Table 1: Concepts to be explored

|  |  |  |
| --- | --- | --- |
| Concept or Technology at the concept Level | Example Resource | SEVERITY |
| Idempotency |  |  |
| GraphQL & Netflix Falcor |  |  |
| Open API, Swagger & IDLs in general. What are client-side stubs and server-side skeletons? |  |  |
| REST APIs once and for all and how the benefits that Fielding mentions are achieved by complying to the REST constraints | <https://shorturl.at/gFhV8> |  |
| Maven scopes and why Lombok is provided scope for example |  |  |
| Authorization Methods (RBAC, ABAC, …) details and implementation. Blocked Ips, Password failures and time-outs, … more about security policies. |  |  |
| Soft deletes in DB and how Hibernate’s dynamic mapping helps you do so. |  |  |
| Encryption and Hashing methods: SHA(256), MD5, PBE, base4, etc. |  |  |
| Git internal architecture and more advanced commands like cherrypick, squash, … |  |  |
| Spring application events, the Observer pattern, and annotation-based event listeners |  |  |
| HTTP in more depth: cookies, headers, caching, etc. why is it firewall friendly? Other verbs, … |  |  |
| User impersonation in spring security |  |  |
| Java Execution Cycle, Memory management, GC, JVM references and all about low level subjects in java. |  |  |
| Elastic Search |  |  |
| Class loader, Memory, execution cycle, JVM and JVM references, Garbage Collection, JRE vs JDK, JIT vs interpreter, GraalVM, etc. class path, compilation class path vs execution class path. | <https://www.geeksforgeeks.org/differences-jdk-jre-jvm/> |  |
| Java8, scope of variables in lambdas, etc.. |  |  |
| Literals in Java, d, f, … |  |  |
| Transparency in dependent classes like in the context of JPA, or test classes to the code under test. |  |  |
| Sealed access modifier in java 17 |  |  |
| Multi-tenancy vs Row-level security vs ACL |  |  |
| PostgreSQL’s Generalized Search Tree and multi-dimensional indexes |  |  |
| More on Enums, overring its methods, indices, etc. |  |  |
| Excellent resource on memory and synchronization  <https://jenkov.com/tutorials/java-concurrency/java-memory-model.html> |  |  |
| LockSupport. park in concurrency |  |  |
| Fail-fast and fail-safe iterations |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Table 2: Skills to be acquired

|  |  |  |
| --- | --- | --- |
| Skill | EXample Resource | SEVERITY |
| GraphQL |  |  |
| gRPC and async gRPC |  |  |
| Writing Solid Tests | * <https://www.youtube.com/watch?v=rUbjV3VY1DI> * <https://www.youtube.com/watch?v=jqwZthuBmZY&list=PL82C6-O4XrHcg8sNwpoDDhcxUCbFy855E> * <https://www.youtube.com/watch?v=hR0bbk2tsF0&list=WL&index=2> * <https://youtu.be/-H5sud1-K5A?si=NqF-ucjVHweeEsEQ> * <https://youtu.be/2bTAb-2vhBk?si=h4xgDcYlykHPEl_r> * <https://docs.spring.io/spring-boot/reference/testing/index.html#testing> |  |
| Spring cloud config |  |  |
| Serialization in Java,  More of Jackson like @JsonSerialize |  |  |
| Concurrency in Java |  |  |
| Multi-module Maven Projects, Parent POMs, BOMs, and aggregate POMs. |  |  |
| Hibernate Dynamic Mapping (@Filter, @Where, etc.) | <https://www.baeldung.com/hibernate-dynamic-mapping> |  |
| @Async in Spring |  |  |
| Using abstraction in spring beans and how they implemented the service layer using it. |  |  |
| MapStruct |  |  |
| Caching in Spring, Second level cache in hibernate, caching in REST, Materialized Views, Application Cache, and caches in general. |  |  |
| Calculated columns and using @Formula in hibernate |  |  |
| Handling Blobs in Database and file storages |  |  |
| Configuration management in spring |  |  |
| How they write things like orElseThrow, buildAndThrow for builders, etc. |  |  |
| Sending http requests in java |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Table 3: Questions to be answered

|  |  |
| --- | --- |
| Question | SEVERITY |
| Why isn’t declaring operations in REST APIs as query parameters or using sub-resources like order/cancel REST compliant? |  |
| Why Should PUT requests be idempotent and how do you treat update operations that are not? |  |
| Synchronous vs Asynchronous vs Reactive vs Non-Blocking vs Blocking |  |
| CGLIB vs Dynamic Proxies vs AspectJ |  |
| Difference between key-value and document stores |  |
| How not to ruin git history when renaming files |  |
| What are the use cases of inner classes? |  |
| Read-only mode in hibernate and its special behavior when dealing with associations especially mapped collections. And remove operations |  |
| How are Java SE codes and other libraries integrated in your application? |  |
| How are arrays implemented behind the scenes in Java |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Table 4: Here and There stuff

|  |  |
| --- | --- |
| Item | SEVERITY |
| Ipconfig/(preferred) DNS/IP settings on windows/gateway/network mask/etc.  [\\192.0.0.1](file:///\\192.0.0.1) in windows file explorer. |  |
| Triggers/with(CTE) clause/exists/ |  |
| Docker/Docker compose in spring |  |
| How does Gitlab and nexus work locally especially Gitlab auth. |  |
| Spring Common Logging spring-jcl |  |
| Camunda framework |  |
| Spel |  |
| <servers>, <mirrors> and setting.xml in Maven |  |
| How to log people’s actions and how to use Elasticsearch and Redis for logging purposes. How can Kafka be used in this context?  How did people in Vira use AOP to do so with @Async? |  |
| Marker annotations and how to use them for debugging |  |
| FreeMarker Java template engine |  |
| SSL, TLS, SSL redirect, Socket Programming |  |
| LDAP vs DB vs FTP |  |
| Schema in @Table |  |
| The Idea of having something like BaseInformation |  |
| Named queries in hibernate and using them to call stored procedures |  |
| Kent Beck, Extreme Programming Explained: Embrace Change |  |
| The difference between @Service, @Repository,… |  |
| Jhipster |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |