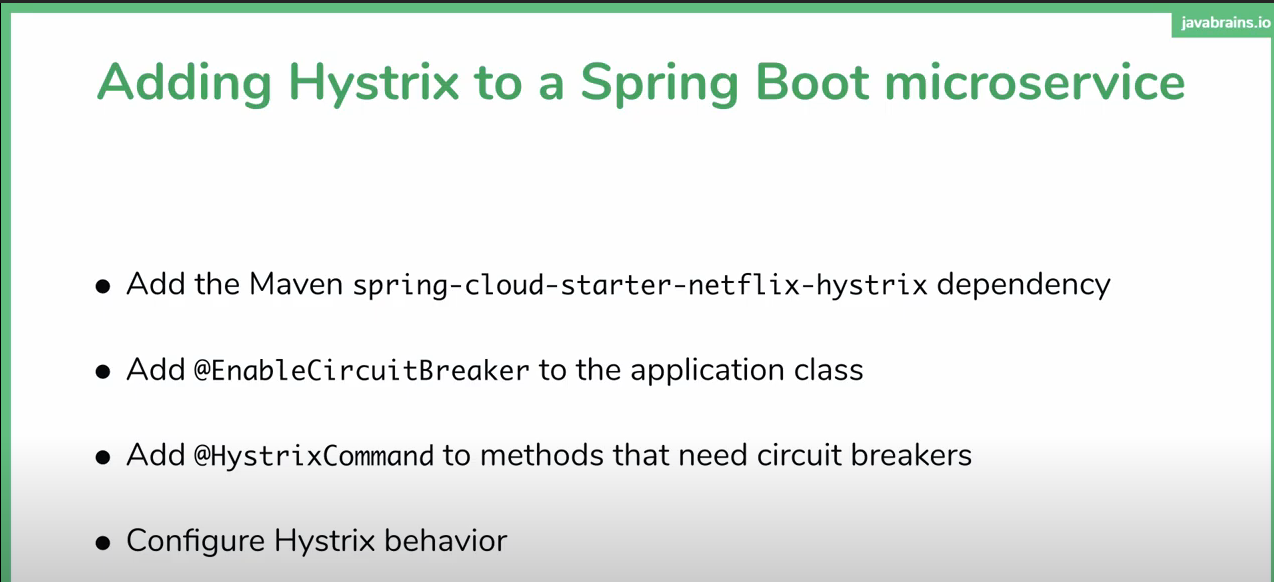
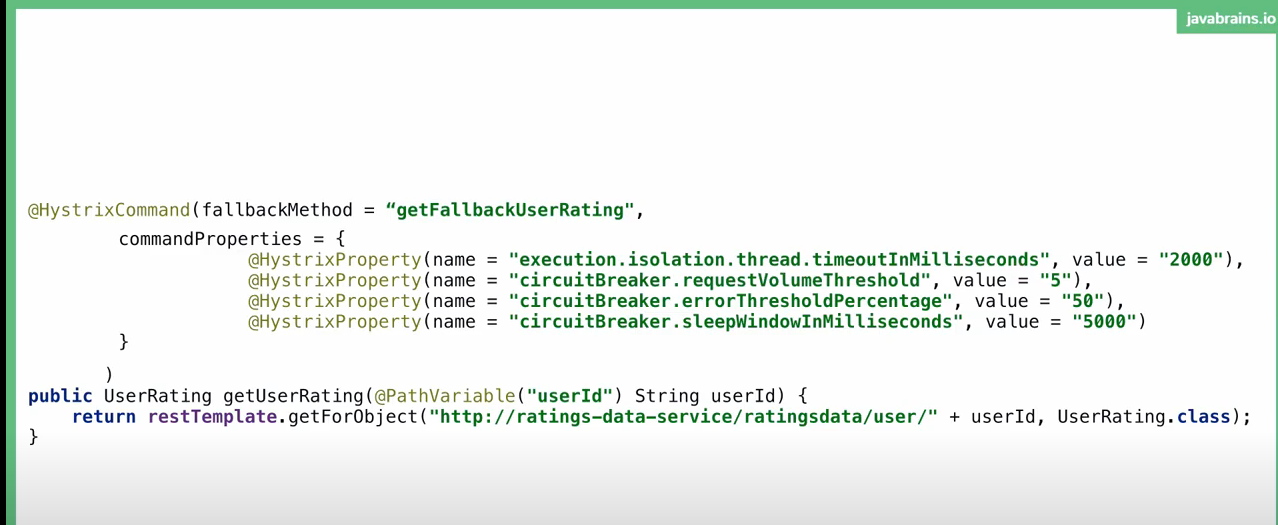
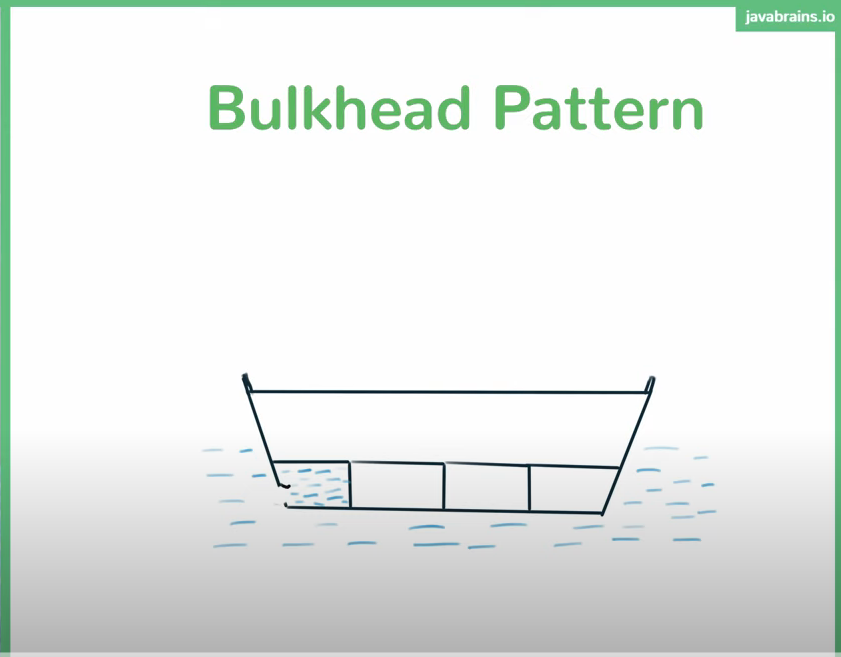
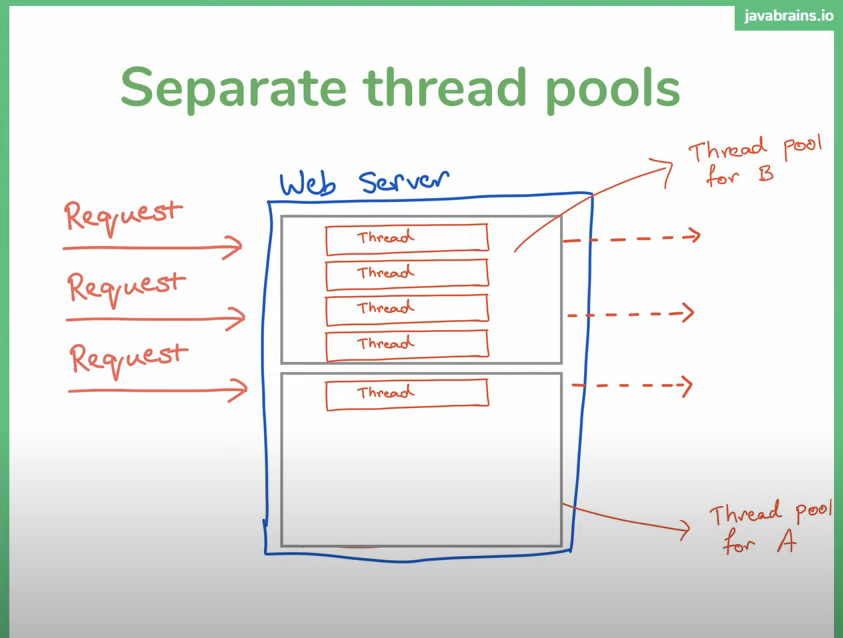
Circuit breaker:

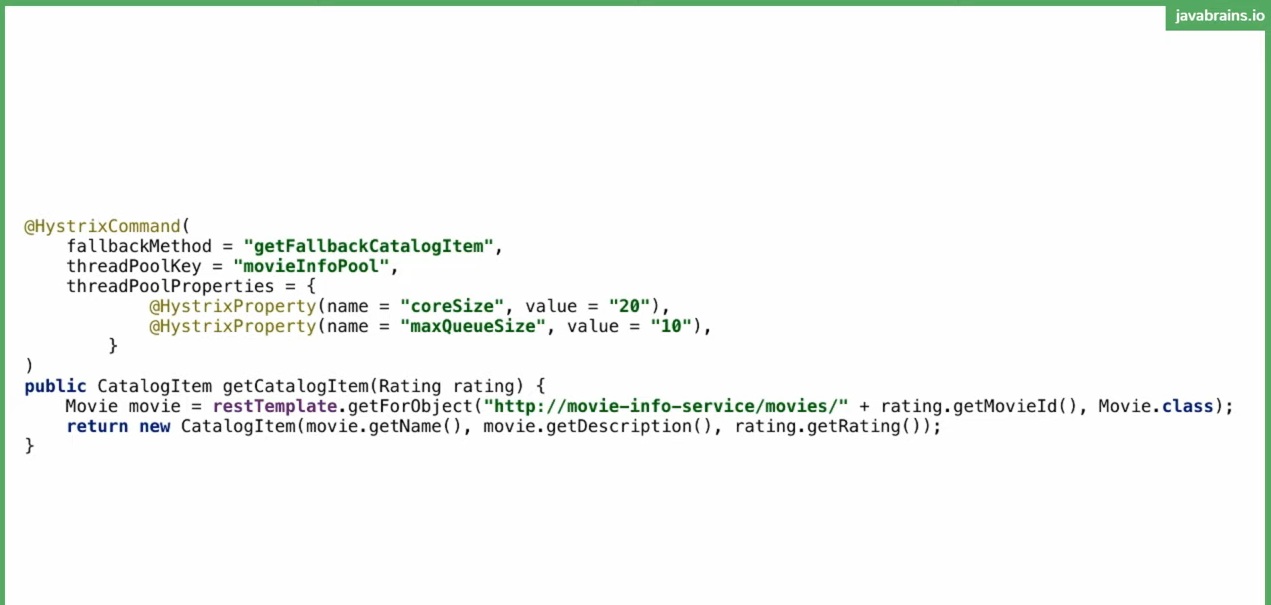


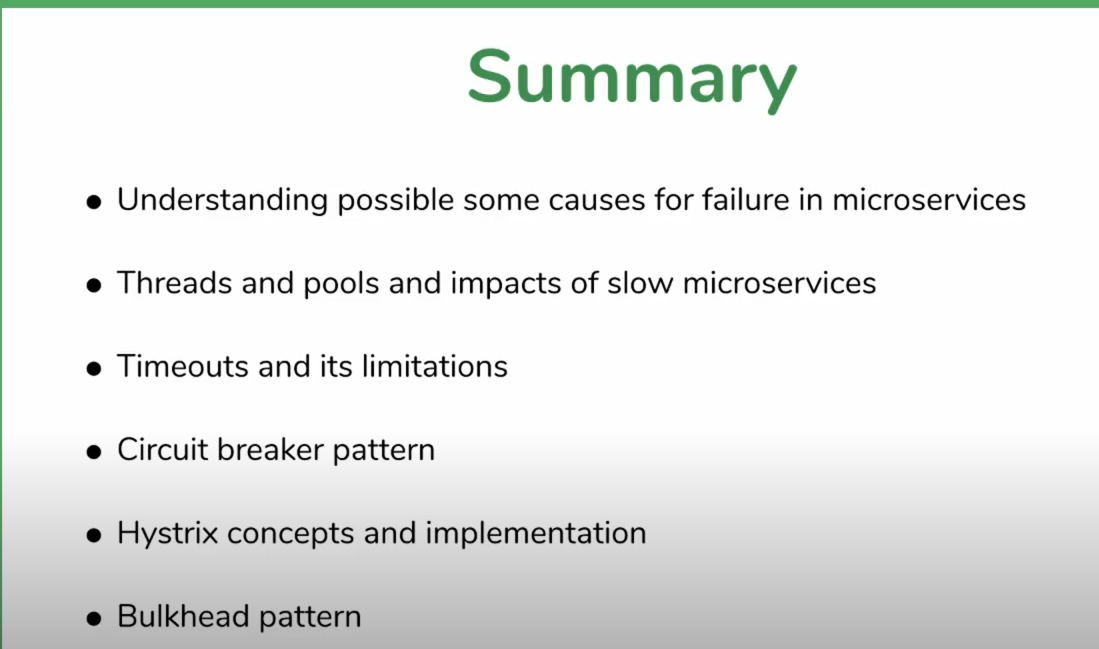




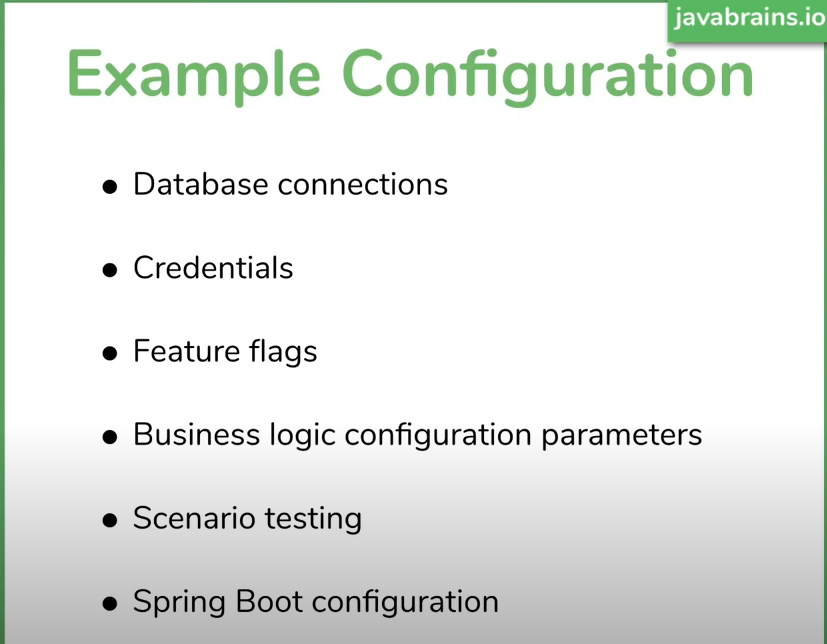




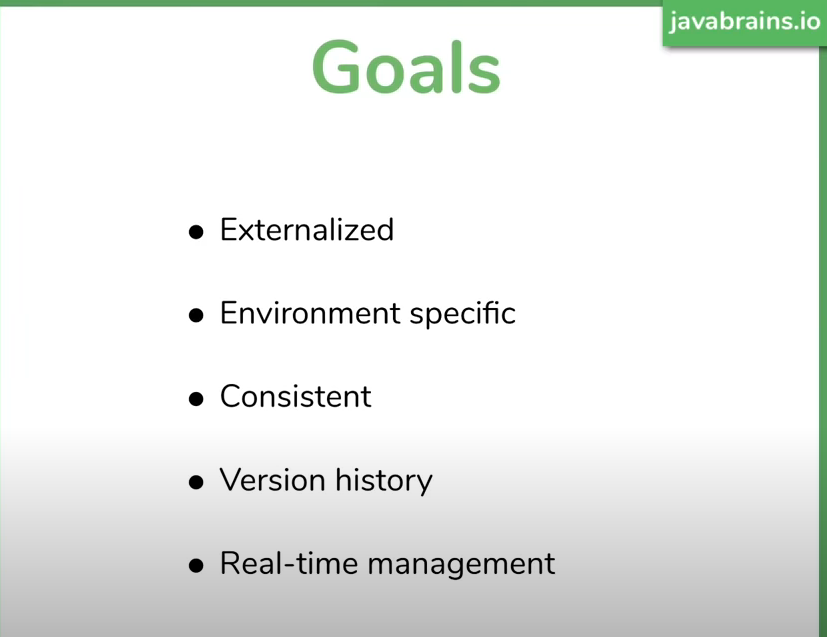




* **Config server**







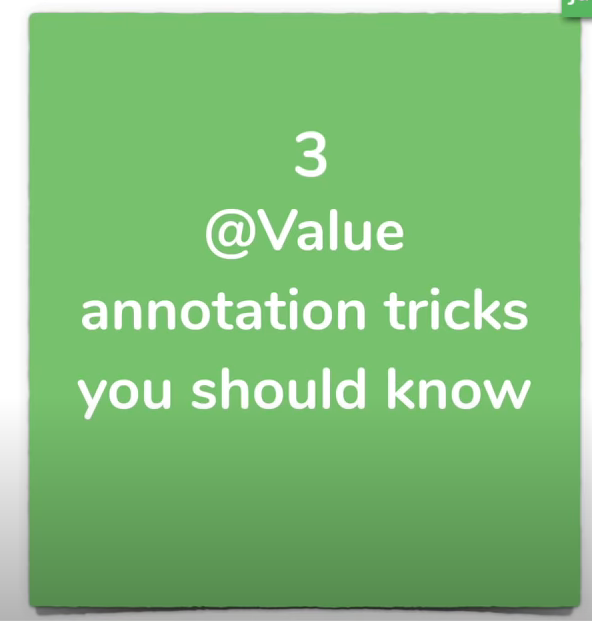


In property file

my.greeting = hello World

@Value("${my.greeting}")  
String greeting;

@GetMapping("greeting")  
public ResponseEntity<?> getGreeting(){  
 return ResponseEntity.*ok*().body(greeting);  
}



Property file

my.greeting = hello World  
  
my.list = one.two,three  
my.map = {firstName:'Shailesh',lastName:'munot',age:'30'}

@Value("${my.greeting:Greeting not available}")  
 String greeting;  
@Value("static msg")  
String staticMsg;  
  
@Value("${my.list}")  
List<String> list;  
  
@Value("#{${my.map}}")  
Map<String,String> map;

@GetMapping("greeting")  
public ResponseEntity<?> getGreeting(){  
 return ResponseEntity.*ok*().body(greeting+staticMsg+list+map);  
}



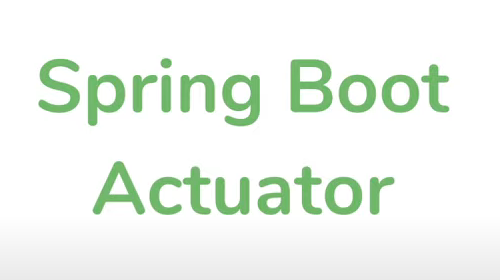
Property file

db.host = host  
db.port 1234  
db.connection = connect

package com.srm.moviecatalogservice.config;  
  
import org.springframework.boot.context.properties.ConfigurationProperties;  
import org.springframework.context.annotation.Configuration;  
  
@Configuration  
@ConfigurationProperties("db")  
public class DbSettings {  
 private String host;  
 private String connection;  
 private int port;  
  
 public String getHost() {  
 return host;  
 }  
  
 public void setHost(String host) {  
 this.host = host;  
 }  
  
 public String getConnection() {  
 return connection;  
 }  
  
 public void setConnection(String connection) {  
 this.connection = connection;  
 }  
  
 public int getPort() {  
 return port;  
 }  
  
 public void setPort(int port) {  
 this.port = port;  
 }  
  
}

@Autowired  
DbSettings dbSettings;

@GetMapping("greeting")  
public ResponseEntity<?> getGreeting() {  
 return ResponseEntity.*ok*().body(greeting + staticMsg + list + map +  
 dbSettings.getConnection() + dbSettings.getHost() + dbSettings.getPort());  
}



Localhost:8080/actuator/configprops

Will give you all the properties from application property and yaml file as well as pre defined spring properties.

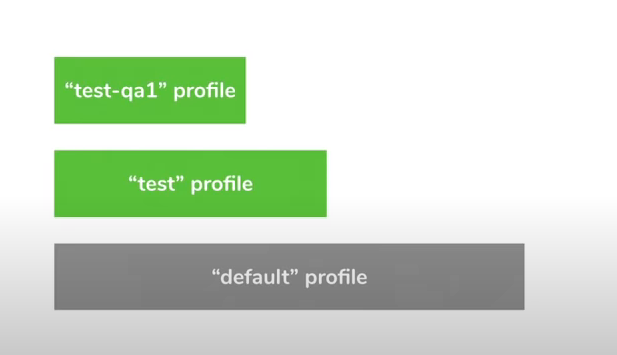


server:  
 port: 8686  
spring:  
 application:  
 name: "movie-catalog-service"  
  
eureka:  
 client:  
 serviceUrl:  
 defaultZone: "http://localhost:8082/eureka/"  
management:  
 endpoints:  
 web:  
 exposure:  
 include: "\*"  
 base-path: "/"  
  
my.greeting: hello World



By adding “spring.profiles.active:local” property in property file will make the profile based setting

Inheritence is applied in profile configuration.



All the values which are common across the all env like prod,qa and dev can be placed in default property file.

**Using cmd line configuration:**

