**--------------------------------------Use only for gateway project----------------------------------**

This is used for catching error from client server when occur.

Dependencies:

<dependency>  
 <groupId>org.springframework.cloud</groupId>  
 <artifactId>spring-cloud-starter-circuitbreaker resilience4j</artifactId>  
<!-- <version>2.1.2</version>-->  
</dependency>

Config property:

feign:  
 client:  
 config:  
 default:  
 connectTimeout: 5000  
 readTimeout: 5000  
 circuitbreaker:  
 enabled: true

fistly, we create an interface to navigate to proper client-server

@FeignClient(name = "statistic-service", url = "http://localhost:9082/api", fallback = StatisticFBServiceImpl.class)  
public interface StatisticService {  
 @GetMapping(value = "/statistic")  
 void getAllAcc();  
  
 @GetMapping(value = "/statistic/{id}")  
 void getAccById(@PathVariable(name = "id") Long id);  
  
 @PostMapping(value = "/statistic")  
 void saveStatistic(@RequestBody Statistic statistic);  
  
 @PostMapping(value = "/statistic/{id}")  
 void deleteStatistic(@PathVariable(name = "id") Long id);  
}

When client server has problems, we handle by class **fallback = StatisticFBServiceImpl.class) like:**

@Component  
public class StatisticFBServiceImpl implements StatisticService {  
 @Override  
 public void getAllAcc() {  
 // handle problems when can not get all accounts  
 }  
  
 @Override  
 public void getAccById(Long id) {  
// handle problems when can not acc by Id  
 }  
  
 @Override  
 public void saveStatistic(Statistic statistic) {  
  
 }  
  
 @Override  
 public void deleteStatistic(Long id) {  
  
 }  
}

Finally, add

@EnableFeignClients

To class Application to enable **FEIGN service breaker feature**, and run project.