**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 is the process of capturing, storing and presenting the behavior and state of the application at runtime. It is important in many aspects.

* Debugging and Troubleshooting:

Logging helps in identifying and resolving issues during development, testing, and production.

* Monitoring and Alerting:

Logs can be used to capture metrics, events, and other relevant information, which can be monitored in real-time to detect anomalies, errors, and performance degradation, and trigger alerts for timely actions.

* Auditing and Compliance:

Logs can capture critical information about user actions, system events, and data changes, which can be used to trace the history of events and ensure compliance with regulatory requirements, such as data protection and privacy regulations.

* Performance Analysis and Optimization:

By analyzing logs, developers can identify performance bottlenecks, optimize resource usage, and fine-tune the application for optimal performance.

* Forensic Analysis:

Logs can provide a trail of events leading up to an incident, allowing investigators to analyze and understand the cause of the incident and take appropriate actions to mitigate the impact.

* Historical Record and Documentation:

They can be used for documentation purposes, as a source of information for troubleshooting, auditing, and analysis, and as evidence in case of disputes or legal issues.

1. What are different types of log levels?

DEBUG:

This log level is used for messages that provide detailed debugging information, such as variable values, method calls, and other internal details of the application's operation.

INFO:

This log level is used for informational messages that provide high-level details about the application's operation, such as successful events or important milestones. These messages are used for tracking the normal operation of an application.

WARN:

This log level is used for messages that indicate potential issues or warnings, such as deprecated features, suboptimal configurations, or other situations that may require attention. They are used to highlight potential issues that do not necessarily result in an error or failure.

ERROR:

This log level is used for messages that indicate errors or failures in the application, such as exceptions, errors in processing data, or other critical issues that require attention. These messages are typically used to indicate issues that need immediate attention and may result in the application not functioning properly.

FATAL:

This log level is used for messages that indicate severe errors or critical failures that result in the termination of the application or system. They indicate catastrophic failures that require immediate action to prevent further damage.

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/>