Tagazine

**작성일자** : 2013. 10. 25

**버전명** : ver.0.1

프로젝트팀

이주희(기획)김새하(디자인)노경래(서버)김현찬(개발)

**Document History**

개정이력

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 버전 | 개정일자 | 작성자 | 개정내용 | 승인자 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| 0.2 | 2013-10-29 | 노경래 | 추가된 테이블에 대한 요청 추가 |  |
| 0.1 | 2013-10-25 | 노경래 | 최초작성 |  |

Tagazine 통신 프로토콜 정의서

**(통신 기준에 대한 기본 내용 작성)**

**REST(Representational State Transfer)**

WOA(Web Oriented Architecture)가 결국은 RESTful 아키텍처를 의미하고, 클라이언트가 서비스(Endpoint)를 요청할 때 사용되는 클라이언트 – 서버 아키텍처 스타일 중의 하나입니다. Web이 바로 REST 아키텍처 스타일을 그대로 따르고 있는 시스템 입니다. RESTful의 특성은 아래와 같습니다.

* 사용자 에이전트가 자원과 상호작용하는데, 자원은 URI로 표현될 수 있는 모든 것들이 대상입니다.
* 자원과의 상호작용은 HTTP의 표준 동사(GET, POST, PUT, and DELETE)를 사용하여 이루어집니다. 또한, 상호작용을 위해 자원의 미디어타입이 결정되는데 HTTP **Content-Type** 헤더를 사용합니다. (XHTML, JSON, JPG, PNG, and JSON are some well-known media types.)
* 자원은 필요한 모든 정보를 다 포함하고 있습니다. (서비스가 stateless 형태로 존재)
* 자원은 다른 자원과의 링크를 포함합니다.

장점들을 나열하면 아래와 같습니다.

* **Caching** HTTP 동사를 통해 RESTfu endpoint가 불리워 질 때, HTTP 동사 GET을 사용합니다. GET으로 불리워진 자원들은 여러 다양한 방법으로 캐싱이 가능합니다. 결국 속도와 확장성이 향상됩니다.
* **Scale-Out** REST는 각 자원이 특정 요청에 필요한 모든 상태 값을 다 가지고 있기 때문에, 즉 stateless기 때문에 확장이 훨씬 용이합니다.
* **Idempotent** GET을 이용해 자원을 요청할 때 추가적인 부작용이 벌어지지 않습니다 (즉, 다시 한 번 GET, DELETE 등을 불러도 두번 중복작업이 이루어지지 않음)
* **Interoperability** SOAP이 클라이언트 서버 프로그램을 통해 상호운용성이 뛰어나다고 하지만, 몇 몇 언어 및 환경은 여전히 SOAP 툴킷을 갖고 있지 못하고, 툴킷을 가지고 있는 언어의 경우에도 최신 표중르 가진 다른 툴킷과 통신할 수 없는 경우가 종종 발생합니다. REST는 HTTP 동사만을 사용하기 때문에 어떤 언어, 플랫폼과도 상호운용이 가능합니다.
* **Simplicity** 너무 단순하게 구현이 가능합니다.

# 클라이언트에서 서버 호출 방법

서버와 클라이언트의 메시지 데이터 송수신은 HTTP 를 사용하고, JSON과 JSON형식의 REST Protocol을 사용하여 필요한 데이터를 요청하거나 전송한다.

## JSON Protocol 사용하는 방법

**JSON** (JavaScript Object Notation)은 경량의 DATA-교환 형식이다. 이 형식은 사람이 읽고 쓰기에 용이하며, 기계가 분석하고 생성함에도 용이하다. [JavaScript Programming Language](http://www.crockford.com/javascript), [Standard ECMA-262 3rd Edition - December 1999](http://www.ecma-international.org/publications/files/ecma-st/ECMA-262.pdf)의 일부에 토대를 두고 있다. JSON은 완벽하게 언어로부터 독립적이지만 C-family 언어 - C, C++, C#, Java, JavaScript, Perl, Python 그 외 다수의 프로그래머들에게 친숙한 관습을 사용하는 텍스트 형식이다. 이러한 속성들이 JSON을 이상적인 DATA교환 언어로 만들고 있다.

JSON은 두개의 구조를 기본으로 두고 있다:

* name/value 형태의 쌍으로 collection 타입. 다양한 언어들에서, 이는 *object*, record, struct(구조체), dictionary, hash table, 키가 있는 list, 또는 연상배열로서 실현 되었다.
* 값들의 순서화된 리스트. 대부분의 언어들에서, 이는 *array*, vector, list, 또는 sequence로서 실현 되었다.

이러한 것들은 보편적인 DATA 구조이다. 사실상 모든 현대의 프로그래밍 언어들은 어떠한 형태로든 이것들을 지원한다.

**기본 요청 방식 정의**

# 에러처리

클라이언트의 요청에 정상적인 응답이 불가능한 경우, 시스템의 예외 혹은 이미 정의된 메시지를 클라이언트에게 전달한다. 클라이언트는 HTTP 프로토콜의 상태 값이 999 값을 가진다면 사용자 정의에 의한 예외 처리로 인지 하여야 한다. 따라서 클라이언트는 HTTP 상태 값이 200인 경우에만 정상 처리하여야 한다.

에러메시지 구조:

* error
  + code: 정의 메시지 코드(정의되지 않은 메시지는 9000)
  + message: 예외 및 에러에 대한 메시지

JSON payload sample:

<?JSON version="1.0" encoding="UTF-8" standalone="yes" ?>

<error>

<code>에러코드</code>

<message>ID 인증에 실패함</message>

</error>

JSON payload sample:

{

"error":{

"code":"에러코드",

"message":" ID 인증에 실패함 "

}

}

# 날자 형 데이터 처리

날자 형식의 데이터 처리를 위해서 모든 전문에서는 직렬화 되어 있는 문자열을 사용한다. 로케일로 인한 날자 오류를 방지 하기 위함과, 가 독성을 위함이다.

* 일자: 20100213
* 일시: 20100213120000
* 시간: 120000

# 프로젝트 팀 Protocols

|  |  |  |  |
| --- | --- | --- | --- |
| 업무구분 | 항목 | URI | 비고 |
|  | 로그인 | /login |  |
|  | 로그아웃 | /logout |  |
| 회원관리 | 회원가입 | /member/join |  |
|  | 회원정보확인 | /member/userId |  |
|  | 회원탈퇴 | /member/quit |  |
|  | 회원ID확인 | /member/userId/check |  |
| 상품관리 | 게시물 등록 | /product/insert |  |
|  | 게시물 목록 얻기 | /product/list |  |
|  | 게시물 삭제 | /product/del |  |
|  | 좋아요 | /product/like |  |
|  | Tag 등록 | /product/tag |  |
|  | Tag 목록 얻기 | /product/tag/list |  |
|  | 댓글 등록 | /product/memo |  |
|  | 댓글 목록 얻기 | /product/memo/list |  |
|  | 댓글 삭제 | /product/memo/del |  |
| 메뉴관리 | 메뉴 목록 확인 | /menu/list |  |
|  |  |  |  |
| Magazine | Magazine 목록 | /magazine/list |  |
|  | Magazine 얻기 | /magazine/date |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **업무 기능** | **로그인** | 업무 ID | LOGIN |
| **Request URL** | /login | | |
| **Request**  **Query String** | USER\_ID : 사용자 ID  USER\_PW : 비밀번호  FACEBOOK : 페이스북 로그인 여부 | | |
| **Server Response JSON Declaration** | | | |
| ---성공시  {  “RESULT” : “SUCCESS”  }  -- 실패시  {  “RESULT” : “FAIL”  “ERR\_MSG” : “원인”  } | | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **업무 기능** | **로그아웃** | 업무 ID | LOGOUT |
| **Request URL** | /logout | | |
| **Request**  **Query String** | USER\_ID : 사용자 ID | | |
| **Server Response JSON Declaration** | | | |
| ---성공시  {  “RESULT” : “SUCCESS”  }  -- 실패시  {  “RESULT” : “FAIL”  “ERR\_MSG” : “원인”  } | | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **업무 기능** | **회원가입** | 업무 ID | MEMBER\_JOIN |
| **Request URL** | /member/join | | |
| **Request**  **Query String** | USER\_ID : 사용자 ID  USER\_PW : 비밀번호  FACEBOOK : 페이스북 로그인 여부 | | |
| **Server Response JSON Declaration** | | | |
| ---성공시  {  “RESULT” : “SUCCESS”  }  -- 실패시  {  “RESULT” : “FAIL”  “ERR\_MSG” : “원인”  } | | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **업무 기능** | **회원정보 확인** | 업무 ID | MEMBER\_INFO |
| **Request URL** | /member/info | | |
| **Request**  **Query String** | USER\_ID : 사용자 ID | | |
| **Server Response JSON Declaration** | | | |
| ---성공시  {  “RESULT” : “SUCCESS”  “USER\_PW”: “USER\_PW”  “FACEBOOK”: BOOL  }  -- 실패시  {  “RESULT” : “FAIL”  “ERR\_MSG” : “원인”  } | | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **업무 기능** | **회원탈퇴** | 업무 ID | MEMBER\_QUIT |
| **Request URL** | /member/quit | | |
| **Request**  **Query String** | USER\_ID : 사용자 ID | | |
| **Server Response JSON Declaration** | | | |
| ---성공시  {  “RESULT” : “SUCCESS”  }  -- 실패시  {  “RESULT” : “FAIL”  “ERR\_MSG” : “원인”  } | | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **업무 기능** | **회원 id 확인** | 업무 ID | MEMBER\_USERID\_CHECK |
| **Request URL** | /member/userId/Check | | |
| **Request**  **Query String** | { USER\_ID : 사용자 ID} | | |
| **Server Response JSON Declaration** | | | |
| ---성공시  {  “RESULT” : “SUCCESS”  }  -- 실패시  {  “RESULT” : “FAIL”  “ERR\_MSG” : “원인”  } | | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **업무 기능** | **게시물 등록** | 업무 ID | PRODUCT\_INSERT |
| **Request URL** | /product/insert | | |
| **Request**  **Query String** | { USER\_ID : 댓글을 적은 유저의 ID  IMAGE : 상품 이미지  COMMENT: 상품 정보  TAG: tag 정보 } | | |
| **Server Response JSON Declaration** | | | |
| ---성공시  {  “ RESULT” : “SUCCESS”  }  ---실패시  {  “RESULT” : “FAIL”  “ERR\_MSG” : “원인”  } | | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **업무 기능** | **게시물 목록 얻기** | 업무 ID | PRODUCT\_ LIST |
| **Request URL** | /product /list | | |
| **Request**  **Query String** | UPLOAD\_DATE : 얻을 목록 요청할 날짜 | | |
| **Server Response JSON Declaration** | | | |
| ---성공시  {  “RESULT” : “SUCCESS”  “TAG\_INFO” [  {  “PRODUCT\_ID”:productid,  “UPLOAD\_DATE”:upload date,  “USER\_ID”:user id,  “IMAGE\_PATH”:image path,  “VIEW\_CNT”:view count,  “LIKE\_CNT”:like count  {  “PRODUCT\_ID”:productid,  “UPLOAD\_DATE”:upload date,  “USER\_ID”:user id,  “IMAGE\_PATH”:image path,  “VIEW\_CNT”:view count,  “LIKE\_CNT”:like count  }  ]  }  -- 실패시  {  “RESULT” : “SUCCESS”  RESULT\_MSG” : “원인”  } | | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **업무 기능** | **게시물 삭제** | 업무 ID | PRODUCT\_DEL |
| **Request URL** | /product/del | | |
| **Request**  **Query String** | { USER\_ID : 댓글을 적은 유저의 ID  PRODUCT\_ID : 삭제할 게시물 ID} | | |
| **Server Response JSON Declaration** | | | |
| ---성공시  {  “ RESULT” : “SUCCESS”  }  ---실패시  {  “RESULT” : “FAIL”  “ERR\_MSG” : “원인”  } | | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **업무 기능** | **게시물 좋아요** | 업무 ID | PRODUCT\_LIKE |
| **Request URL** | /product/like | | |
| **Request**  **Query String** | { USER\_ID : 댓글을 적은 유저의 ID  PRODUCT\_ID : 좋아요 할 게시물 ID} | | |
| **Server Response JSON Declaration** | | | |
| ---성공시  {  “ RESULT” : “SUCCESS”  }  ---실패시  {  “RESULT” : “FAIL”  “ERR\_MSG” : “원인”  } | | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **업무 기능** | **게시물 좋아요 취소** | 업무 ID | PRODUCT\_LIKE\_DEL |
| **Request URL** | /product/like/del | | |
| **Request**  **Query String** | { USER\_ID : 댓글을 적은 유저의 ID  PRODUCT\_ID : 좋아요 한 게시물 ID} | | |
| **Server Response JSON Declaration** | | | |
| ---성공시  {  “ RESULT” : “SUCCESS”  }  ---실패시  {  “RESULT” : “FAIL”  “ERR\_MSG” : “원인”  } | | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **업무 기능** | **테그 등록** | 업무 ID | PRODUCT\_TAG |
| **Request URL** | /product/tag | | |
| **Request**  **Query String** | PRDUCT\_ID : 상품 ID  BRAND : 브랜드  TITLE : 제목  TAG\_KIND : 테그 종류  PRICE :가격  LOCATION : 테그 위치 | | |
| **Server Response JSON Declaration** | | | |
| ---성공시  {  “RESULT” : “SUCCESS”  }  -- 실패시  {  “RESULT” : “FAIL”  “ERR\_MSG” : “원인”  } | | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **업무 기능** | **Tag 목록 얻기** | 업무 ID | PRODUCT\_TAG\_LIST |
| **Request URL** | /product/tag/list | | |
| **Request**  **Query String** | PRDUCT\_ID : 상품 ID | | |
| **Server Response JSON Declaration** | | | |
| ---성공시  {  “RESULT” : “SUCCESS”,  “PRODUCT\_ID”:id,  “TAG\_INFO” [  {  “BRAND”:brand,  “TITLE”:title,  “TAG\_KIND”:tag\_kind,  “PRICE”:price,  “LOCATION”:location  {  “BRAND”:brand,  “TITLE”:title,  “TAG\_KIND”:tag\_kind,  “PRICE”:price,  “LOCATION”:location  }  ]  }  -- 실패시  {  “RESULT” : “FAIL”  “ERR\_MSG” : “원인”  } | | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **업무 기능** | **댓글 등록** | 업무 ID | PRODUCT\_MEMO |
| **Request URL** | /product/memo | | |
| **Request**  **Query String** | PRDUCT\_ID : 상품 ID  USER\_ID : 작성자  MEMO : 메모 내용 | | |
| **Server Response JSON Declaration** | | | |
| ---성공시  {  “RESULT” : “SUCCESS”  }  -- 실패시  {  “RESULT” : “SUCCESS”  “ERR\_MSG” : “원인”  } | | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **업무 기능** | **댓글 목록 얻기** | 업무 ID | PRODUCT\_MEMO\_LIST |
| **Request URL** | /product/memo/list | | |
| **Request**  **Query String** | PRDUCT\_ID : 상품 ID | | |
| **Server Response JSON Declaration** | | | |
| ---성공시  {  “RESULT” : “SUCCESS”  “PRODUCT\_ID”: product\_id  “COMMENT\_INFO” [  {  “USER\_ID”:user\_id,  “MEMO”:memo,  “UPLOAD\_DATE”:date  {  “USER\_ID”:user\_id,  “MEMO”:memo,  “UPLOAD\_DATE”:date  }  ]  }  -- 실패시  {  “RESULT” : “SUCCESS”  “ERR\_MSG” : “원인”  } | | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **업무 기능** | **댓글 삭제** | 업무 ID | PRODUCT\_MEMO\_DEL |
| **Request URL** | /product/memo/del | | |
| **Request**  **Query String** | PRDUCT\_ID : 상품 ID  USER\_ID : 작성자 | | |
| **Server Response JSON Declaration** | | | |
| ---성공시  {  “RESULT” : “SUCCESS”  }  -- 실패시  {  “RESULT” : “SUCCESS”  “ERR\_MSG” : “원인”  } | | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **업무 기능** | **메뉴 목록 확인** | 업무 ID | MENU\_LIST |
| **Request URL** | /menu/list | | |
| **Request**  **Query String** |  | | |
| **Server Response JSON Declaration** | | | |
| ---성공시  {  “RESULT” : “SUCCESS”  “TITLE”: “제목”  “KINDE” : “분류 제목”  “UPDATE\_DATE” : “업데이트 날짜”  }  -- 실패시  {  “RESULT” : “SUCCESS”  “ERR\_MSG” : “원인”  } | | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **업무 기능** | **magazine 목록 확인** | 업무 ID | MAGAZINE\_LIST |
| **Request URL** | /magazine/list | | |
| **Request**  **Query String** |  | | |
| **Server Response JSON Declaration** | | | |
| ---성공시  {  “RESULT” : “SUCCESS”  “TITLE”: “제목”  }  -- 실패시  {  “RESULT” : “SUCCESS”  “ERR\_MSG” : “원인”  } | | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **업무 기능** | **magazine 얻기** | 업무 ID | MAGAZINE\_DATE |
| **Request URL** | /magazine/date | | |
| **Request**  **Query String** |  | | |
| **Server Response JSON Declaration** | | | |
| ---성공시  {  “RESULT” : “SUCCESS”  “IMAGE\_PATH”: path  }  -- 실패시  {  “RESULT” : “SUCCESS”  “ERR\_MSG” : “원인”  } | | | |
|  | | | |