<http://allbegray.tistory.com/m/entry/JsonPath-08x-%EB%A5%BC-%EC%9D%B4%EC%9A%A9%ED%95%9C-Json-%ED%8C%8C%EC%8B%B1>

## JsonPath 0.8.x 를 이용한 Json 파싱

[*알버러지*](http://allbegray.tistory.com)

2013.07.30 23:48 [댓글수0](http://allbegray.tistory.com/m/entry/JsonPath-08x-%EB%A5%BC-%EC%9D%B4%EC%9A%A9%ED%95%9C-Json-%ED%8C%8C%EC%8B%B1#comment)

**1. JsonPath**

XML 문서에서 데이터를 추출해 낼 때 XPath 를 사용하는데

이와 유사하게 JSON 의 데이터를 추출해 낼 때 사용하는 라이브러리다.

Spring 3.2의 MVC Mock Test 관련을 보다가 눈에 띄여 간략하게 사용법을 알아보았다.

소스 저장소 : <https://github.com/jayway/JsonPath>

Test Case : <https://github.com/jayway/JsonPath/tree/master/json-path/src/test/java/com/jayway/jsonpath>

JsonPath expressions

|  |  |  |
| --- | --- | --- |
| **XPath** | **JSONPath** | **Description** |
| / | $ | the root object/element |
| . | @ | the current object/element |
| / | . or [] | child operator |
| .. | n/a | parent operator |
| // | .. | recursive descent. JSONPath borrows this syntax from E4X. |
| \* | \* | wildcard. All objects/elements regardless their names. |
| @ | n/a | attribute access. JSON structures don't have attributes. |
| [] | [] | subscript operator. XPath uses it to iterate over element collections and for [predicates](http://www.w3.org/TR/xpath#predicates). In Javascript and JSON it is the native array operator. |
| | | [,] | Union operator in XPath results in a combination of node sets. JSONPath allows alternate names or array indices as a set. |
| n/a | [start:end:step] | array slice operator borrowed from ES4. |
| [] | ?() | applies a filter (script) expression. |
| n/a | () | script expression, using the underlying script engine. |
| () | n/a | grouping in Xpath |

**2. 환경설정**

예제를 위해 jsonpath, jackson, net.sf.json-lib 라이브러리를 사용하였다.

Maven dependency

|  |
| --- |
| <dependency>  <groupId>net.sf.json-lib</groupId>  <artifactId>json-lib</artifactId>  <version>2.4</version>  <classifier>jdk15</classifier>  </dependency>  <dependency>  <groupId>com.jayway.jsonpath</groupId>  <artifactId>json-path</artifactId>  <version>0.8.1</version>  </dependency>  <dependency>  <groupId>org.codehaus.jackson</groupId>  <artifactId>jackson-mapper-asl</artifactId>  <version>1.9.12</version>  </dependency> |

**3. 예제 코드**

예제코드를 위해 샘플로 넣은 도서는 yes24 에서 랜덤으로 발췌하였다.

3.1. Json 문자열 샘플

|  |
| --- |
| package jsonpath;  import net.sf.json.JSONArray;  import net.sf.json.JSONObject;  public class Sample {    public static String DOCUMENT = null;    static {  JSONArray books = new JSONArray();    JSONObject book1 = new JSONObject();  book1.put("카테고리", "모바일");  book1.put("제목", "Do it! 안드로이드 앱 프로그래밍");  book1.put("저자", "정재곤");  book1.put("가격", 32000);  book1.put("ISBN-13", 9788997390205L);    JSONObject book2 = new JSONObject();  book2.put("카테고리", "자바");  book2.put("제목", "Java의 정석");  book2.put("저자", "남궁성");  book2.put("가격", 21000);    JSONObject book3 = new JSONObject();  book3.put("카테고리", "모바일");  book3.put("제목", "모던 웹 디자인을 위한 HTML5+CSS3 입문");  book3.put("저자", "윤인성");  book3.put("가격", 24000);    JSONObject book4 = new JSONObject();  book4.put("카테고리", "자바");  book4.put("제목", "난 정말 JAVA를 공부한적이 없다구요");  book4.put("저자", "윤성우");  book4.put("가격", 21000);  book4.put("ISBN-13", 9788996094029L);    books.add(book1);  books.add(book2);  books.add(book3);  books.add(book4);    JSONObject music = new JSONObject();  music.put("앨범", "Random Access Memories");  music.put("아티스트", "Daft Punk");  music.put("가격", 14900);    JSONObject root = new JSONObject();  root.put("도서", books);  root.put("음반", music);    DOCUMENT = root.toString();  }    } |

DOCUMENT 에 저장되는 문자열을 아래와 같다.

{

    "도서": [

        {

            "카테고리": "모바일",

            "제목": "Do it! 안드로이드 앱 프로그래밍",

            "저자": "정재곤",

            "가격": 32000,

            "ISBN-13": 9788997390205

        },

        {

            "카테고리": "자바",

            "제목": "Java의 정석",

            "저자": "남궁성",

            "가격": 21000

        },

        {

            "카테고리": "모바일",

            "제목": "모던 웹 디자인을 위한 HTML5+CSS3 입문",

            "저자": "윤인성",

            "가격": 24000

        },

        {

            "카테고리": "자바",

            "제목": "난 정말 JAVA를 공부한적이 없다구요",

            "저자": "윤성우",

            "가격": 21000,

            "ISBN-13": 9788996094029

        }

    ],

    "음반": {

        "앨범": "Random Access Memories",

        "아티스트": "Daft Punk",

        "가격": 14900

    }

}

3.2. JsonPath

|  |
| --- |
| package jsonpath;  import java.util.List;  import net.minidev.json.JSONArray;  import net.minidev.json.JSONObject;  import com.jayway.jsonpath.JsonPath;  /\*\*  \* json path, 리턴값의 다양한 예제  \*  \*/  public class JsonPathTest {  public static void main(String args[]) {  // 음반만 찾기  System.out.println(JsonPath.read(Sample.DOCUMENT, "$.음반"));    // 도서 안의 저자만 찾기  System.out.println(JsonPath.<List<String>>read(Sample.DOCUMENT, "$.도서[\*].저자"));    // 도서 안의 가격만 찾기  System.out.println(JsonPath.<List<Integer>>read(Sample.DOCUMENT, "$.도서[\*].가격"));    // 도서 안의 ISBN-13 만 찾기  System.out.println(JsonPath.<List<Long>>read(Sample.DOCUMENT, "$.도서[\*].ISBN-13"));    // 루트의 자식의 자식의 제목만 찾기  System.out.println(JsonPath.<List<String>>read(Sample.DOCUMENT, "$..제목"));    // ISBN-13 속성이 있는 도서만 찾기  System.out.println(JsonPath.<JSONArray>read(Sample.DOCUMENT, "$.도서[?(@.ISBN-13)]"));    // 가격이 30000 이상의 도서만 찾기  System.out.println(JsonPath.<JSONArray>read(Sample.DOCUMENT, "$.도서[?(@.가격>30000)]"));    // 제목이 Java의 정석 인 도서의 저자 찾기  System.out.println(JsonPath.<JSONArray>read(Sample.DOCUMENT, "$.도서[?(@.제목=='Java의 정석')].저자"));    // 도서중 가장 마지막 데이터 찾기  System.out.println(JsonPath.<JSONObject>read(Sample.DOCUMENT, "$.도서[-1:]"));    // 도서중 첫번째 두번째만 찾기  System.out.println(JsonPath.<List<Object>>read(Sample.DOCUMENT, "$.도서[0,1]"));  }    } |

결과 :

{"앨범":"Random Access Memories","가격":14900,"아티스트":"Daft Punk"}

["정재곤","남궁성","윤인성","윤성우"]

[32000,21000,24000,21000]

[9788997390205,9788996094029]

["Do it! 안드로이드 앱 프로그래밍","Java의 정석","모던 웹 디자인을 위한 HTML5+CSS3 입문","난 정말 JAVA를 공부한적이 없다구요"]

[{"저자":"정재곤","제목":"Do it! 안드로이드 앱 프로그래밍","카테고리":"모바일","ISBN-13":9788997390205,"가격":32000},{"저자":"윤성우","제목":"난 정말 JAVA를 공부한적이 없다구요","카테고리":"자바","ISBN-13":9788996094029,"가격":21000}]

[{"저자":"정재곤","제목":"Do it! 안드로이드 앱 프로그래밍","카테고리":"모바일","ISBN-13":9788997390205,"가격":32000}]

["남궁성"]

{"저자":"윤성우","제목":"난 정말 JAVA를 공부한적이 없다구요","카테고리":"자바","ISBN-13":9788996094029,"가격":21000}

[{"저자":"정재곤","제목":"Do it! 안드로이드 앱 프로그래밍","카테고리":"모바일","ISBN-13":9788997390205,"가격":32000},{"저자":"남궁성","제목":"Java의 정석","카테고리":"자바","가격":21000}]

3.3. Transformer

Transformer 를 통해 데이터를 일괄 수정해보도록 하자.

샘플 json 데이터를 알아보기 쉽게 속성명을 한글로 하다보니 모델의 클래스, 변수명까지 한글로 하였다...;

|  |
| --- |
| package jsonpath;  import java.util.List;  import java.util.Map;  import com.jayway.jsonpath.JsonModel;  import com.jayway.jsonpath.Transformer;  /\*\*  \* 샘플 json 데이터를 알아보기 쉽게 속성명을 한글로 하다보니 모델의 클래스, 변수명까지 한글로 했으니 이해하시길...  \*/  public class ConvertTest {    /\*\*  \* 음반을 찾은 후 가격 속성을 지우고 발매일 속성을 추가  \*/  private static void convert() {  JsonModel model = JsonModel.model(Sample.DOCUMENT);  Transformer<Map<String, Object>> transformer = new Transformer<Map<String, Object>>() {  public Object transform(Map<String, Object> map) {  map.remove("가격");  map.put("발매일", "2013.05.21");  return map;  }  };  음반 item = model.opsForObject("$.음반").transform(transformer).to(음반.class);  System.out.println(item);  }    /\*\*  \* 도서를 찾아 ISBN-13 속성을 지우고 저자 속성을 일괄 수정  \*/  private static void convertEach() {  JsonModel model = JsonModel.model(Sample.DOCUMENT);  Transformer<Object> transformer = new Transformer<Object>() {  public Object transform(Object obj) {  Map<String, Object> map = (Map<String, Object>) obj;  map.remove("ISBN-13");  map.put("저자", "익명");  return map;  }  };  List<도서> list = model.opsForArray("$.도서").each(transformer).toList().of(도서.class);  System.out.println(list);  }      public static class 도서 {  public String 저자;  public String 제목;  public String 카테고리;  public int 가격;  @Override  public String toString() {  return "도서 [저자=" + 저자 + ", 제목=" + 제목 + ", 카테고리=" + 카테고리 + ", 가격=" + 가격 + "]";  }  }    public static class 음반 {  public String 앨범;  public String 아티스트;  public String 발매일;    @Override  public String toString() {  return "음반 [앨범=" + 앨범 + ", 아티스트=" + 아티스트 + ", 발매일=" + 발매일 + "]";  }  }    public static void main(String args[]) {  convert();  convertEach();  }    } |

결과 :

음반 [앨범=Random Access Memories, 아티스트=Daft Punk, 발매일=2013.05.21]

[도서 [저자=익명, 제목=Do it! 안드로이드 앱 프로그래밍, 카테고리=모바일, 가격=32000], 도서 [저자=익명, 제목=Java의 정석, 카테고리=자바, 가격=21000], 도서 [저자=익명, 제목=모던 웹 디자인을 위한 HTML5+CSS3 입문, 카테고리=모바일, 가격=24000], 도서 [저자=익명, 제목=난 정말 JAVA를 공부한적이 없다구요, 카테고리=자바, 가격=21000]]

**관련 태그 목록** [#json](http://allbegray.tistory.com/m/tag/json) [#jsonpath](http://allbegray.tistory.com/m/tag/jsonpath) [#Parser](http://allbegray.tistory.com/m/tag/Parser) [#태그를 입력해 주세요.](http://allbegray.tistory.com/m/tag/%ED%83%9C%EA%B7%B8%EB%A5%BC%20%EC%9E%85%EB%A0%A5%ED%95%B4%20%EC%A3%BC%EC%84%B8%EC%9A%94.)