DailyHW\_Day2\_Week6

03/30/2022

Notes and Q&A:

<https://docs.google.com/document/d/1O7ogrWUuttxBATKADPk98zPqyQcgCWbdF18YKx0xcSQ/edit?usp=sharing>

Checklist:

<https://docs.google.com/document/d/1fmYhFHUIy44UIId1fpUUV-vWt9RdjD0ghkv9n5PqzBw/edit?usp=sharing>

1. What have you reviewed today?

* Hibernate features
  + ORM
  + Persistence
  + JPA Provider
  + … etc
* Hibernate Framework
  + Config File
  + Session Factory
  + Entity
  + Session
* Fetch Type: how Hibernate fetches data
* ER mapping
* Cascade: used to manage the state of the mapped entity when the owner of the relationship gets affected
* Cache: Hibernate first-level and second-level cache

Reflection

* java.lang.reflect package
* Reflection can be used to get information about –
* Class
* Constructor
* Method

* Can access private variables and methods via reflection
* Advantage
* **Extensibility Features**
* **Debugging and testing tools**
* Drawback
* **Performance Overhead**
* **Exposure of Internals**

Design Pattern

* **Singleton**
* Reduce initialization overhead
* **Factory**
* Creating object without exposing the logic and the details to the client
* **Builder**
* Simplifies the creation of an object
* **Proxy**
* create an intermediary that acts as an interface to another resource, also hiding the underlying complexity of the component.

1. What new things you have learned today?

Application Architecture

MVC pattern

* Model
* View
* Controller

Frontend-backend separation

* Backend servers send data to frontend through APIs using JSON and other formats.
* Frontend includes the UI related elements and rendering logic.

3 tier architecture

* Web layer
* Service layer
* Data layer

Monolithic

* Pro:
* Simplicity of development, debugging, deployment
* Cross-cutting concerns and customizations are used only once
* Low cost in the early stages of the application
* Con:
* High code coupling
* Lack of flexibility
* Problem of deployment

SOA

* Service oriented architecture
* Focusing on interaction and integration between independent services, usually using a service bus, SOAP or messaging system.

Microservice

* Service Discovery
* Eureka, AWS
* Communication
* Http, restful api
* Log tracing
* Spring Cloud Sleuth, Zipkin
* Config Server
* Fault Tolerance
* Spring Cloud Sleuth

1. What is your learning plan tomorrow?

* 2 coding questions
* Review Securities and Spring related concepts