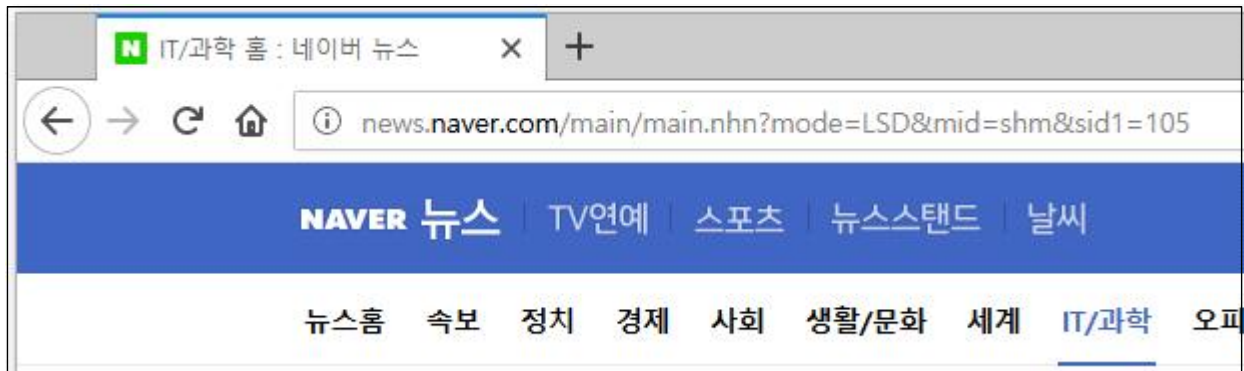


■ RESTful Web Service란?

1) Representational State transfer의 줄임말

(<https://ko.wikipedia.org/wiki/REST>)

2) 하나의 URI는 하나의 고유한 리소스를 대표할 수 있도록 uri를 설계해야 한다는 개념



- 변경 : <http://news.naver.com/it>

3) 네트워크 전체로 보면 다양한 의미가 있음

- uniform (유니폼 인터페이스)

Uniform Interface는 URI로 지정한 리소스에 대한 조작을 통일되고 한정적인 인터페이스로 수행하는 아키텍처 스타일

- Stateless (무상태성)

REST는 무상태성 성격을 갖습니다. 다시 말해 작업을 위한 상태정보를 따로 저장하고 관리하지 않음. 세션 정보나 쿠키정보를 별도로 저장하고 관리하지 않기 때문에 API 서버는 들어오는 요청만을 단순히 처리하면 됨. 때문에 서비스의 자유도가 높아지고 서버에서 불필요한 정보를 관리하지 않음으로써 구현이 단순해짐

- Cacheable (캐시 가능)

REST의 가장 큰 특징 중 하나는 HTTP라는 기존 웹표준을 그대로 사용하기 때문에, 웹에서 사용하는 기존 인프라를 그대로 활용이 가능. 따라서 HTTP가 가진 캐싱 기능이 적용 가능함. HTTP 프로토콜 표준에서 사용하는 Last-Modified태그나 E-Tag를 이용하면 캐싱 구현이 가능

- Self-descriptiveness (자체 표현 구조)

REST의 또 다른 큰 특징 중 하나는 REST API 메시지만 보고도 이를 쉽게 이해 할 수 있는 자체 표현 구조로 되어 있다는 것

- Client - Server 구조

REST 서버는 API 제공, 클라이언트는 사용자 인증이나 컨텍스트(세션, 로그인 정보)등을 직접 관리하는 구조로 각각의 역할이 확실히 구분되기 때문에 클라이언트와 서버에서 개발해야 할 내용이 명확해지고 서로간 의존성이 줄어들게 됨

- 계층형 구조

REST 서버는 다중 계층으로 구성될 수 있으며 보안, 로드 밸런싱, 암호화 계층을 추가해 구조상의 유연성을 둘 수 있고 PROXY, 게이트웨이 같은 네트워크 기반의 중간매체를 사용할 수 있게 함

- 하지만 개발자 관점에서는 RESTAPI 설계시 **URI**와 **메서드**만 잘 생각하면 됨

4) 리소스(자원) / 메서드(행위) / 메세지(표현)으로 분류

- 음식점, 게임대회, 글, 서비스, 여행지, 카페, 유저, 드라마 : 리소스(자원)
 - 음식점을 등록하다 / 불러오다 / 수정하다 / 삭제하다 : 메서드(행위)
 - 7번 음식점 이름을 '토끼정'으로 변경하다 : 메세지(표현)
- (예 : {"name", "토끼정"})

5) 리소스(자원)는 uri로(리소스는 동사보다는 명사를 사용)

- 음식점의 경우 : http://localhost/restaurant
- 게임대회의 경우 : http://localhost/grandprix

5) 메서드는 HTTP프로토콜의 메서드로

메서드 이름	의미
GET	얻어오기(SELECT) / 일반적인 HTML view
POST	입력(INSERT)
PUT	수정(UPDATE)
DELETE	삭제(DELETE)

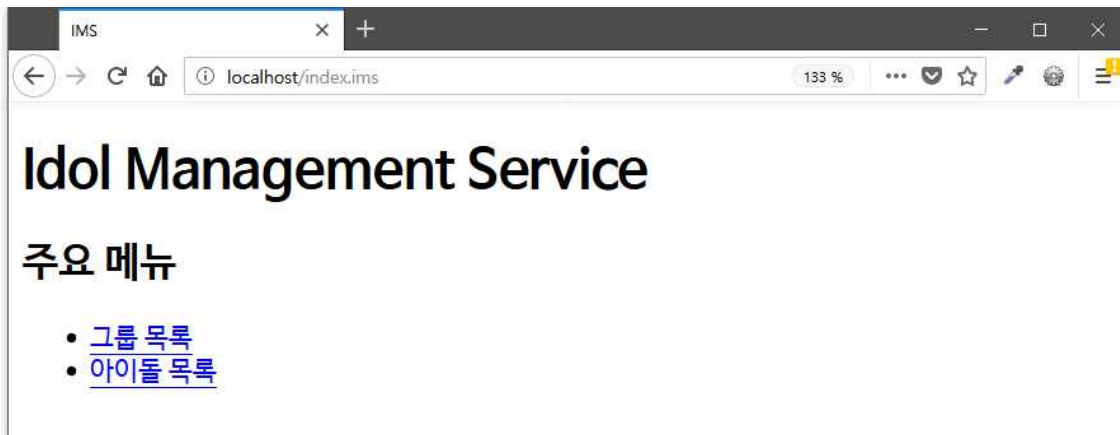
6) 메세지는 JSON으로

7) 예

URI	메서드	설명
http://localhost/restaurant	GET	음식점리스트 페이지
http://localhost/restaurant	POST	음식점 입력
http://localhost/restaurant	PUT	없음(있다면 전부 수정)
http://localhost/restaurant	DELETE	없음(있다면 전부 지우기)
http://localhost/restaurant/7	GET	7번 음식점 상세페이지
http://localhost/restaurant/7	POST	없음
http://localhost/restaurant/7	PUT	7번 음식점 수정
http://localhost/restaurant/7	DELETE	7번 음식점 삭제

7) 우리가 만든 IMS(Idol Management Service)를 변경한다면

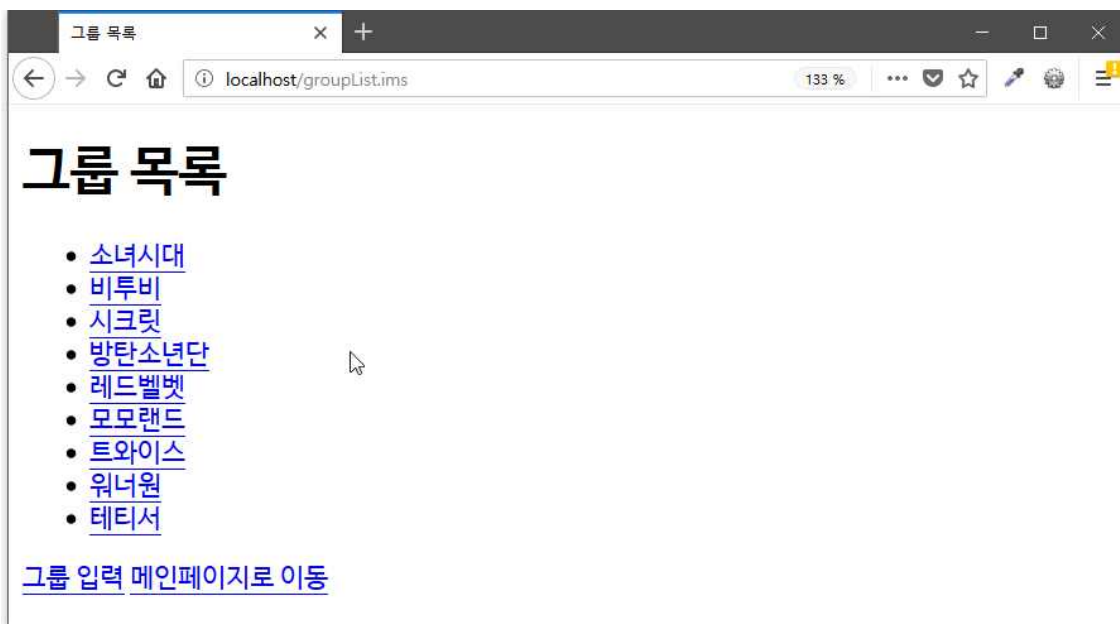
- 메인 페이지



변경 전 : GET /index.ims

변경 후 : GET /index

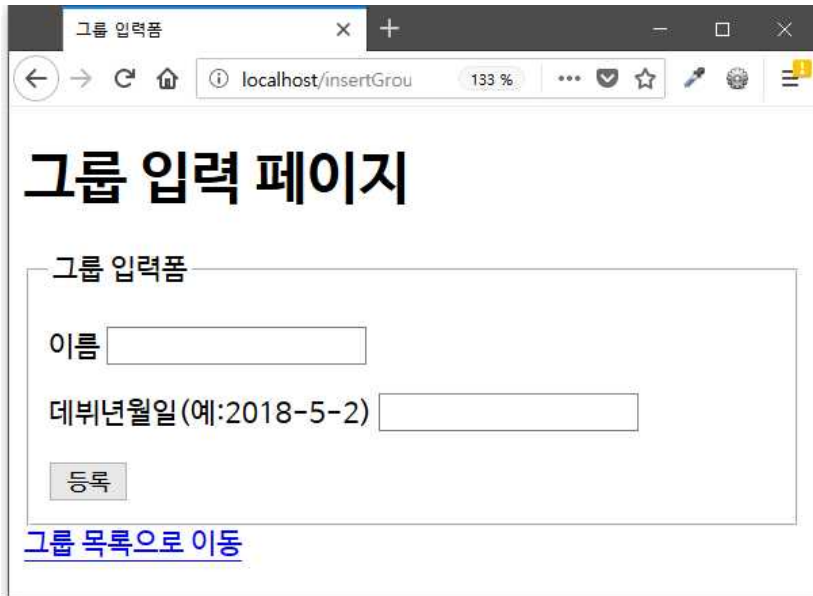
- 그룹목록 페이지



변경 전 : GET /groupList.ims

변경 후 : GET /group

- 그룹입력폼 페이지



그룹 입력 폼

이름

데뷔년월일(예:2018-5-2)

[그룹 목록으로 이동](#)

변경 전 : GET /insertGroup.ims

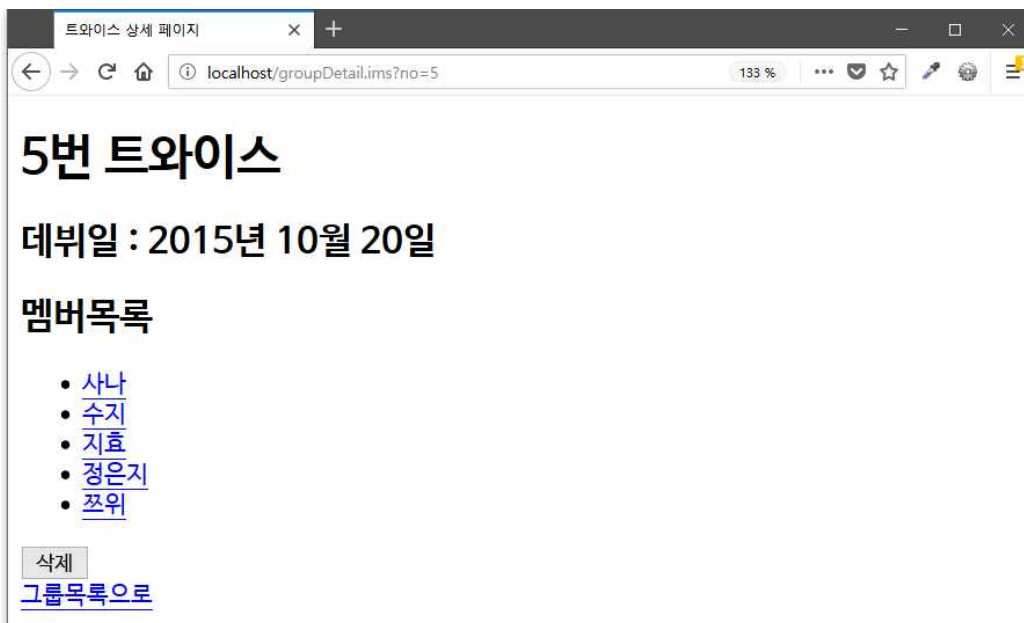
변경 후 : GET /group/register

- 그룹입력(뷰가 아님)

변경 전 : POST /insertGroup.ims

변경 후 : POST /group

- 그룹상세 페이지



트와이스 상세 페이지

5번 트와이스

데뷔일 : 2015년 10월 20일

멤버목록

- 사나
- 수지
- 지효
- 정은지
- 썬위

[그룹목록으로](#)

변경 전 : GET /groupDetail.ims?no=5

변경 후 : GET /group/5

- 그룹삭제 (뷰가 아님)

변경 전 : GET /deleteGroup.ims?no=5

변경 후 : DELETE /group/5

- 그룹수정 페이지

그룹 수정 폼

이름

데뷔년월일(예:2018-5-2)

[수정 취소](#)

변경 전 : GET /updateGroup.ims?no=1

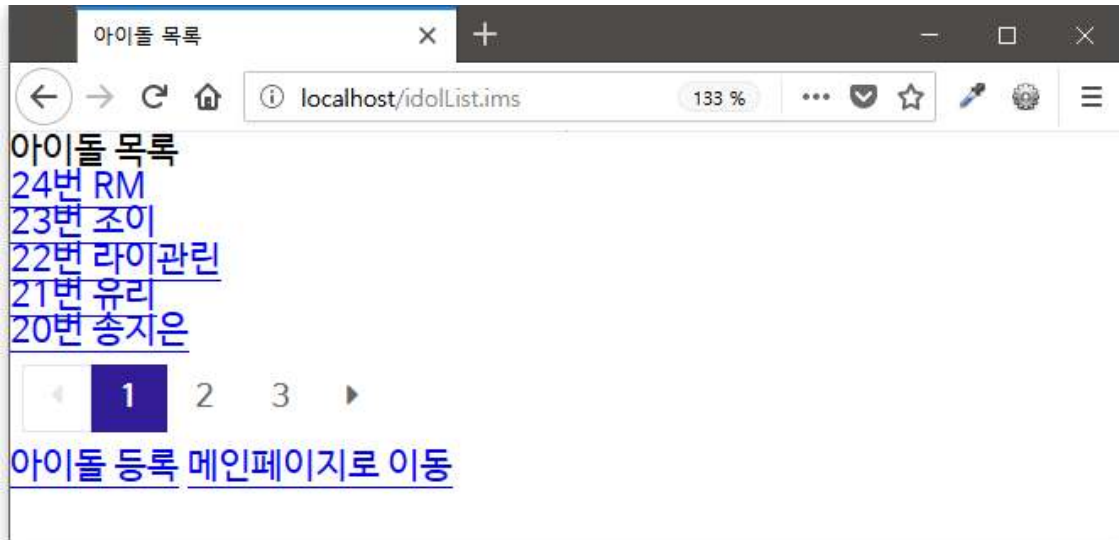
변경 후 : GET /group/1/update

- 그룹 수정(뷰 아님)

변경 전 : POST /updateGroup.ims?no=1

변경 후 : PUT /group/

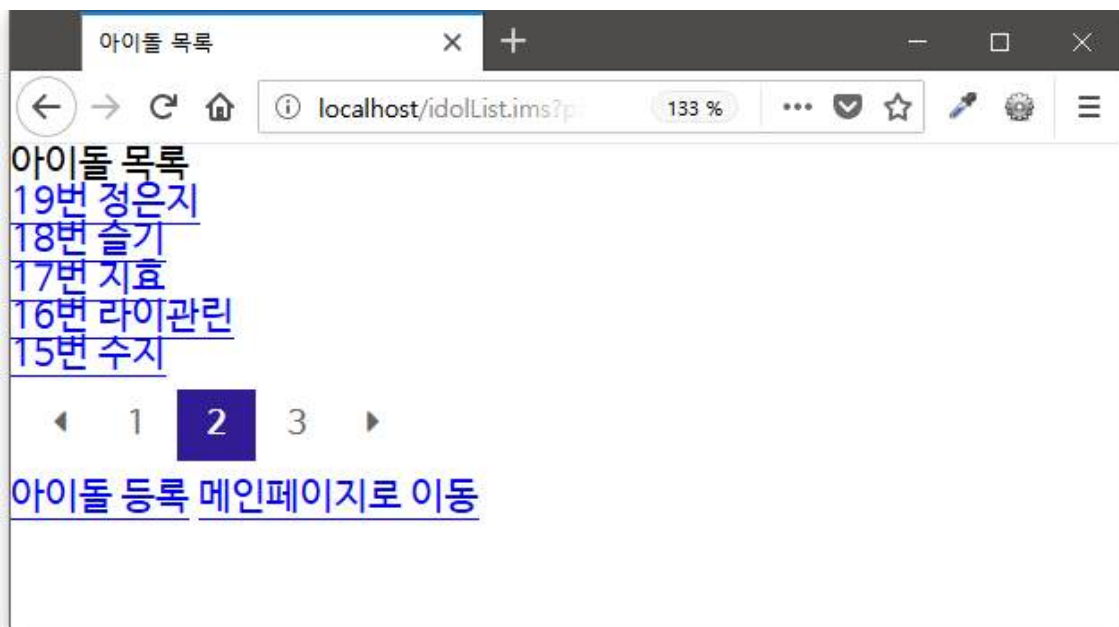
- 아이돌 목록 페이지



변경 전 : GET /idolList.ims?page=1

변경 후 : GET /idol/page/1

- 아이돌 목록 2페이지



변경 전 : GET /idolList.ims?page=2

변경 후 : GET /idol/page/2

- 아이돌 입력 페이지

아이돌 입력 폼

이름

키

몸무게

생년월일(예:2018-5-2)

☐ 소녀시대 ☐ 비투비 ☐ 시크릿 ☐ 방탄소년단 ☐ 레드벨벳
☐ 모모랜드 ☐ 트와이스 ☐ 워너원 ☐ 테티서

[아이돌 목록으로 이동](#)

변경 전 : GET /insertIdol.ims

변경 후 : GET /idol/register

- 아이돌 입력(뷰 아님)

변경 전 : POST /insertIdol.ims

변경 후 : POST /idol

- 아이돌 상세 페이지

RM의 상세정보

이름 RM

키 181

몸무게 64

생년월일 1994-09-12

그룹 방탄소년단

[수정](#)

[목록으로](#)

변경 전 : GET /idolDetail.ims?no=24

변경 후 : GET /idol/24

- 아이돌 삭제(뷰 아님)

변경 전 : POST /deleteIdol.ims?no=24

변경 후 : DELETE /idol/24

- 아이돌 수정폼 페이지

아이돌 수정 폼

이름 RM

키 181

몸무게 64

생년월일(예:2018-5-2) 1994-09-12

☐ 소녀시대 ☐ 비투비 ☐ 시크릿 ☒ 방탄소년단 ☐ 레드벨벳
☐ 모모랜드 ☐ 트와이스 ☐ 워너원 ☐ 테티서

아이돌 수정

[상세페이지로 이동](#)

변경 전 : GET /updateIdol.ims?no=24

변경 후 : GET /idol/24/update

- 아이돌 수정(뷰 아님)

변경 전 : POST /updateIdol.ims?no=24

변경 후 : PUT /idol/24

- 여기에 로그인(뷰 아님)

변경 전 : POST /log.ims

변경 후 : POST /session

- 로그아웃(뷰 아님)

변경 전 : GET /log.ims

변경 후 : DELETE /session

■ RESTful Web Service로 설정

1) web.xml에서 DispatcherServlet 설정

```
<servlet>
    <servlet-name>phone</servlet-name>

    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
</servlet>

<servlet-mapping>
    <servlet-name>phone</servlet-name>
    <url-pattern>/</url-pattern>
</servlet-mapping>
```

- 이 경우 모든 요청이 DispatcherServlet으로 들어오기 때문에 css나 js등 적용이 안됨(servlet.xml에 리소스 설정)

```
<mvc:resources location="/css/" mapping="/css/**"/>
<mvc:resources location="/img/" mapping="/img/**"/>
<mvc:resources location="/js/" mapping="/js/**"/>
<mvc:resources location="/fonts/" mapping="/fonts/**"/>
```

- 브라우저의 폼에서 PUT / DELETE 작동하지 않음
- 거의 모든 프레임워크들이 POST를 변경하여 사용(hidden 필드 추가)

```
<input type="hidden" name="_method" value="PUT">
<input type="hidden" name="_method" value="DELETE">
```

- 스프링의 필터를 이용하여 PUT / DELETE도 사용 가능하게 변경

```
<filter>
    <filter-name>httpMethodFilter</filter-name>

    <filter-class>org.springframework.web.filter.HiddenHttpMethodFilter</filter-class>
</filter>

<filter-mapping>
    <filter-name>httpMethodFilter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```

- delete / update의 경우 요청의 파라미터가 아니라 url경로로 넘어옴

이전방식 : http://localhost/update.html?no=12
변경방식 : http://localhost/article/12

- 이 경우 요청의 파라미터가 아니기 때문에

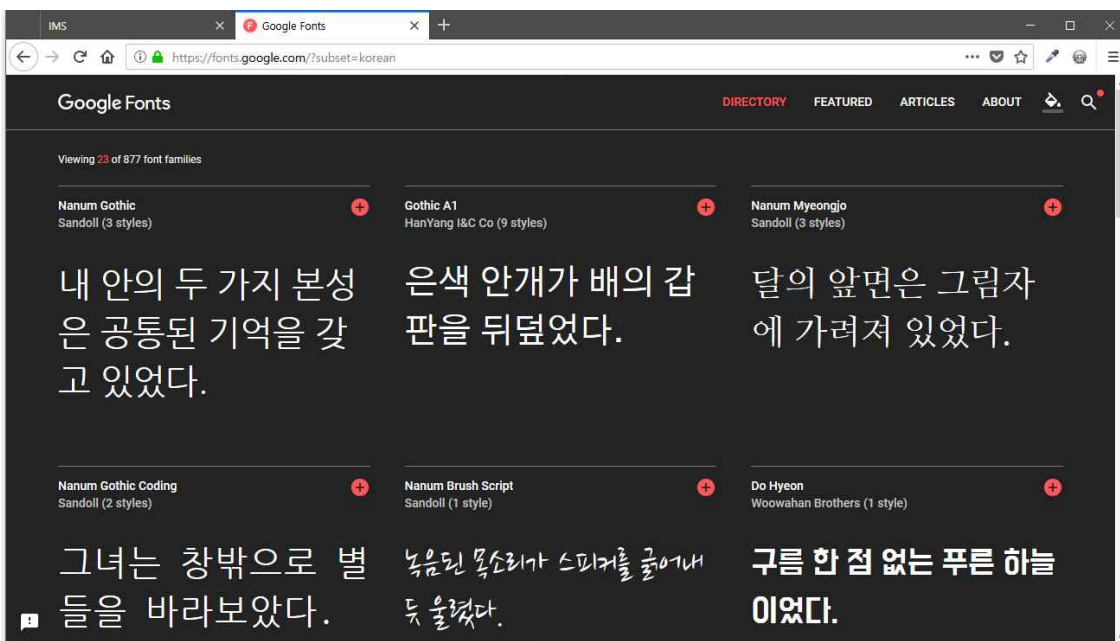
<pre>@RequestMapping(value="/update/{no}", method=RequestMethod.GET) public String updateForm(Model model, @PathVariable int no) {</pre>
--

■ IMS를 업데이트

1) 템플릿을 적용



- 구글 웹폰트에서 새로운 한글폰트 적용(early access가 아님 : 정식으로 변경)

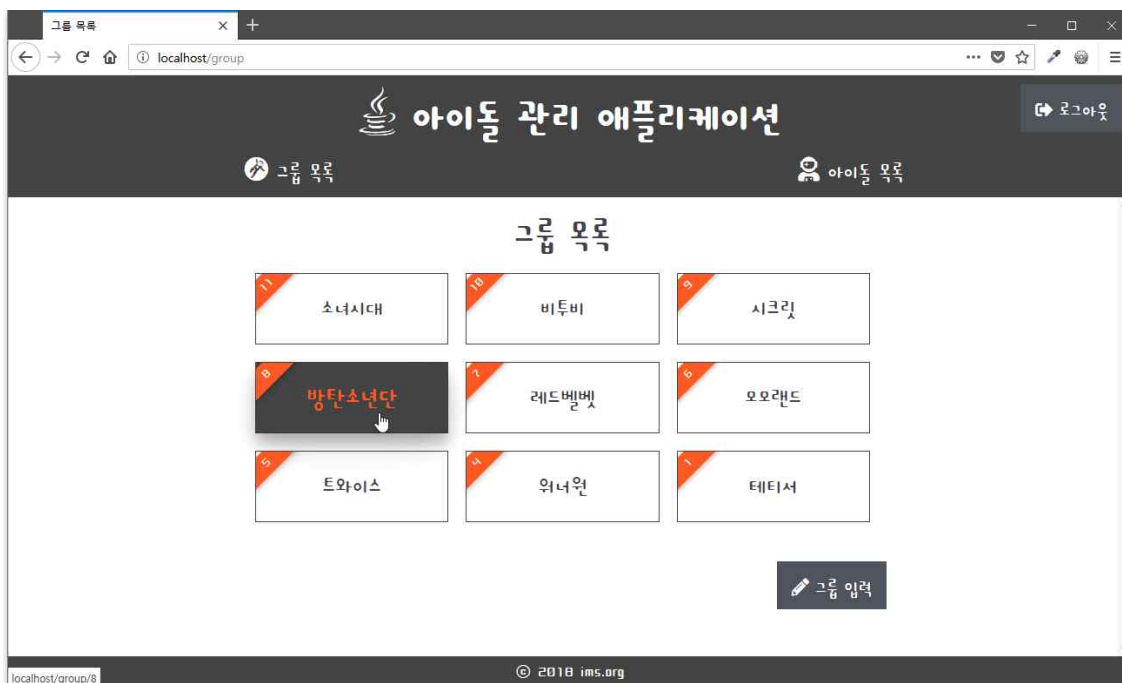


- 버전업된 fontawesome 적용(자바 아이콘등 추가)

2) /index 로그인 후



3) 그룹목록



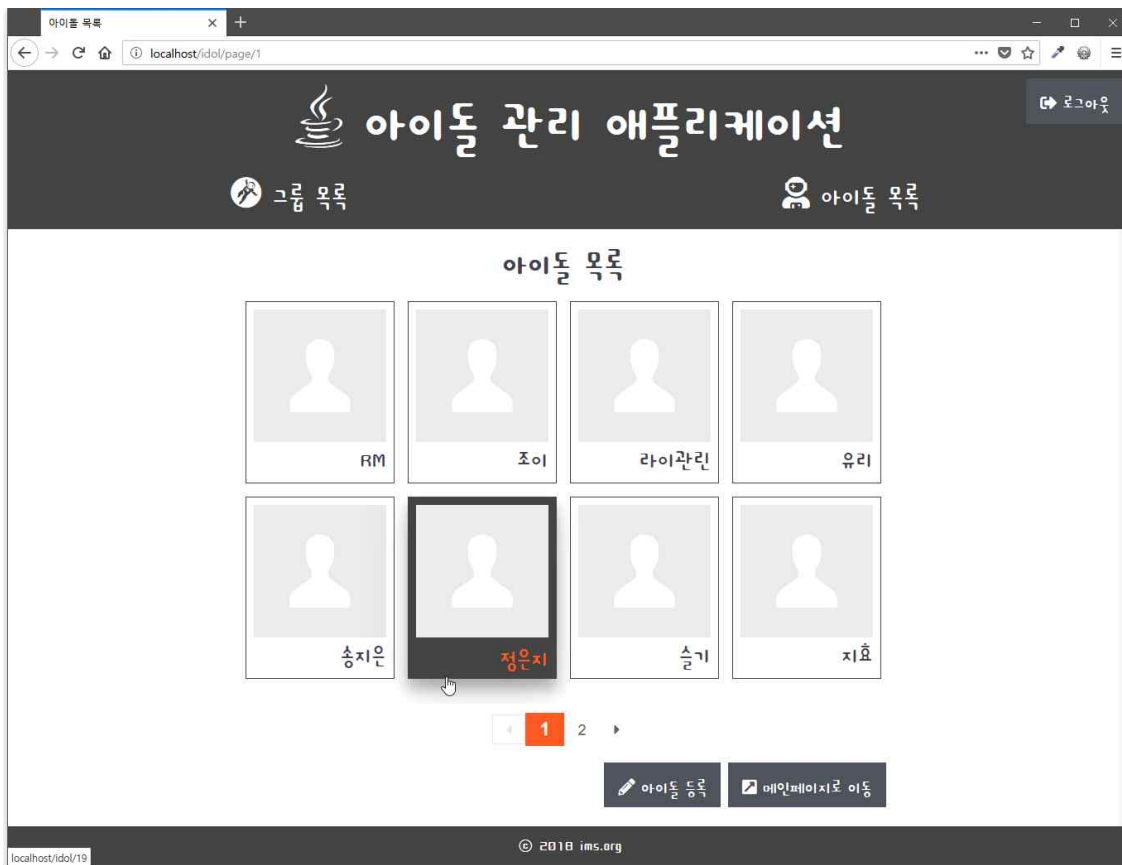
4) 그룹상세 페이지



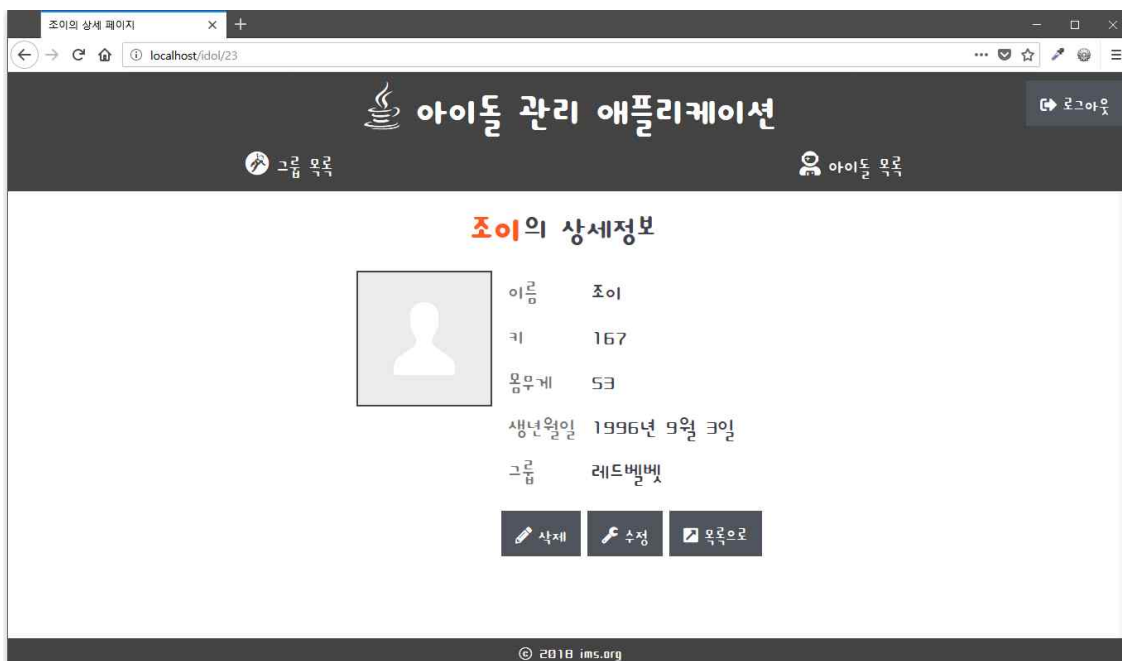
5) 그룹입력 페이지



6) 그룹목록 페이지



7) 아이돌 상세 페이지



8) 아이돌 수정 페이지

아이돌 수정 폼

localhost/idol/23/update

로그아웃

그룹 목록

아이돌 목록

아이돌 수정 페이지

이름:

생년월일:

키: cm

몸무게: kg

그룹:

사진:

© 2018 ims.org

9) 파일 업로드 적용

10) 인터셉터 적용