



```

        HashMap<String, Data> hm = new HashMap<String,
Data>();
        try {
            ObjectInputStream ois = new ObjectInputStream(new
FileInputStream("data.txt"));
            hm = (HashMap<String, Data>) ois.readObject();
//data.txt
        } catch (IOException ex) {
Logger.getLogger(Tester.class.getName()).log(Level.SEVERE,
null, ex);
        } catch (ClassNotFoundException ex) {
Logger.getLogger(Tester.class.getName()).log(Level.SEVERE,
null, ex);
        }

        System.out.println("intval :
"+hm.get("Key1").intval+", strval : "+hm.get("Key1").strval+",
boolval : "+hm.get("Key1").boolval);
    }

```

```

run:
intval : 1, strval : Key1 String Value, boolval : true
BUILD SUCCESSFUL (total time: 1 second)
|

```

## IntelliJ, VSCode, Eclipse, Visual Studio

	IntelliJ	VSCode	Eclipse	VS
Code completion	Ctrl + Space		Ctrl + Space	Alt + Right
SmartType code completion	Ctrl + Shift + Space		-	
Reformat, Code Tidy	Ctrl + Alt + L	Ctrl + Alt + F	Ctrl + Shift + F	
Quick Fix	Alt + Enter		Ctrl + 1	

IDE	IntelliJ	VSCode	Eclipse	VS
Navigate   Declaration (Open Declaration)	Ctrl + B		F3	
Go to file(Open Resource)	Ctrl + Shift + N	Ctrl + P	Ctrl + Shift + R	
Find in Path (Search Project)	Ctrl + Shift + F		Ctrl + H	
Rename	Shift + F6	F2	Ctrl + Alt + R	F2 or Ctrl + R, R
Extract Variable	Ctrl + Alt + V			
Extract Method	Ctrl + Alt + M			
Quick Documentation	Ctrl + Q		F2	
Go to class (Go to Declaration)	Ctrl + N		Ctrl + Shift + T	Ctrl + ,
Go to symbol	Ctrl + Alt + Shift + N		-	
Go to previous (Go to Previous)	Ctrl + Alt + ←		Alt + ←	Ctrl + -
Go to next (Go to Next)	Ctrl + Alt + →		Alt + →	Shift + Ctrl + -
Go to previous (Go to Previous)	Ctrl + Shift + ↑	Alt + ↑	Alt + ↑	Alt + ↑
Go to next (Go to Next)	Ctrl + Shift + ↓	Alt + ↓	Alt + ↓	Alt + ↓

기능	IntelliJ	VSCode	Eclipse	VS
한 줄씩 줄 바꿈	Ctrl + D	Ctrl + C Alt + Shift + ↓	Ctrl + D	Ctrl + C
기능	IntelliJ	VSCode	Eclipse	VS
빠른 검색 (Quick Search Action)	Ctrl + Shift + A	Ctrl + Shift + P	Ctrl + 3	Ctrl + Q
Search everywhere (Quick Search Plugin)	Double Shift		Ctrl + Shift + L	
View Hierarchy (Show Hierarchy)	Ctrl + H		Ctrl + T	
Find Usages (Find References)	Alt + F7		Ctrl + Shift + G	
Find	Ctrl + F	Ctrl + F	Ctrl + F	
Find next	F3		Ctrl + K	
Find previous	Shift + F3		Ctrl + Shift + K	
Replace	Ctrl + R		Ctrl + F	
Find in path (Search)	Ctrl + Shift + F		Ctrl + H	
개발자 도구 열기				Shift + Alt + F10
파일 열기				Ctrl + W, S
파일 닫기		Ctrl + ,		Ctrl + W, O

기능	IntelliJ	VSCode	Eclipse	VS
클래스 뷰 열기				Ctrl + ;
클래스 뷰 닫기	Ctrl + /	Ctrl + /	Ctrl + /	Ctrl + E, C
클래스 뷰 숨기기	Ctrl + /	Ctrl + /	Ctrl + /	Ctrl + E, U
클래스 뷰 (클래스 뷰, 클래스, 클래스 뷰) 열기			Ctrl + M, M	
클래스 뷰 열기 및 클래스 뷰 열기				Ctrl + M, L
클래스 뷰 열기				Alt + S, W, T (클래스 뷰 열기)
클래스 뷰 열기				Ctrl + , Ctrl + M
클래스 뷰 열기				Ctrl + R, A
클래스 뷰 열기 및 클래스 뷰 열기				Ctrl + R, T
클래스 뷰 열기 및 클래스 뷰 열기				Alt + 1

## log4j2 설정 방법

- log4j2 설정
  - monitorInterval : 로그를 모니터링하는 시간 (기본값: 60초)
  - 링크 : <https://blog.daum.net/tingle/9032397>
  - java jar run [jar], [jar] config [jar] application.yml [jar]

□□.

- config□□□ jar □□□□□ □□□ □□□ □□.
- config □ □□□□□ □□□□ □□□ □□□ □□ (./config/\*/□□□)

▪ □□

- docker □□□ □□ □, config □□□ □□ □□□ □□□ □□.
- config □□□□□ application.yml □ □□□.
- application.yml □□ logging.config □ □□□□.
  - logging.config=config/log4j2.yml
- □□□ log4j2 □□□□ □□□ □□□.
  - monitorInterval □ □□□□.

▪ □□□

- □□□□ □□□ □□ □□
  - config □□□ □□ □□□ □□ docker □□
  - □□□□□□ □□□□□ □□□□□ □□
- □□□□ □□□ □□
  - docker □□ □, □□□□□ □□□ □□
  - □□ log4j2.yml □□ □□□□□ □□ □□
  - □□□ □□ □, □□□□□ □□□ □□□□ □□□□□ □□

---

# flatMap & concatMap & flatMapSequential

▪ flatMap

- □□/□□ □□□ □□□□ □□□.
- □□ □□□□



▪ concatMap

- □□/□□ □□ □□
- □□ □□□□



- `doOnSuccess()` : 실행할 코드, 콜백
- `onErrorReturn` : 실패 시 반환할 Return 값
  - 실패 시 반환할 값은 반드시 반환할 수 있는 값이어야 함
- `onErrorResume` : 실패 시 반환할 Flux 또는 Return 값
  - 실패 시 반환할 값은 반드시 반환할 수 있는 값이어야 함
- `onErrorContinue` : 실패 시 반환할 값은 `skip()` 또는 `skip()`.
  - 실패 시 반환할 값은 반드시 반환할 수 있는 값이어야 함

```

public Mono<Object> doExec(
    ServerHttpRequest request, ServerHttpResponse
    clientResponse, RbcAuth rbcAuth) {

    String finalAuthToken = ...;

    try{
        return webClient.get()
            .uri(rbcUrl.getCategories())
            .accept(MediaType.APPLICATION_JSON)
            .headers(httpHeaders ->
httpHeaders.setBearerAuth(finalAuthToken))
            .retrieve()
            .onStatus(HttpStatus::isError,
res -> doError(res))
            .bodyToMono(CustomResult.class)
            .doOnError(ex ->
doExcept(clientResponse, ex))
            .flatMap(res ->
doSucc(clientResponse, res))
            .doOnSuccess(res ->
doFinish(clientResponse, res))
            .subscribe()
            .log()
            ;
    }catch (Exception e){
        log.error("webClient fail. e={}",
e.getMessage(), e);
        return doExcept(clientResponse, e);
    }
}

```

```

private Mono<Object> doFinish(ServerHttpResponse response,
CustomResult res) {
    ...
    return Mono.just("finish");
}

private Mono<Object> doSucc(ServerHttpResponse response,
CustomResult res) {
    ...
    return Mono.just("success");
}

private Mono<Object> doExcept(ServerHttpResponse response,
Throwable ex) {
    ...
    return Mono.just("fail. e={}", e.getMessage());
}

private Mono<Exception> doError(ClientResponse res) {
    if (res.statusCode().isError()) {
        return Mono.error(new
RuntimeException(res.toString()));
    }
    else{
        return Mono.error(new
RuntimeException(res.toString()));
    }
}

```