**Jobhunt Workbook**

**Java:**

**What is Java Serialization?**

Serialization is the conversion of the state of an object into a byte stream; deserialization does the opposite. Stated differently, serialization is the conversion of a Java object into a static stream (sequence) of bytes, which we can then save to a database or transfer over a network.

**Classes that are eligible for serialization need to implement a special marker interface**,**** Serializable.

When a class implements the java.io.Serializable interface, all its sub-classes are serializable as well.

When an object has a reference to another object, these objects must implement the Serializable interface separately, or else a *NotSerializableException* will be thrown

**What is “transient” keyword?**

If a class implements the Serializable interface, we can give transient keywod for any field that we don’t want to include in Serialization.

## **How do we use Serialization in spring and spring boot framework?**

Serialization plays a crucial role in Spring and Spring Boot applications. Here are some important uses of serialization in these frameworks:

****1. Stateful session replication:****  
Serialization allows Spring applications to replicate and distribute session state across multiple servers in a clustered environment. This ensures high availability and fault tolerance by allowing sessions to be seamlessly ransferred between different instances. By configuring a session store that supports serialization, such as Redis or Hazelcast, Spring Session can automatically serialize session objects and distribute them across a cluster of servers.

****2 . Caching :****  
Serialization is used in Spring's caching framework to store objects in a cache. Serialized objects can be stored in a distributed cache or a local cache, allowing for efficient retrieval and sharing of data between application instances.  
spring's caching abstraction, combined with a cache implementation like Ehcache or Redis, uses serialization to store and retrieve cached objects.  
By annotating methods with [@Cacheable](https://hashnode.com/@Cacheable" \t "https://www.adityatechinsights.com/_blank) or other cache-related annotations, Spring serializes the return values of those methods and stores them in the cache for subsequent fast retrieval.

****3. Messaging and remote communication:****  
Spring Integration, a module for building enterprise integration solutions, uses serialization to exchange messages between different components or systems. Messages can be serialized using various message formats such as JSON, XML, or Java objects and sent over message channels or messaging systems like JMS or AMQP.

****4 . Web session management:****  
In Spring Web applications, the HttpSession object is serialized and persisted between requests. This allows session data to be maintained across multiple requests and is managed by the servlet container or external session stores like Redis. Also serialized session objects can be stored in various locations such as cookies, databases, or distributed caches, ensuring session continuity and maintaining user-specific state.

****5. RESTful APIs and JSON/XML conversion:****  
Spring MVC, the web framework in Spring, automatically serializes Java objects to JSON or XML format for RESTful API responses. By using appropriate annotations such as [@RestController](https://hashnode.com/@RestController" \t "https://www.adityatechinsights.com/_blank) and [@ResponseBody](https://hashnode.com/@ResponseBody" \t "https://www.adityatechinsights.com/_blank), Spring handles the serialization and deserialization of objects as part of request processing.

****6 . Application state persistence:****  
Spring Data, a module that simplifies data access and persistence, often relies on serialization to store application-specific state in databases. For example, when using Spring Data JPA, entities are serialized and persisted to the underlying database tables.

****7 . Integration with external systems:****  
When integrating Spring applications with external systems like message brokers (e.g., Apache Kafka) or databases, serialization is used to exchange data between systems. For instance, when publishing or consuming messages with Spring Kafka, objects are serialized and deserialized to be sent or received over Kafka topics.

# **What is Vite.js?**

****Vite.js****is a development tool that comes with a dev server and is used for modern web applications. It offers a faster and smoother workflow in terms of development. It has two major parts:

* A ****dev server**** serves the source files over native ES modules, with build-in features and fast Hot Module Replacement (HMR) for updating modules during the execution of the application. When changes are made to the source code of an application, only those changes are updated, without the need to reload the entire application. This helps speed up the development time.
* A ****build command**** enables us to bundle our code with ***Rollup*** and offers highly optimized static assets for production.

**What is Bootstrap?**

 Bootstrap is a free, open source front-end development framework for the creation of websites and web apps. Designed to enable responsive development of mobile-first websites, Bootstrap provides a collection of syntax for template designs.

It’s main features:

* Responsive Grid System.
* Pre-designed UI Components.
* Extensive JavaScript Plugins.
* Bootstrap Themes and Customization.
* Active Community and Support.