

# Gson 라이브러리

# Gson

## ❖ Gson

- Java 객체와 JSON 문자열간의 변환 처리 라이브러리
- Javascript의 JSON 객체와 동일한 역할 제공
- 의존성

```
<dependency>  
  <groupId>com.google.code.gson</groupId>  
  <artifactId>gson</artifactId>  
  <version>2.8.5</version>  
</dependency>
```

- 사용법

```
// 객체 생성  
Gson gson = new Gson();  
  
// Java 객체 -> JSON 문자열 : toJson(인스턴스)  
String json = gson.toJson(obj);  
  
// JSON 문자열 -> Java 객체 : fromJson(json 문자열, Class)  
FamousSaying fs = gson.fromJson(json, FamousSaying.class);
```

# Gson

---

## ❖ FamousSaying.java

```
@Data
@AllArgsConstructor
@Builder
public class FamousSaying {
    private long id;
    private String statement;
    private String who;
}
```

# Gson

---

## ❖ 자바 객체의 JSON 문자열 변환 처리

```
public class ObjectToJson {  
  
    public static void main(String[] args) {  
        try {  
  
            Gson gson = new Gson();  
            FamousSaying fs = new FamousSaying(  
                1, "모르는게 약이다", "속담");  
  
            System.out.println(fs);  
            String json = gson.toJson(fs);  
            System.out.println(json);  
  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

# Gson

---

## ❖ json 파일 준비

- src/main/resources에 FamousSaying2.json 구성

```
{  
  "id": 0,  
  "statement": "큰 목표를 이루고 싶으면 허락을 구하지 마라.",  
  "who": "미상"  
}
```

## ❖ FileUtil.java

- Common 프로젝트 연결 설정

```
public class FileUtil {  
    :  
    public static String readAll(String path) throws Exception {  
        StringBuilder sb = new StringBuilder();  
        try(  
            BufferedReader br = new BufferedReader(  
                new FileReader(path));  
        ){  
            int ch;  
            while((ch=br.read())!=-1) {  
                sb.append((char)ch);  
            }  
        }  
        return sb.toString();  
    }  
}
```

# Gson

---

## ❖ FileUtil.java

- Common 프로젝트 연결 설정

```
public static String readResource(String path) throws Exception {  
    path = FileUtil.class.getResource(path).getPath();  
    return FileUtil.readAll(path);  
}  
  
}
```

# Gson

---

## ❖ JSON 문자열에서 단일 객체 인스턴스 변환

```
public class JsonToObject {  
  
    public static void main(String[] args) {  
        try {  
            String json = FileUtil.readResource("/famoussaying2.json");  
  
            Gson gson = new Gson();  
  
            // 단일 객체 복원  
            FamousSaying fs = gson.fromJson(json, FamousSaying.class);  
  
            System.out.println(fs);  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```



# Gson

---

## ❖ JSON 문자열에서 배열 변환

```
public class JsonToArray {  
  
    public static void main(String[] args) {  
        try {  
            String json = FileUtil.readResource("/famoussaying.json");  
  
            Gson gson = new Gson();  
  
            // 배열로 복원하기  
            FamousSaying[] arr =  
                gson.fromJson(json, FamousSaying[].class);  
  
            System.out.println(Arrays.toString(arr));  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

# Gson

## ❖ JSON 문자열에서 List 변환

```
public class JsonToList {  
  
    public static void main(String[] args) {  
        try {  
            String json = FileUtil.readResource("/famoussaying.json");  
  
            Gson gson = new Gson();  
  
            Type collectionType =  
                (new TypeToken<ArrayList<FamousSaying>>() {}).getType();  
  
            List<FamousSaying> list = gson.fromJson(json, collectionType);  
  
            System.out.println(list);  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```