



**WELCOME BACK
LESSON 9**



Spring/Spring Boot Crash Course

For TD Bank

MEET YOUR CRASH COURSE TEAM



TANGY F.
CEO



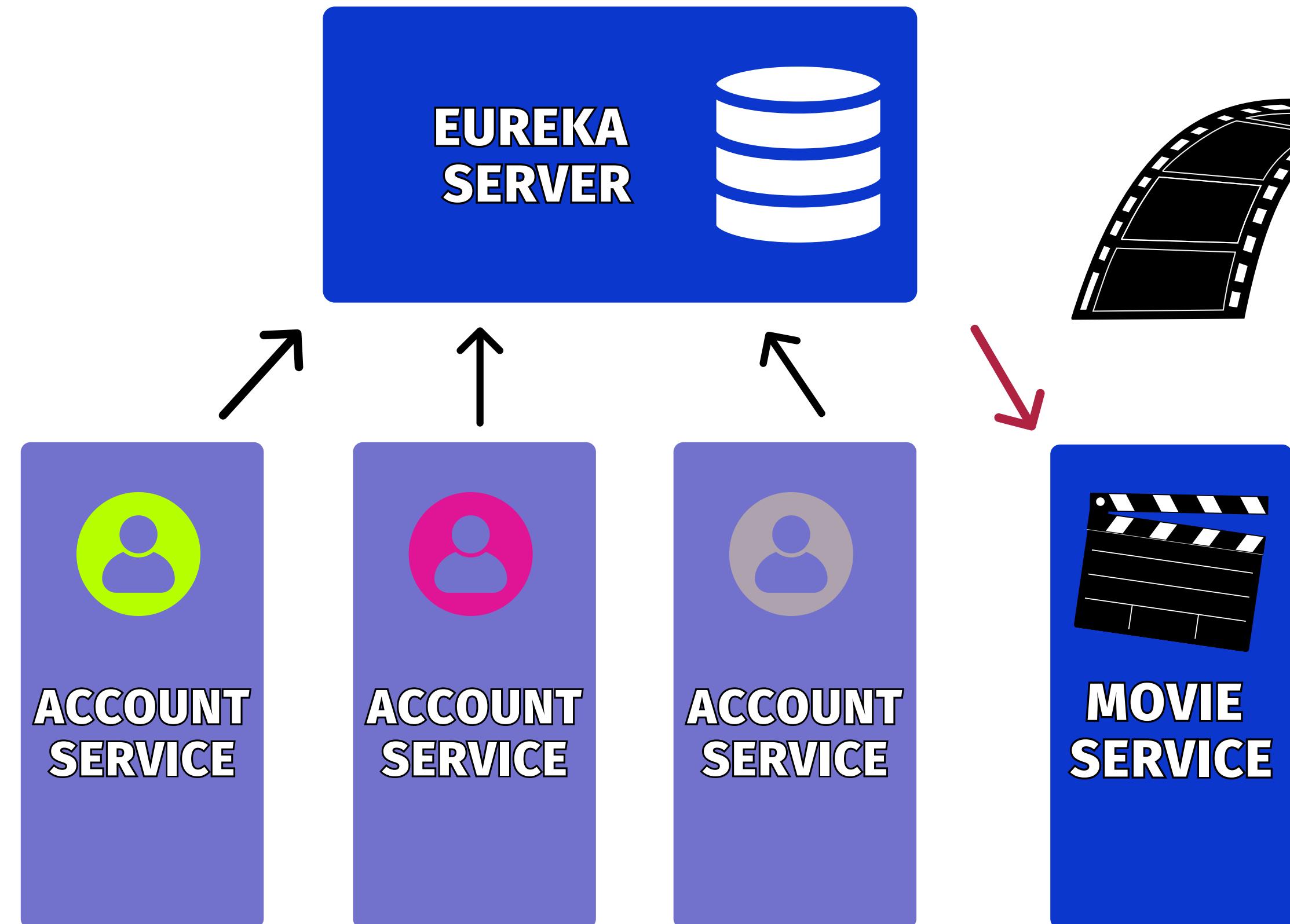
HAL M.
DEVELOPER



WILLIAM D.
DEVELOPER

WE ARE BUILDING

BITE SIZE
MOVIE REVIEW APP



<Cre8tive
Software/>
Devs.

LESSON 9

AGENDA

1

Recap Lesson of 8
=>Rapid review of Lesson Eight

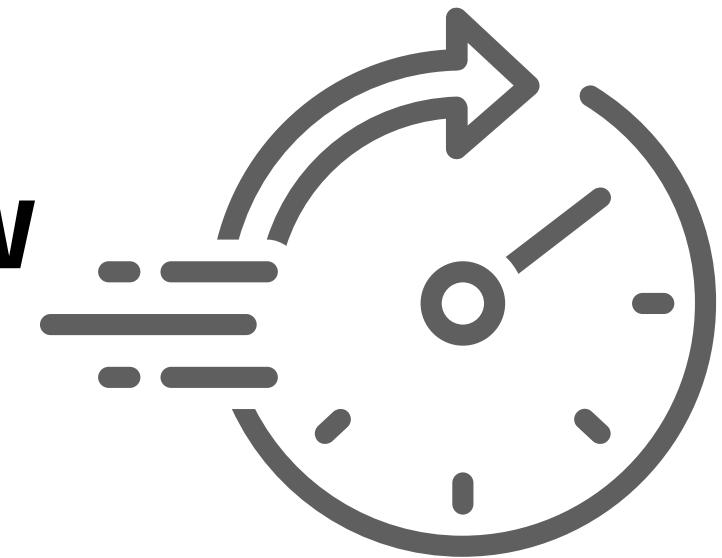
2

Current Lesson
=>Fault tolerance
=>Design Patterns
 =>>The design patterns in Spring boot
 =>>Singleton Pattern
=>12 Factor App

3

Lab / Assessment and Q&A

WELCOME TO LESSON 9



Rapid Review Trivia



Rapid Review
of
**LESSON
7&8**

- 1. What is the difference between JPA and Hibernate**
- 2. How can we disable Eureka acting as a client?**

RUNNING THE CODE HANDS ON EXERCISE

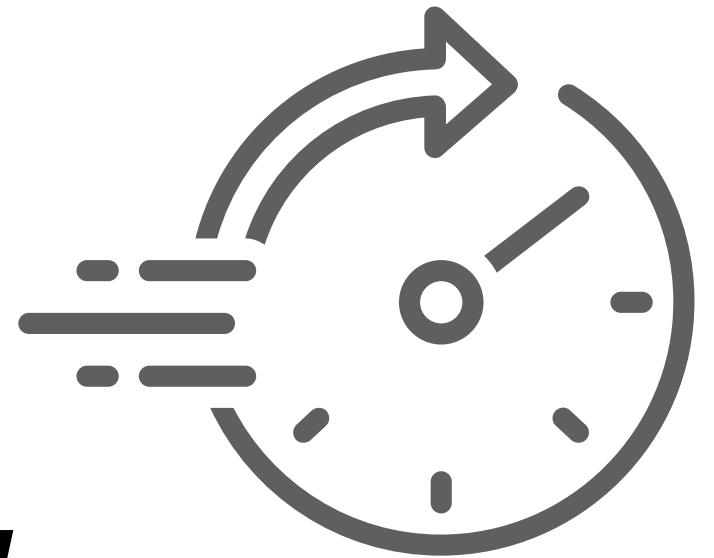
RUNNING THE CODE EXERCISE



A screenshot of a GitHub repository page titled "Source". The repository is named "Cre8tivePilot". The README.md file contains instructions for a Spring Boot pre-assessment. It includes a list of 12 steps for setting up and running the application, as well as a section for a longer pre-assessment. The GitHub interface shows a sidebar with options like "Commits", "Branches", and "Pull requests".



WELCOME TO
LESSON 9



Rapid Review
of
LESSON 8



Rapid Review
Eureka Server

LESSON 9

Fault Tolerance

Design Patterns

12 Factor App

LESSON 9

Fault Tolerance

Fault Tolerance in Spring Boot applications

The application needs to be reliable and robust.
There are few methods to handle the fault tolerance
in Spring like using different patterns.
Having exception handling.

Different patterns to handle Fault Tolerance

- Circuit break pattern
- Retry Pattern
- Timeout handling
- Bulkhead Pattern
- Distributed tracking and Monitoring

LESSON 9

Design Patterns

Benefits of Singleton Pattern

- By default all the beans defined in Spring are singleton
- Resource sharing
- Better performance
- Maintaining consistency

LESSON 9

12 Factor App

1. Codebase

>> Version control
best practice

Example: Git

3. Config

>> Have all configuration in
properties or any YAML file
instead of hard coding them.
>> Using environmental
variable and external config
files

2. Dependencies

>> Using a dependency
management system
like Maven or Gradle
>> Spring Boot provides
dependency
management plugins

4. Backing Services

- >>DB or external services are considered as backing services
- >>Spring boot provides integration modules and libraries such as JPA, Hibernate for connection

5. Build, Release, Run

- >>Use Maven or Gradle for build application into executable JAR or WAR files
- >>Spring supports these executable file to easily run and deploy.

6. Processes

- >>Designing a stateless process and having the data store in DB or in Caches
- >>Spring supports the DB storage and external cache service

7. Port Binding

- >>Bind the services or Rest applications to specific port
- >>In spring we can adjust the port number for Tomcat or Jetty

8. Concurrency

- >>Scaling the applications horizontally
- >>Spring gives the content caching or distributed caching to handle concurrency

9. Disposability

- >> Ensure fast startup and graceful shut down of application
- >> Spring provides lifecycle hooks and shutdown events to cleanup activity.

11. Logs

- >> Having logging enabled for the application. It helps in monitoring and trouble shooting.
- >> Spring Boot has a built-in logging framework like Log4j

10. Dev/ Prod Parity

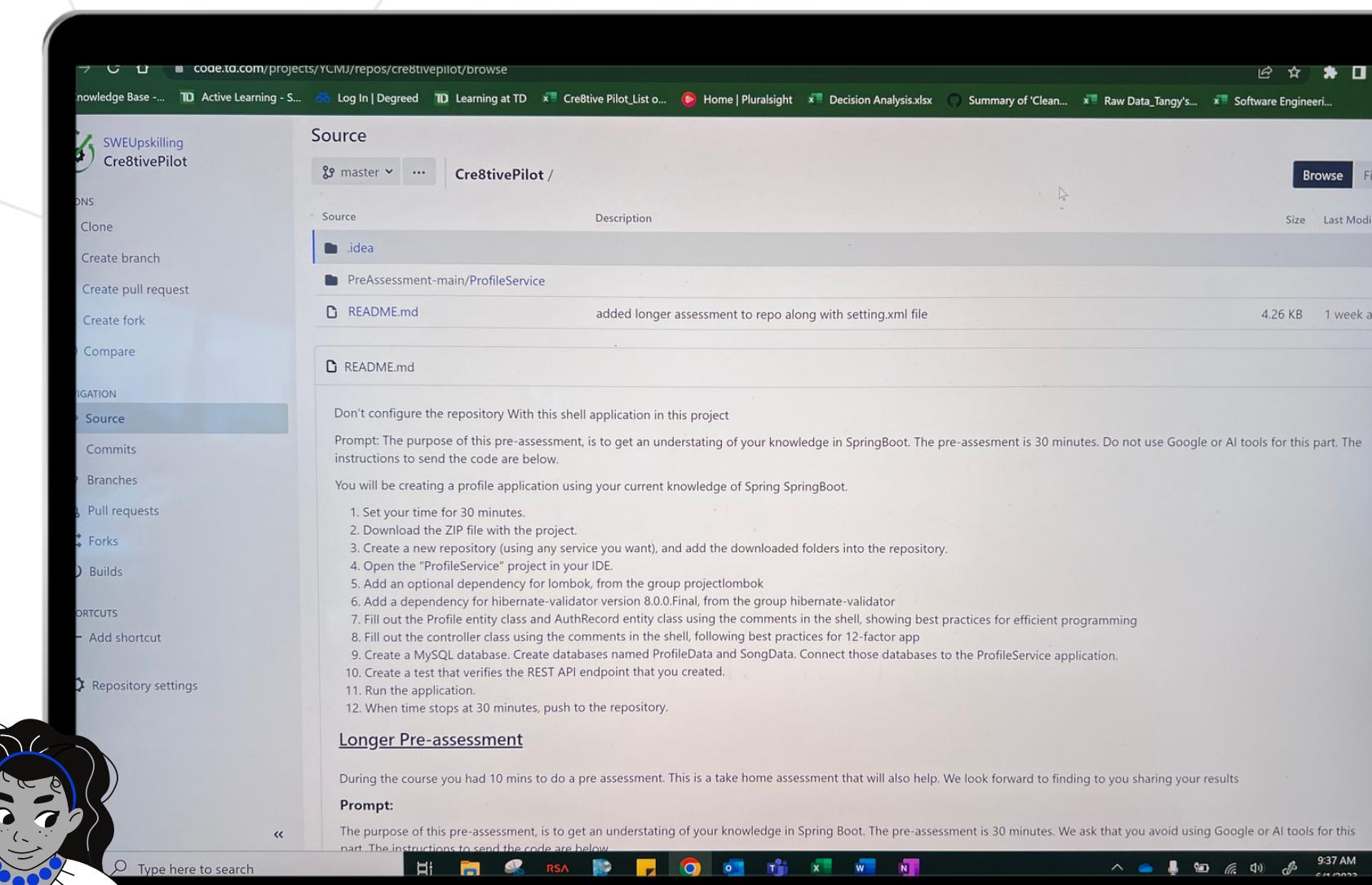
- >> Maintaining all the environments as possible.
- >> Spring Profiles can be used to manage environment specific configs.

12. Admin Processes

- >> Perform administrative tasks as separate.
- >> Spring boot features like Spring Batch and scheduling tasks to execute the administrative processes.

YOU CAN WORK ON YOUR HANDS ON EXERCISE

HANDS ON
EXERCISE TIME



<Cre8tive
Devs.
Software/>



THANK YOU

<Creative Software/>

Crash Course

We will see you on Tomorrow