

```
#####  
Circuit Breaker  
#####
```

-> Circuit Breaker is a design pattern

-> It is used to implement fault tolerance systems

-> Fault Tolerance systems are also called as resilience systems

Requirement:

-> When m1 () method failed to give response to client then m2() method should provide response to client.

```
-----  
@RestController  
public class DemoRestController {  
  
    @GetMapping("/")  
    public String m1() {  
        System.out.println("*****m1() method executed.....");  
        String msg = "This is m1() method response";  
        try {  
            int i = 10 / 0;  
        } catch (Exception e) {  
            e.printStackTrace();  
            msg = m2();  
        }  
        return msg;  
    }  
  
    public String m2() {  
        System.out.println("*****m2() method executed.....");  
        String msg = "This is m2() method response";  
        return msg;  
    }  
}
```

-> As per above program when exeception occurred in 'try' block then catch block will be executed and it is calling 'm2 ()' method.

-> When m1() method is failing continuously (ex : for 5 requests) then i want to execute only m2 () method directley for next 30 minutes. We can achieve this requirement by using 'Circuit Breaker'.

```
-----  
@SpringBootApplication  
@EnableHystrix  
public class Application {  
  
    public static void main(String[] args) {  
        SpringApplication.run(Application.class, args);  
    }  
}
```

```

-----
@RestController
public class DataRestController {

    @GetMapping("/data")
    @HystrixCommand(
        fallbackMethod="getDataFromDB",
        commandProperties= {

            @HystrixProperty(name="circuitBreaker.requestVolumeThreshold",
value="5"),

            @HystrixProperty(name="circuitBreaker.sleepWindowInMilliseconds",value
="10000")
        }
    )
    public String getDataFromRedis() {
        System.out.println("***Redis() method called**");

        if (new Random().nextInt(10) <= 10) {
            //throw new RuntimeException("Redis Server Is Down");
        }
        // logic to access data from redis
        return "data accessed from redis (main logic) ....";
    }

    public String getDataFromDB() {
        System.out.println("***DB() method called**");
        // logic to access data from db
        return "data accessed from database (fall back logic) ....";
    }
}
-----

```