

38. JSON 형식으로 읽기 / 쓰기

- ① XML 처럼 구조적인 데이터를 표현하기 쉽다.
- ② tag 처럼 데이터를 구분하는 메타데이터가 있다.
- ③ text 형식으로 프로그래밍 언어가 쉽게 처리할 수 있다.

Open API 에서
많이 사용

JSON ← 표현

JavaScript

Object

Notation

property 명은 반드시 문자열
" "

예) {

"title": "제목",

"content": "내용",

"deadline": "2024-1-1"

}

예)

{

title: "제목",

'content': '내용',

"deadline": "2024-1-1"

}

* Gson의 Builder 패턴 - 복잡한 객체의 생성 과정을 캡슐화 (감추기)

① 간단한 객체

```
Gson gson = new Gson();
```

② builder 패턴

```
GsonBuilder builder = new GsonBuilder();  
builder.setDateFormat("yyyy-MM-dd");
```

```
Gson gson = builder.create();
```

Builder 객체

객체 생성이 필요한
필요한 설정

← 객체 생성

Gson 객체 + 날짜 처리 플러그인
+ :

이러한 객체는 복잡해서
만들어야 하는 경우,
복잡한 객체 생성 과정은
감추기 위해
임의 객체 (빌더)에 의뢰해서
객체를 얻는 기법.

* Gson 나 JsonSerializer 아답터

작성, 아답터

① registerTypeAdapter()

GsonBuilder

생성

Gson

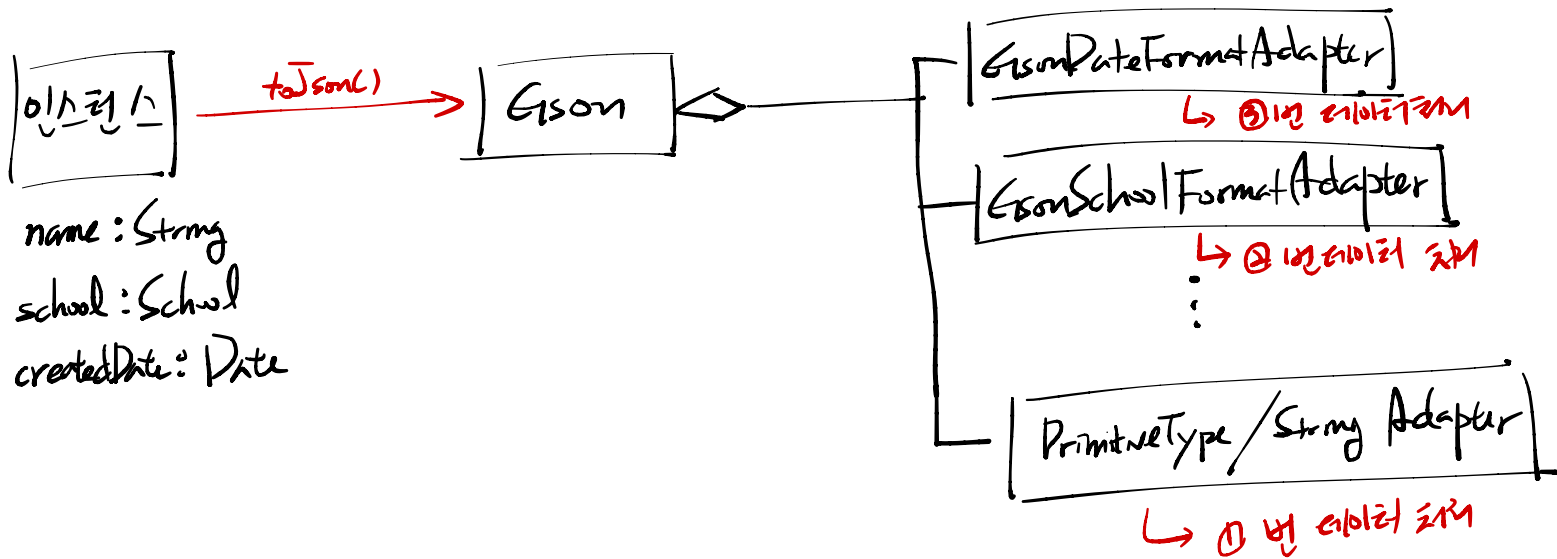
② Gson 객체에 아답터 등록

JsonSerializer<Date>

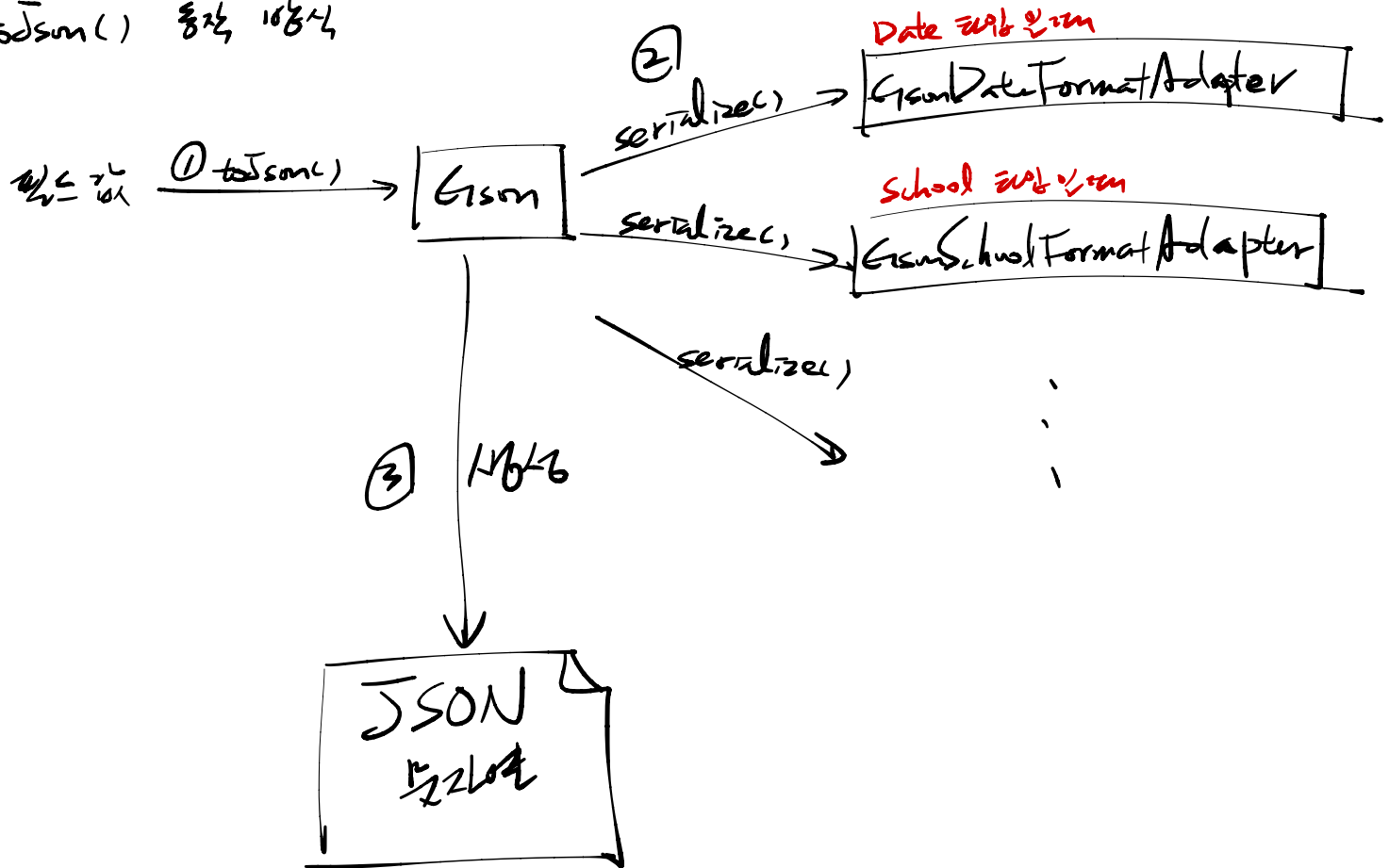
JsonSerializer<School>

⋮

* Gson 동작 방식

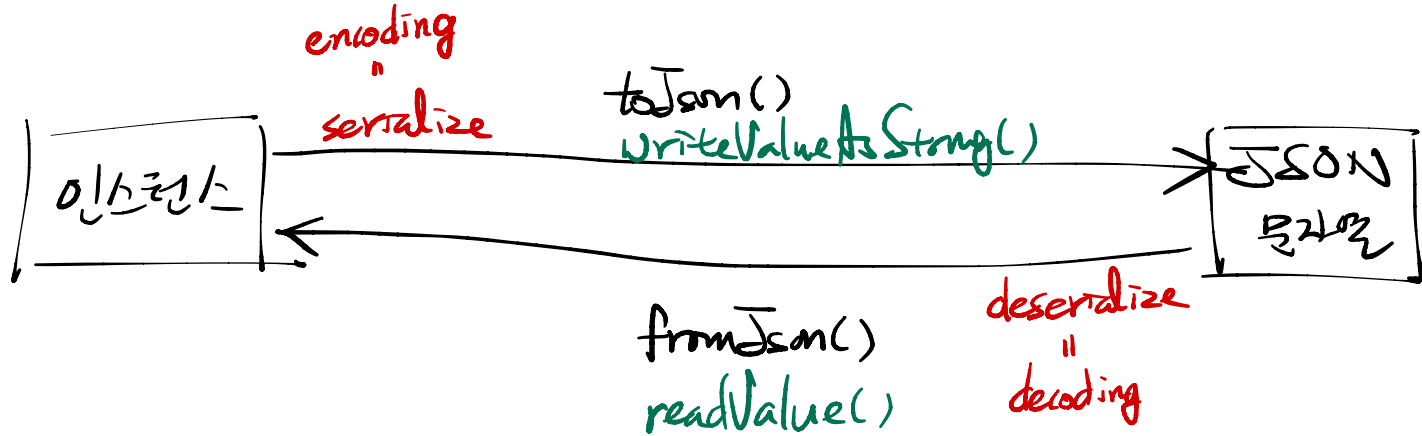


* toJson() 함수, 1884



* JSON serialize/deserialize


Gson



Jackson

* Predicate 인터페이스를 메서드 레퍼런스로 작성하기

`Predicate<String> obj = String::isEmpty;`



=



`Predicate<String> obj = (String value) -> { return value.isEmpty(); }`

