

# MEAN 스택 활용 웹 개발

## Node.js JSON 통신



2016년도 2학기 2학년 방과후학교

## 웹 서비스 vs 모바일 서비스

- 대표적인 웹 서비스로 HTTP 클라이언트인 웹 브라우저를 들 수 있다.
  - ✓ 웹 브라우저에서 주소를 요청하면 웹 서버는 해당 URL의 리소스(html, css, js)를 보내주어, 화면에서 렌더링한 결과를 보여준다.
- 모바일 클라이언트의 경우 자체 UI를 작성한다.
  - ✓ 문서 구조와 렌더링 정보를 포함하는 HTML이 필요하지 않다.
  - ✓ 모바일 기반 서버는 JSON, XML 기반의 데이터를 보내주고, 모바일 앱 화면에서는 이를 파싱하여 자체 UI로 출력해서 보여준다.
  - ✓ 모바일 앱을 위해 JSON, XML 데이터를 서비스해 주는 서비스를 API 서비스라고도 한다.

# JSON vs XML

```
{
  "melon":{
    "menuId":54020101,
    "count":10,
    "page":1,
    "totalPages":10,
    "rankDay":"20160730",
    "rankHour":"09",
    "songs":{
      "song":[
        {
          "songId":8254545,
          "songName":"Why So Lonely",
          "artists":{
            "artist":[
              {
                "artistId":217764,
                "artistName":"원더걸스"
              }
            ]
          },
          "albumId":2695552,
          "albumName":"Why So Lonely",
          "currentRank":1,
          "pastRank":1,
          "playTime":206,
          "issueDate":"20160705"
        }
      ]
    }
  }
}
```

```
<melon>
  <menuId>54020101</menuId>
  <count>10</count>
  <page>1</page>
  <totalPages>10</totalPages>
  <rankDay>20160730</rankDay>
  <rankHour>09</rankHour>
  <songs>
    <song>
      <songId>8254545</songId>
      <songName>Why So Lonely</songName>
      <artists>
        <artist>
          <artistId>217764</artistId>
          <artistName>원더걸스</artistName>
        </artist>
      </artists>
      <albumId>2695552</albumId>
      <albumName>Why So Lonely</albumName>
      <currentRank>1</currentRank>
      <pastRank>1</pastRank>
      <playTime>206</playTime>
      <issueDate>20160705</issueDate>
    </song>
  </songs>
</melon>
```

# JSON 다루기

- JSON 문서 생성 (JSON → 문자열)
  - ✓ `JSON.stringify( )`
- JSON 문서 분석 (문자열 → JSON)
  - ✓ `JSON.parse( )`

# JSON을 이용한 요청과 응답 처리

- 클라이언트 → 서버로 JSON 데이터 전송
  - ✓ Request에서 요청 메시지 분석하여 JSON 문자열 얻기
  - ✓ JSON 문자열을 JSON 객체로 변환 → `JSON.parse()` 이용
  - ✓ 내부 로직 처리
- 서버 → 클라이언트로 JSON 데이터 응답
  - ✓ 데이터를 JSON 객체로 생성
  - ✓ Response로 JSON 객체를 문자열로 변환 → `JSON.stringify()` 이용
  - ✓ 클라이언트로 전송
  - ✓ Content-Type: application/json

# REST 서비스

- REST 혹은 RESTful은 Representational State Transfer의 약자로
- 네트워크를 이용해서 필요한 정보에 접근하고, 그 정보인 리소스를 다루는 방법과 그러한 방법을 제공하는 아키텍처를 의미함
- REST 인터페이스 원칙
  - ✓ 자원 식별 : 개별 리소스 간에 식별할 수 있어야 한다. URL(URI)를 이용해서 네트워크를 통한 리소스를 식별할 수 있다.
  - ✓ 메시지로 리소스 조작 : 메시지에는 리소스를 다루는 정보를 포함할 수 있고, 이를 이용해서 리소스를 다룬다.
  - ✓ 자기 서술적 메시지 : 요청과 응답 메시지에는 메시지에 담은 리소스를 다룰 수 있는 정보를 포함한다.
  - ✓ 하이퍼미디어 : 하이퍼링크를 이용해서 문서간에 유기적으로 연결되어 있다. HTML은 다른 문서와 하이퍼링크로 연결되어서 간단하게 연관된 다른 문서에 접근할 수 있다.

# REST 기반 서비스 설계

## ■ 인터페이스 설계

- ✓ 리소스를 접근하는 URL 설계 - 명사 사용 권장, 목록은 복수형 명사  
: 간단하고 직관적인 API 설계 (예) `api.service.com/v1/items`
- ✓ 리소스를 얻어오는 조건은 쿼리 문자열을 이용  
: `api.service.com/v1/items?year=2016&name=kim`
- ✓ 리소스를 다루기 위한 HTTP 메소드 선택  
: GET(R), POST(C), PUT(U), DELETE(D)

## ■ 요청과 응답 메시지 설계

- ✓ 요청 메시지 구조
- ✓ 응답 메시지 구조 : 메타성 데이터 포함(데이터 개수, 요청 시간 등),  
에러 정보 제공(에러코드, 에러 메시지)

# 뮤직 REST 서비스 작성 (조회)

GET 127.0.0.1:3000/music

Authorization Headers Body Pre-request Script

Type

Body Cookies Headers (4) Tests

Pretty Raw Preview JSON

```
1 {
2   "count": 3,
3   "data": [
4     {
5       "id": 0,
6       "title": "Why So Lonely",
7       "singer": "원더걸스",
8       "members": [
9         "유빈",
10        "예은",
11        "선미",
12        "혜림"
13      ],
14       "year": 2016
15     },
16     {
17       "id": 1,
18       "title": "Day Day",
19       "singer": "BewhY",
20       "members": [],
21       "year": 2016
22     },
23     {
24       "id": 2,
25       "title": "너 그리고 나",
26       "singer": "여자친구",
27       "members": [
28         "소원",
29         "예린",
30         "은하",
31         "규주",
32         "신비",
33         "엄지"
34      ],
35       "year": 2015
36     }
37   ]
38 }
```

목록 조회

GET 127.0.0.1:3000/music/0

Authorization Headers Body Pre-re

Type

Body Cookies Headers (4) Tests

Pretty Raw Preview JSON

```
1 {
2   "id": 0,
3   "title": "Why So Lonely",
4   "singer": "원더걸스",
5   "members": [
6     "유빈",
7     "예은",
8     "선미",
9     "혜림"
10  ],
11   "year": 2016
12 }
```

상세 조회



# 뮤직 REST 서비스 작성 (등록)

POST 127.0.0.1:3000/music

Authorization Headers (1) Body Pre-request Script Tests

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 {  
2   "title": "뱅뱅뱅",  
3   "singer": "빅뱅",  
4   "members": ["GD", "TOP", "태양"],  
5   "year": 2015  
6 }
```

노래 등록

127.0.0.1:3000/music +

GET 127.0.0.1:3000/music

Body Cookies Headers (4) Tests

Pretty Raw Preview JSON

```
1 {  
2   "count": 4,  
3   "data": [  
4     {  
5       "id": 0,  
6       "title": "Why So Lonely",  
7       "singer": "원더걸스",  
8       "members": [  
9         "유빈",  
10        "예은",  
11        "선미",  
12        "혜림"  
13      ],  
14      "year": 2016  
15    },  
16    {  
17      "id": 1,  
18      "title": "Day Day",  
19      "singer": "BewhY",  
20      "members": [],  
21      "year": 2016  
22    },  
23    {  
24      "id": 2,  
25      "title": "너 그리고 나",  
26      "singer": "여자친구",  
27      "members": [  
28        "소원",  
29        "예린",  
30        "은하",  
31        "유주",  
32        "신비",  
33        "임지"  
34      ],  
35      "year": 2015  
36    },  
37    {  
38      "id": 4,  
39      "title": "뱅뱅뱅",  
40      "singer": "빅뱅",  
41      "members": [  
42        "GD",  
43        "TOP",  
44        "태양"  
45      ],  
46      "year": 2015  
47    }  
48  ]  
49 }
```

목록 조회

# 뮤직 REST 서비스 작성 (수정)

127.0.0.1:3000/music/4 +

PUT 127.0.0.1:3000/music/4

Authorization Headers (1) Body Pre-request Script Tests

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json)

```
1 {  
2   "title": "뱅뱅뱅2",  
3   "singer": "빅뱅2",  
4   "members": ["GD", "TOP", "태양", "승리", "대성"],  
5   "year": 2014  
6 }
```

노래 수정

127.0.0.1:3000/music/4 +

GET 127.0.0.1:3000/music/4

Authorization Headers (1) Body Pr

Type

Body Cookies Headers (4) Tests

Pretty Raw Preview JSON

```
1 {  
2   "id": 4,  
3   "title": "뱅뱅뱅2",  
4   "singer": "빅뱅2",  
5   "members": [  
6     "GD",  
7     "TOP",  
8     "태양",  
9     "승리",  
10    "대성"  
11  ],  
12   "year": 2014  
13 }
```

상세 조회

# 뮤직 REST 서비스 작성 (삭제)

127.0.0.1:3000/music/4

**DELETE** 127.0.0.1:3000/music/4

Authorization Headers (1) Body

Type

Body Cookies Headers (4) Tests

Pretty Raw Preview JSON

```
1 {
2   "id": 4,
3   "title": "뽕뽕뽕 2",
4   "singer": "빅 뽕 2",
5   "members": [
6     "GD",
7     "TOP",
8     "태양",
9     "승리",
10    "대성"
11  ],
12   "year": 2014
13 }
```

노래 삭제

127.0.0.1:3000/music

**GET** 127.0.0.1:3000/music

Authorization Headers Body Pre-re

Type

Body Cookies Headers (4) Tests

Pretty Raw Preview JSON

```
1 {
2   "count": 3,
3   "data": [
4     {
5       "id": 0,
6       "title": "Why So Lonely",
7       "singer": "원더걸스",
8       "members": [
9         "유빈",
10        "예은",
11        "선미",
12        "해림"
13      ],
14       "year": 2016
15     },
16     {
17       "id": 1,
18       "title": "Day Day",
19       "singer": "BewhY",
20       "members": [],
21       "year": 2016
22     },
23     {
24       "id": 2,
25       "title": "너 그리고 나",
26       "singer": "여자친구",
27       "members": [
28         "소원",
29         "예린",
30         "은하",
31         "유주",
32         "신비",
33         "엄지"
34       ],
35       "year": 2015
36     }
37   ]
38 }
```

목록 조회

# [실습] 영화 REST 서비스 작성 (조회)

GET 127.0.0.1:3000/movies

Authorization Headers Body Pre-request Script Tests

Type No A

Body Cookies Headers (4) Tests

Pretty Raw Preview JSON

```
1 {
2   "count": 3,
3   "data": [
4     {
5       "id": 0,
6       "title": "아바타",
7       "director": "제임스 카메론",
8       "year": 2009,
9       "synopsis": "인류의 마지막 희망, 행성 판도라! 이 곳
10                  독성을 지닌 대기로 인해 자원 획득에 어려움을 주
11     },
12     {
13       "id": 1,
14       "title": "스타워즈",
15       "director": "조지 루카스",
16       "year": 1977,
17       "synopsis": "평화를 던 은하계에서 타칸총독(피터 커슨
18                  죽음의 별 설계를 입수하려고 하지만 공주는 타
19     },
20     {
21       "id": 2,
22       "title": "인터스텔라",
23       "director": "크리스토퍼
24       "year": 2014,
25       "synopsis": "세계 각국의 정부와 경제가 완전히 붕괴
26       인류를 구해야 하는 임무가 지워진다. 사랑하는 가

목록 조회


```

GET 127.0.0.1:3000/movies/0

Authorization Headers Body Pre-request Script Tests

Type No

Body Cookies Headers (4) Tests

Pretty Raw Preview JSON

```
1 {
2   "id": 0,
3   "title": "아바타",
4   "director": "제임스 카메론",
5   "year": 2009,
6   "synopsis": "인류의 마지막 희망, 행성 판도라! 이 곳을
7   지닌 대기로 인해 자원 획득에 어려움을 겪게 된 인류
8 }
```

**상세 조회**

# [실습] 영화 REST 서비스 작성 (등록)

POST 127.0.0.1:3000/movies/0

Authorization Headers Body Pre-request Script Tests

form-data x-www-form-urlencoded raw binary Text

```
1 {
2   "title": "부산행",
3   "director": "연상호",
4   "year": 2016,
5   "synopsis": "정체 불명의 바이러스가 전국으로 확산되고 대한민국을
6   지키고 싶은, 지켜야만 하는 사람들의 극한의 사투!"
7 }
```

영화 등록

GET 127.0.0.1:3000/movies

Authorization Headers Body Pre-request Script

Type

Body Cookies Headers (4) Tests

Pretty Raw Preview JSON

```
1 {
2   "count": 4,
3   "data": [
4     {
5       "id": 0,
6       "title": "아바타",
7       "director": "제임스 카메론",
8       "year": 2009,
9       "synopsis": "인류의 마지막 희망, 한
10      독성을 지닌 대기로 인해 자원
11    },
12    {
13      "id": 1,
14      "title": "스타워즈",
15      "director": "조지 루카스",
16      "year": 1977,
17      "synopsis": "평화를던 은하계에서
18      죽음의 별 설계도를 입수하려고
19    },
20    {
21      "id": 2,
22      "title": "인터스텔라",
23      "director": "크리스토퍼 놀란",
24      "year": 2014,
25      "synopsis": "세계 각국의 정부와 경
26      인류를 구해야 하는 임무가 지워
27    },
28    {
29      "id": 3,
30      "title": "부산행",
31      "director": "연상호",
32      "year": 2016,
33      "synopsis": "정체 불명의 바이러스가
34      지키고 싶은, 지켜야만 하는 사
35    }
36  ]
37 }
```

목록 조회

# [실습] 영화 REST 서비스 작성 (수정)

PUT 127.0.0.1:3000/movies/3

Authorization Headers Body Pre-request S

☐ form-data ☐ x-www-form-urlencoded ☒ raw

```
1 {  
2   "title": "부산행2",  
3   "director": "연상호2",  
4   "year": 2015,  
5   "synopsis": "좀비 영화!!"  
6 }
```

영화 수정

GET 127.0.0.1:3000/movies/3

Authorization Headers Body Pre-request Scri

Type

Body Cookies Headers (4) Tests

Pretty Raw Preview JSON

```
1 {  
2   "id": 3,  
3   "title": "부산행2",  
4   "director": "연상호2",  
5   "year": 2015,  
6   "synopsis": "좀비 영화!!"  
7 }
```

상세 조회

# [실습] 영화 REST 서비스 작성 (삭제)

DELETE 127.0.0.1:3000/movies/3

Authorization Headers Body Pre-request Script

Type

Body Cookies Headers (4) Tests

Pretty Raw Preview JSON

```
1 {
2   "id": 3,
3   "title": "부산행2",
4   "director": "연상호2",
5   "year": 2015,
6   "synopsis": "좀비 영화!!"
7 }
```

영화 삭제

GET 127.0.0.1:3000/movies

Authorization Headers Body Pre-request Script

Type

Body Cookies Headers (4) Tests

Pretty Raw Preview JSON

```
1 {
2   "count": 3,
3   "data": [
4     {
5       "id": 0,
6       "title": "아바타",
7       "director": "제임스 카메론",
8       "year": 2009,
9       "synopsis": "인류의 마지막 희망, 행성
10                  독성을 지닌 대기로 인해 자원 획득
11     },
12     {
13       "id": 1,
14       "title": "스타워즈",
15       "director": "조지 루카스",
16       "year": 1977,
17       "synopsis": "평화를던 은하계에서 타칸
18                  죽음의 별 설계도를 입수하려고 하지
19     },
20     {
21       "id": 2,
22       "title": "인터스텔라",
23       "director": "크리스토퍼 놀란",
24       "year": 2014,
25       "synopsis": "세계 각국의 정부와 경제가
26                  인류를 구해야 하는 임무가 지워진다
27     }
28   ]
29 }
```

목록 조회