

## Today topics of Discussion

=====

Note: SpringBoot Profiles(6th april Enterprise java batch video will be uploaded)

1. Springcloud Config Server
2. Hystrix Circuit breaker
3. Ribbon(Client side Load Balancing)

## Evening discussion

=====

1. Distributed Logging
2. RedisCache
3. Apache-Kafka Integration

## Wednesday(Evening slot)

- a. Webflux
- b. SpringSecurity(Basic Authentication)

## SpringCloud Config Server

=====

=> Spring Cloud provided Config server  
=> Config Server is used to externalize the configuration properties from our application.

## Scenario

Microservice -----> Config Server -----> Git Repo

## Steps to work with Config Server

=====

1. Create a SpringBoot application with below dependencies
  - a. spring-boot-starter-web
  - b. spring-cloud-config-server
2. Configure @EnableConfigServer at springboot starter class
3. Create git repository and create config-properties file in repository and then configure properties in the key-value pair  
url:https://github.com/TEK-Leads/Config-Properties
4. Configure below properties in application.properties/application.yml file
  - a. server.port = 9999
  - b.  
spring.cloud.config.server.git.uri=https://github.com/TEK-Leads/Config-Properties.git
  - c. spring.cloud.config.server.git.cloneOnStart=true
  - d. management.security.enabled=false

## Steps to work with ConfigServer client

=====

1. Create a SpringBoot application with below dependencies
  - a. spring-boot-starter-web
  - b. spring-cloud-starter-config
2. Configure below properties in application.properties/application.yml file(specify file name as bootstrap.properties)
  - a. server.port = 8888
  - b. spring.application.name = client-config

```
c. spring.profiles.active= prod
d. spring.cloud.config.uri=http://localhost:9999
e. management.security.enabled = false
```

### 3. Injecting properties to the variables

```
@Value("${msg}")
String msg;
```

### 4. Run the client application

```
refer:SB-Config-Server-App
      SB-Config-Server-Client-App
```

### Circuit Breaker

=====

A circuit breaker is an electrical safety device designed to protect an electrical circuit from damage caused by an overcurrent or short circuit.

Its basic function is to interrupt current flow to protect equipment and to prevent the risk of fire.

How is it related to SpringBoot?

Using SpringBoot we are developing Microservices in Java.

When we follow Microservices Architecture, we will have several services.

Our Services is nothing but RestApi's

```
refer:: SB-Cloud-Hystrix-Circuit-Breaker
```