior of the application and to recognize the issues with the application.

1. What are different types of log levels?

Fatal - Log level that tells that the application encountered an event or entered a state in which one of the crucial business functionality is no longer working.

Error- Used for a non-recoverable error. Application hits an issue preventing one or more functionalities from properly functioning

Warning – Used for recoverable error. Something unexpected happened in the application

Info – Used for audit purpose. Indicates that something happened, the application entered a certain state.

Debug – Used for investigation

Trace – Used for detailed investigation

1. Use a logging framework and add logs to your application
2. Push your updated project to “spring-boot-app-v2”
3. Add your codes and answer sheet to a directory named “spring-boot-basic-training-v2” and push it to your training github repository
4. Create a pull request to main branch and assign it to your trainer

**Resources**

<https://nordicapis.com/10-best-practices-for-naming-api-endpoints/>

<https://restfulapi.net/resource-naming/>

<https://www.geeksforgeeks.org/spring-boot-code-structure/>