

삼성 청년 SW 아카데미

Spring Framework

<알림>

본 강의는 삼성 청년 SW아카데미의 콘텐츠로
보안서약서에 의거하여
강의 내용을 어떠한 사유로도 임의로 복사,
촬영, 녹음, 복제, 보관, 전송하거나
허가 받지 않은 저장매체를
이용한 보관, 제3자에게 누설, 공개,
또는 사용하는 등의 행위를 금합니다.

SpringFramework - REST API

목차

1. REST API
2. REST API 실습

REST API

삼성 청년 SW 아카데미

REST(Representational State Transfer) API

✓ OPEN API?? (Application Programming Interface)

- OPEN API는 프로그래밍에서 사용할 수 있는 개방되어 있는 상태의 Interface.
- naver, kakao등 포털 서비스 사이트나 통계청, 기상청, 우체국 등과 같은 관공서, 공공 데이터 포털 (<https://www.data.go.kr>)이 가지고 있는 데이터를 외부 응용 프로그램에서 사용할 수 있도록 OPEN API를 제공하고 있다.
- OPEN API와 함께 거론되는 기술이 REST이며, 대부분의 OPEN API는 REST방식으로 지원.

REST(Representational State Transfer) API

✓ REST(Representational State Transfer).

- 2000년도 로이 필딩(Roy Fielding)의 박사학위 논문에 최초로 소개.
- REST는 'Representational State Transfer'의 약어로 하나의 URI는 하나의 고유한 리소스(Resource)를 대표하도록 설계된다는 개념에 전송방식을 결합해서 원하는 작업을 지정한다.



- 웹의 장점을 최대한 활용할 수 있는 아키텍처(설계구조)로써 REST를 발표.
- HTTP URI를 통해 제어할 자원(Resource)를 명시하고, HTTP Method(GET, POST, PUT, DELETE)을 통해 해당 자원(Resource)를 제어하는 명령을 내리는 방식의 아키텍처.

REST(Representational State Transfer) API

✓ REST 구성.

- 자원 (Resource). - URI
- 행위 (Verb). - HTTP Method
- 표현 (Representations).

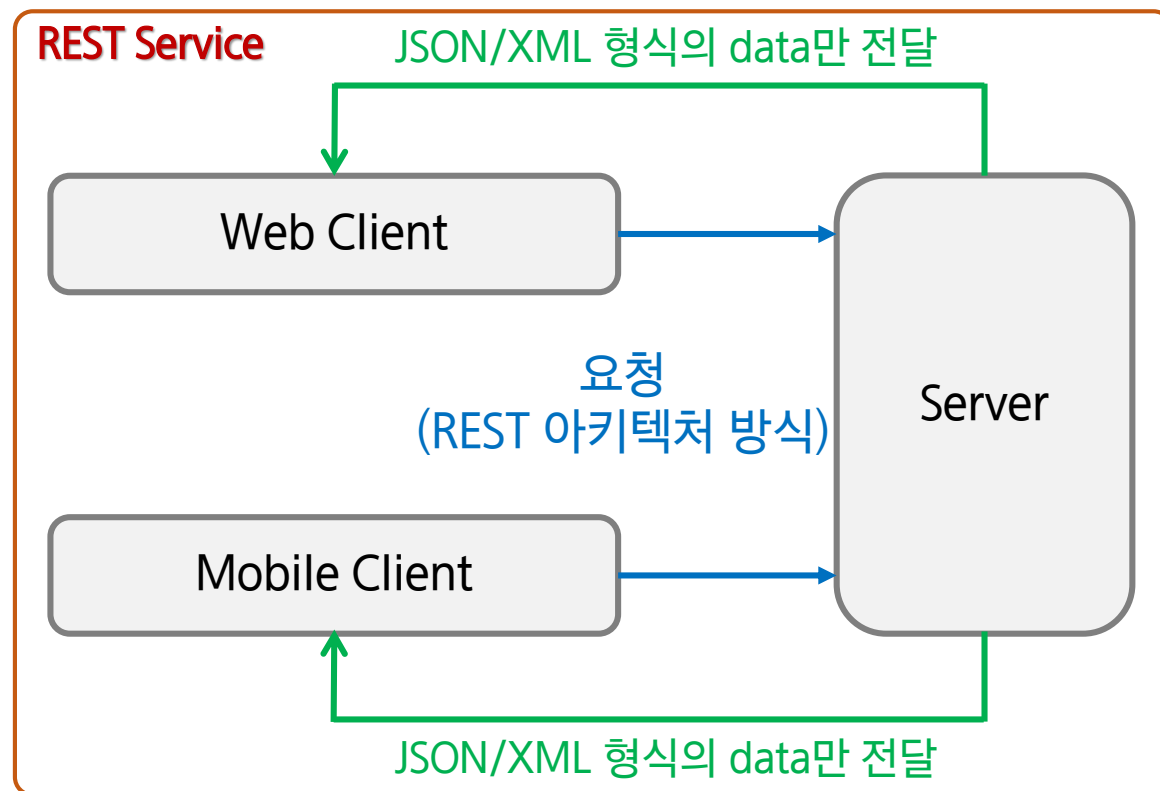
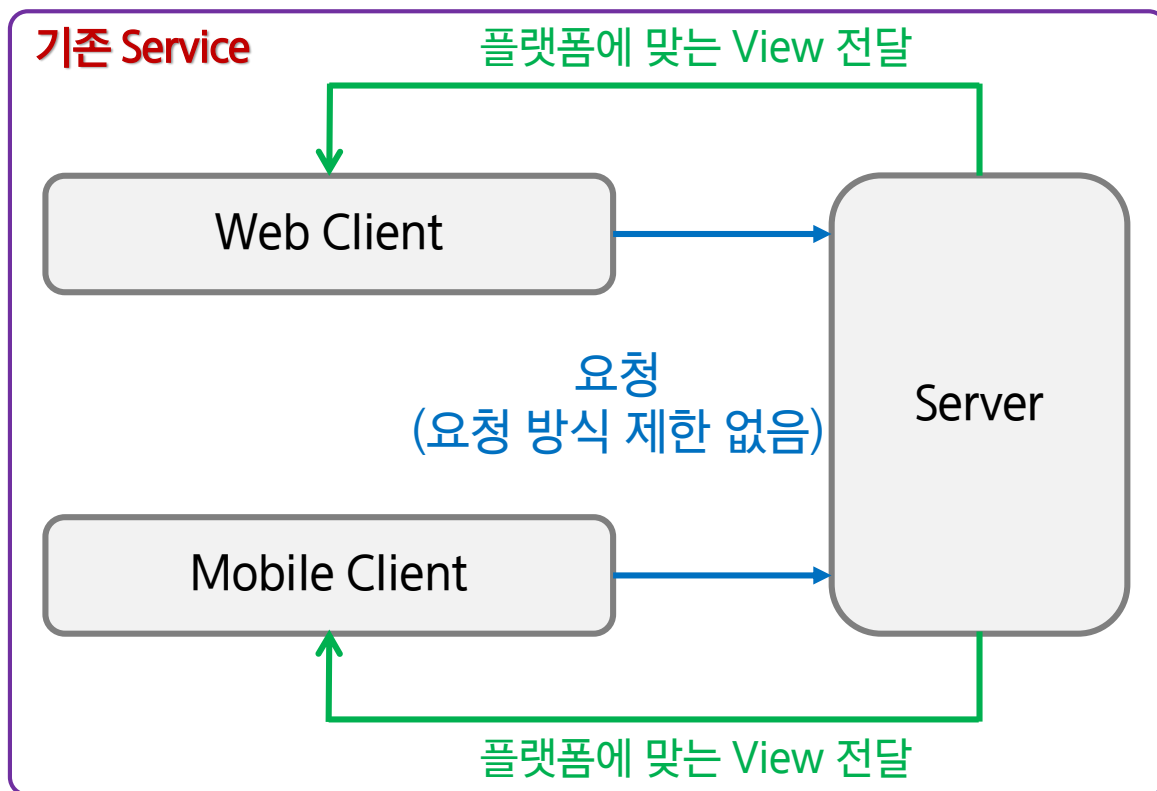
잘 표현된 HTTP URI로 리소스를 정의하고 HTTP method로 리소스에 대한 행위를 정의한다.

리소스는 JSON, XML과 같은 여러 가지 언어로 표현할 수 있다.

REST(Representational State Transfer) API

✓ 기존 Service와 REST Service.

- 기존 Service : 요청에 대한 처리를 한 후 가공된 data를 이용하여 특정 플랫폼에 적합한 형태의 View로 만들어서 반환.
- REST Service : data 처리만 한다 거나, 처리 후 반환될 data가 있다면 JSON이나 XML 형식으로 전달. View에 대해서는 신경 쓸 필요가 없다. >> 이러한 이유로 Open API에서 많이 사용.



REST(Representational State Transfer) API

✓ REST.

- 기존의 전송방식과는 달리 서버는 요청으로 받은 리소스에 대해 순수한 데이터를 전송한다.
- 기존의 GET/POST 외에 PUT, DELETE 방식을 사용하여 리소스에 대한 CRUD 처리를 할 수 있다.
- HTTP URI을 통해 제어할 자원(Resource)을 명시하고, HTTP METHOD(GET/POST/PUT/DELETE)를 통해 해당 자원(Resource)를 제어하는 명령을 내리는 방식의 Architecture이다.
- 가장 큰 단점은 딱 정해진 표준이 없어 '다들 이렇게 쓰더라~~' 정도의 암묵적인 표준만 정해져 있다.
 - 하이픈(-)은 사용 가능하지만 언더바(_)는 사용하지 않는다.
 - 특별한 경우를 제외하고 대문자 사용은 하지 않는다.(대소문자 구분을 하기 때문)
 - URI 마지막에 슬래시(/)를 사용하지 않는다.
 - 슬래시(/)로 계층 관계를 나타낸다.
 - 확장자가 포함된 파일 이름을 직접 포함시키지 않는다.
 - URI는 명사를 사용한다.

REST(Representational State Transfer) API

✓ REST.

- 기존의 웹 접근 방식과 REST API 방식의 차이점.

작업	기존 방식		REST 방식		비고
Create(Insert)	POST	/write.do?id=troment	POST	/blog/troment	글쓰기
Read(Select)	GET	/view.do?id=troment&articleno=25	GET	/blog/troment/25	글읽기
Update(Update)	POST	/modify.do?id=troment	PUT	/blog/troment	글수정
Delete>Delete)	GET	/delete.do?id=troment&articleno=25	DELETE	/blog/troment/25	글삭제

- 기존의 블로그등은 GET과 POST만으로 자원에 대한 CRUD를 처리하며, URI는 액션을 나타냈다.
- REST로 변경할 경우 4가지 method를 모두 사용하여 CRUD를 처리하며, URI는 제어하려는 자원을 나타낸다

REST(Representational State Transfer) API

✓ REST API 설정.

- Jackson library.
 - jackson-databind 라이브러리는 객체를 **JSON 포맷**의 문자열로 변환시켜서 브라우저로 전송한다.
 - <https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-databind>
 - jackson-dataformat-xml 라이브러리는 객체를 **xml**로 브라우저로 전송한다.
 - <https://mvnrepository.com/artifact/com.fasterxml.jackson.dataformat/jackson-dataformat-xml>
 - pom.xml에 library 추가.

```
<!-- https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-databind -->
<dependency>
  <groupId>com.fasterxml.jackson.core</groupId>
  <artifactId>jackson-databind</artifactId>
  <version>${jackson-databind-version}</version>
</dependency>
```

REST(Representational State Transfer) API

✓ REST 관련 Annotation.

Annotation	Description
@RestController	Controller 가 REST 방식을 처리하기 위한 것임을 명시.
@ResponseBody	JSP 같은 뷰로 전달되는 것이 아니라 데이터 자체를 전달.
@PathVariable	URL 경로에 있는 값을 파라미터로 추출.
@CrossOrigin	Ajax의 크로스 도메인 문제를 해결.
@RequestBody	JSON 데이터를 원하는 타입으로 바인딩.

이어서..

삼성 청년 SW 아카데미

REST API 실습

삼성 청년 SW 아카데미

REST(Representational State Transfer) API

✓ REST API 예 : 회원 정보.

- 로그인 후 회원 목록 클릭.



REST(Representational State Transfer) API

✓ REST API 예.

- list.jsp

```
$.ajax({  
  url: '${root}/admin/user',  
  type: 'GET',  
  contentType: 'application/json;charset=utf-8',  
  dataType: 'json',  
  success: function(users) {  
    makeList(users);  
  },  
  error: function(xhr, status, msg){  
    console.log("상태값 : " + status + " Http에러메시지 : "+msg);  
  }  
});
```

REST(Representational State Transfer) API

✓ REST API 예.

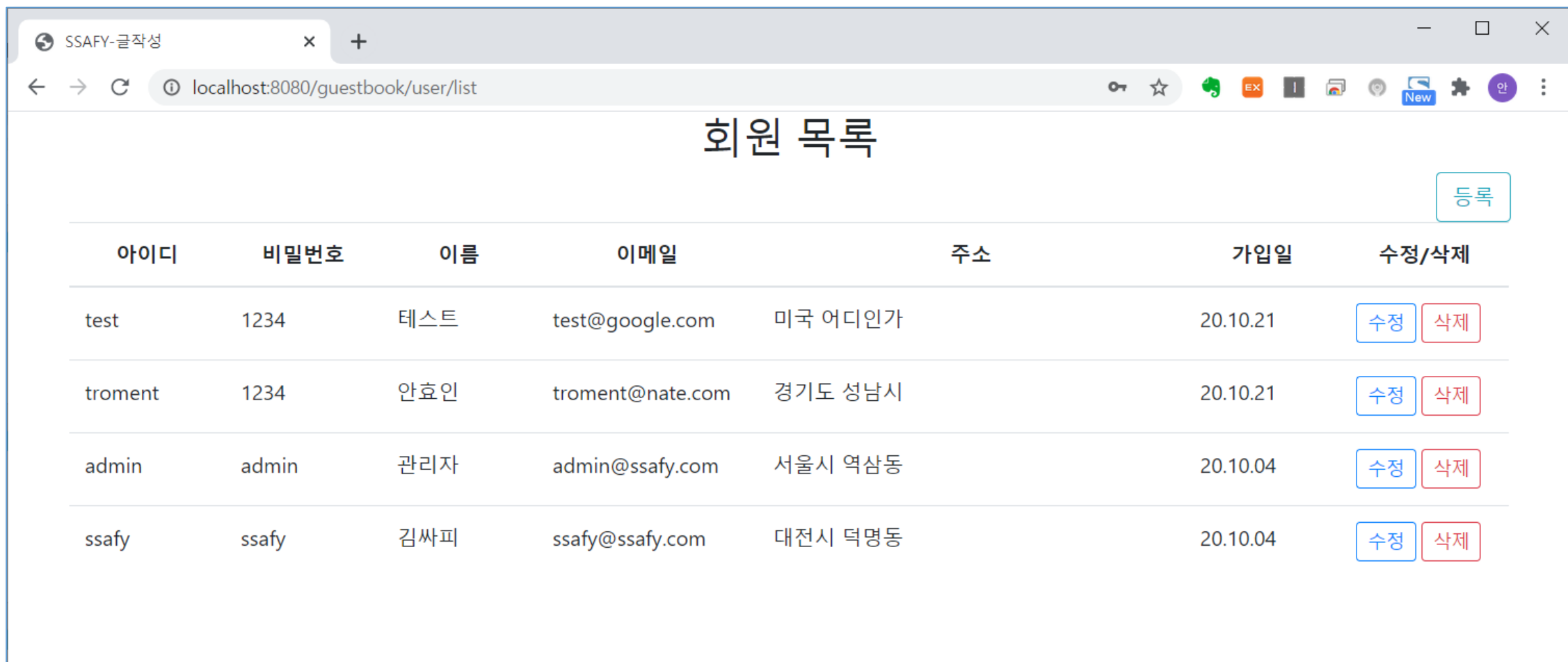
- AdminController.java

```
@RequestMapping(value = "/user", method = RequestMethod.GET,  
                headers = { "Content-type=application/json" })  
public List<MemberDto> userList() {  
    return userService.userList();  
}
```

REST(Representational State Transfer) API

✓ REST API 예.

- 회원 목록 결과.

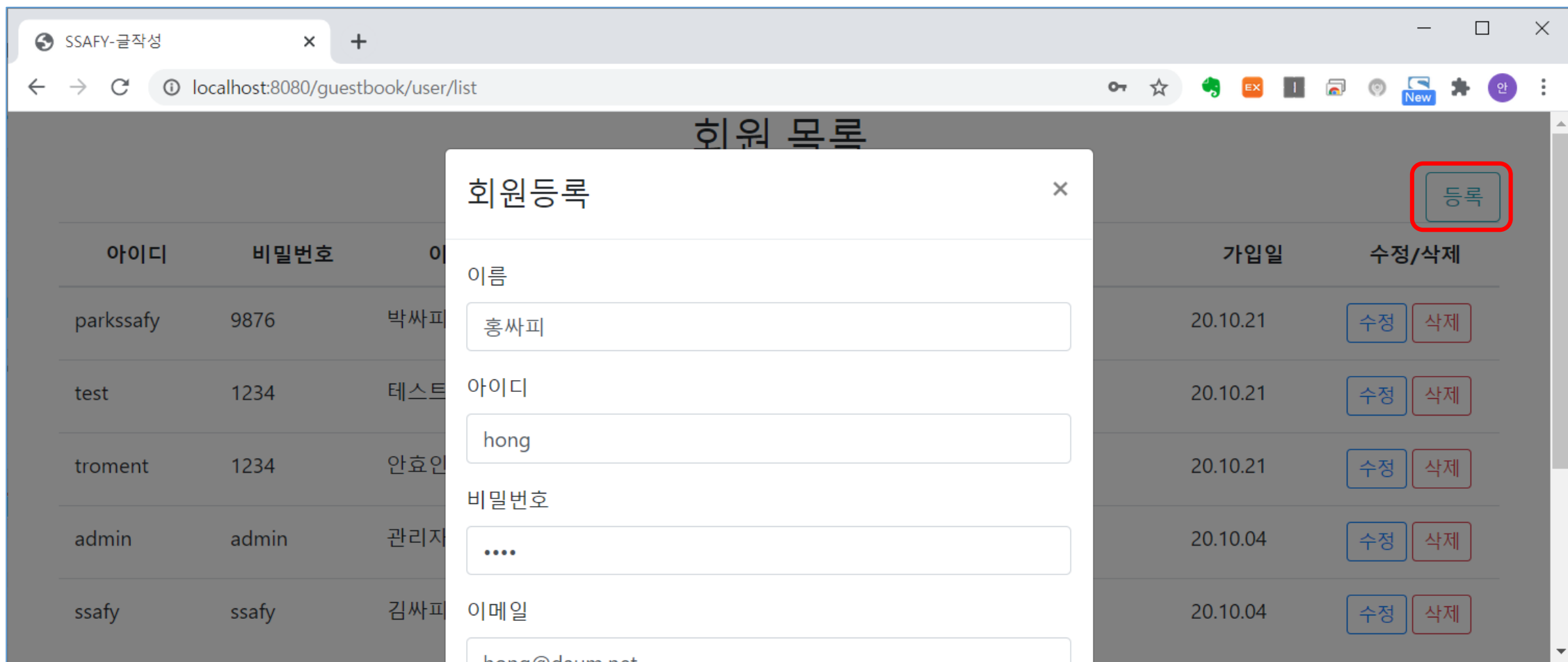


아이디	비밀번호	이름	이메일	주소	가입일	수정/삭제
test	1234	테스트	test@google.com	미국 어디인가	20.10.21	수정 삭제
troment	1234	안효인	troment@nate.com	경기도 성남시	20.10.21	수정 삭제
admin	admin	관리자	admin@ssafy.com	서울시 역삼동	20.10.04	수정 삭제
ssafy	ssafy	김싸피	ssafy@ssafy.com	대전시 덕명동	20.10.04	수정 삭제

REST(Representational State Transfer) API

✓ REST API 예.

- 회원 목록에서 등록 클릭.
- 정보입력 후 회원가입.



REST(Representational State Transfer) API

✓ REST API 예.

▪ list.jsp

```
$("#registerBtn").click(function() {  
    let registerinfo = JSON.stringify({  
        "username" : $("#username").val(), "userid" : $("#userid").val(), "userpwd" : $("#userpwd").val(),  
        "email" : $("#email").val(), "address" : $("#address").val()  
    });  
    $.ajax({  
        url: '${root}/admin/user',  
        type: 'POST',  
        contentType: 'application/json;charset=utf-8',  
        dataType: 'json',  
        data: registerinfo,  
        success: function(users) {  
            $("#username").val(''); $("#userid").val(''); $("#userpwd").val('');  
            $("#email").val(''); $("#address").val('');  
            $("#userRegModal").modal("hide");  
            makeList(users);  
        },  
        error: function(xhr, status, msg){  
            console.log("상태값 : " + status + " Http에러메시지 : "+msg);  
        }  
    });  
});
```

REST(Representational State Transfer) API

✓ REST API 예.

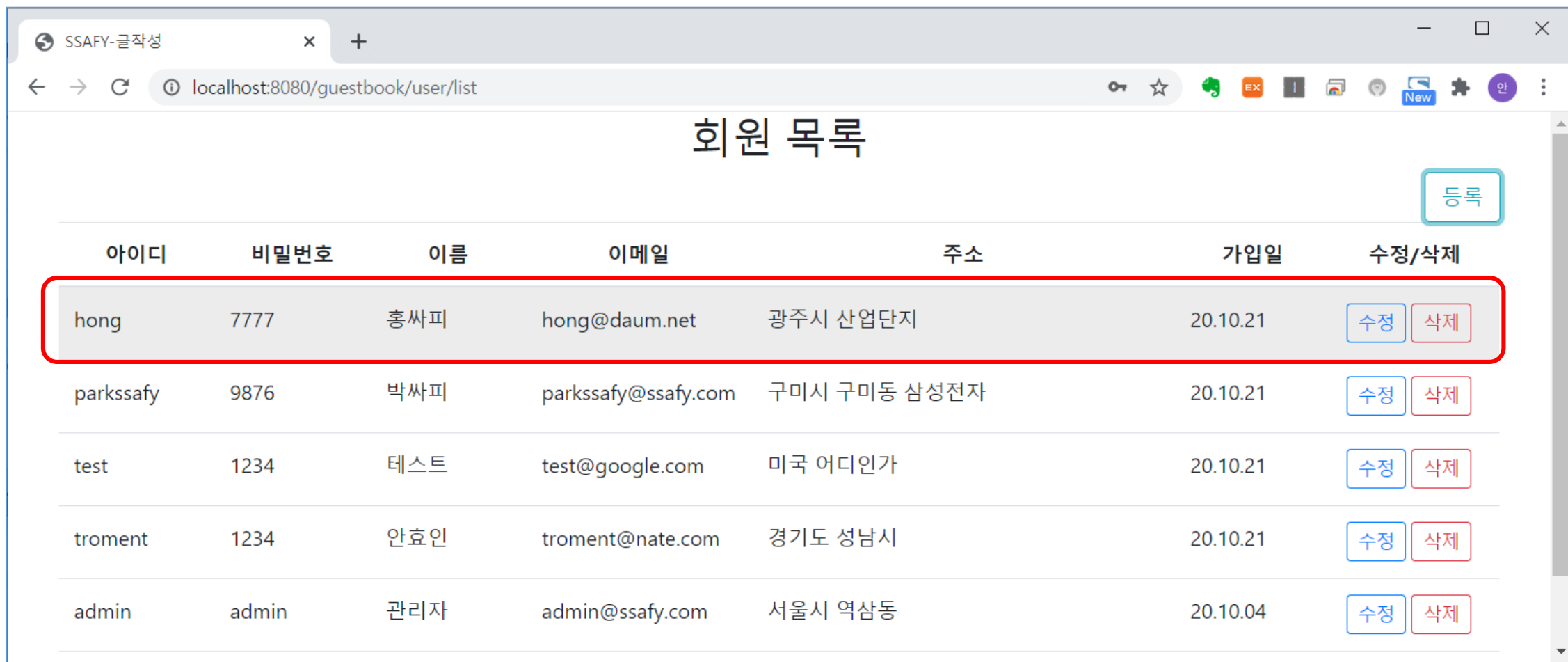
- AdminController.java

```
@RequestMapping(value = "/user", method = RequestMethod.POST,  
                  headers = { "Content-type=application/json" })  
public List<MemberDto> userRegister(@RequestBody MemberDto memberDto) {  
    userService.userRegister(memberDto);  
    return userService.userList();  
}
```

REST(Representational State Transfer) API

✓ REST API 예.

- 회원 등록 결과.



아이디	비밀번호	이름	이메일	주소	가입일	수정/삭제
hong	7777	홍싸피	hong@daum.net	광주시 산업단지	20.10.21	수정 삭제
parkssafy	9876	박싸피	parkssafy@ssafy.com	구미시 구미동 삼성전자	20.10.21	수정 삭제
test	1234	테스트	test@google.com	미국 어디인가	20.10.21	수정 삭제
troment	1234	안효인	troment@nate.com	경기도 성남시	20.10.21	수정 삭제
admin	admin	관리자	admin@ssafy.com	서울시 역삼동	20.10.04	수정 삭제

REST(Representational State Transfer) API

✓ REST API 예.

- 검색하고자 하는 회원을 더블클릭.

회원 목록

등록

아이디	비밀번호	이름	이메일	주소	가입일	수정/삭제
parkssafy	9876	박싸피	parkssafy@ssafy.com	구미시 구미동 삼성전자	20.10.21	수정 삭제
test	1234	테스트	test@google.com	미국 어디인가	20.10.21	수정 삭제
troment	1234	안효인	troment@ssafy.com	서울시 역삼동	20.10.21	수정 삭제
admin	admin	관리자	admin@ssafy.com	서울시 역삼동	20.10.04	수정 삭제
ssafy	ssafy	김싸피	ssafy@ssafy.com	대전시 덕명동	20.10.04	수정 삭제

REST(Representational State Transfer) API

✓ REST API 예.

▪ list.jsp

```
$(document).on("dblclick", "tr.view", function() {  
    let vid = $(this).attr("data-id");  
    $.ajax({  
        url: '${root}/admin/user/' + vid,  
        type: 'GET',  
        contentType: 'application/json;charset=utf-8',  
        success: function(user) {  
            $("#vid").text(user.userid);  
            $("#vname").text(user.username);  
            $("#vemail").text(user.email);  
            $("#vaddress").text(user.address);  
            $("#vjoindate").text(user.joindate);  
            $("#userViewModal").modal();  
        },  
        error: function(xhr, status, msg){  
            console.log("상태값 : " + status + " Http에러메시지 : "+msg);  
        }  
    });  
});
```

REST(Representational State Transfer) API

✓ REST API 예.

- AdminController.java

```
@RequestMapping(value = "/user/{userid}", method = RequestMethod.GET,  
                  headers = { "Content-type=application/json" })  
public MemberDto userInfo(@PathVariable("userid") String userid) {  
    return userService.userInfo(userid);  
}
```

REST(Representational State Transfer) API

✓ REST API 예.

- 회원정보 보기 결과.

The screenshot shows a web browser window with the address bar displaying 'localhost:8080/guestbook/user/list'. The page title is '회원 목록' (Member List). A modal window titled '회원정보' (Member Information) is open, showing the details for the user 'troment'.

ID	이메일	회원명	가입일
troment	troment@nate.com	안효인	20.10.21
주소: 경기도 성남시			

The background page shows a table of members with columns: 아이디, 비밀번호, 이름, 이메일, 주소, 가입일, and 수정/삭제. The table contains several rows, including 'parkssafy', 'test', 'troment', 'admin', and 'ssafy'.

REST(Representational State Transfer) API

✓ REST API 예.

- 정보 수정할 회원의 수정 버튼 클릭.
- 수정 화면으로 전환 후 회원 정보 변경..

아이디	비밀번호	이름	이메일	주소	가입일	수정/삭제
hong	9876	홍싸피	hong@kakao.com	강주시 산업단지 삼성전자	20.10.21	수정 취소
parkssafy	9876	박싸피	parkssafy@ssafy.com	구미시 구미동 삼성전자	20.10.21	수정 삭제
test	1234	테스트	test@google.com	미국 어디인가	20.10.21	수정 삭제
troment	1234	안효인	troment@nate.com	경기도 성남시	20.10.21	수정 삭제
admin	admin	관리자	admin@ssafy.com	서울시 역삼동	20.10.04	수정 삭제

REST(Representational State Transfer) API

✓ REST API 예.

- list.jsp

```
$(document).on("click", ".modiBtn", function() {  
    let mid = $(this).parents("tr").attr("data-id");  
    $("#view_" + mid).css("display", "none");  
    $("#mview_" + mid).css("display", "");  
});
```

REST(Representational State Transfer) API

✓ REST API 예.

▪ list.jsp

```
$(document).on("click", ".modifyBtn", function() {  
    let mid = $(this).parents("tr").attr("data-id");  
    let modifyinfo = JSON.stringify({  
        "userid" : mid, "userpwd" : $("#userpwd" + mid).val(),  
        "email" : $("#email" + mid).val(), "address" : $("#address" + mid).val()  
    });  
    $.ajax({  
        url: '${root}/admin/user',  
        type: 'PUT',  
        contentType: 'application/json;charset=utf-8',  
        dataType: 'json',  
        data: modifyinfo,  
        success: function(users) {  
            makeList(users);  
        }  
    });  
});
```

REST(Representational State Transfer) API

✓ REST API 예.

- AdminController.java

```
@RequestMapping(value = "/user", method = RequestMethod.PUT,  
                  headers = { "Content-type=application/json" })  
public List<MemberDto> userModify(@RequestBody MemberDto memberDto) {  
    userService.userModify(memberDto);  
    return userService.userList();  
}
```

REST(Representational State Transfer) API

✓ REST API 예.

- 회원 정보 수정 결과.

아이디	비밀번호	이름	이메일	주소	가입일	수정/삭제
hong	9876	홍싸피	hong@kakao.com	광주시 산업단지 삼성전자	20.10.21	수정 삭제
parkssafy	9876	박싸피	parkssafy@ssafy.com	구미시 구미동 삼성전자	20.10.21	수정 삭제
test	1234	테스트	test@google.com	미국 어디인가	20.10.21	수정 삭제
troment	1234	안효인	troment@nate.com	경기도 성남시	20.10.21	수정 삭제
admin	admin	관리자	admin@ssafy.com	서울시 역삼동	20.10.04	수정 삭제

REST(Representational State Transfer) API

✓ REST API 예.

- 삭제할 회원의 삭제 버튼 클릭.

The screenshot shows a web browser window with the address bar displaying `localhost:8080/guestbook/user/list`. The page contains a table of users and a confirmation dialog box.

Confirmation Dialog:

localhost:8080 내용:
정말 삭제?

Buttons:

User List Table:

아이디	비밀번호	이름	가입일	수정/삭제
hong	9876	홍싸피	20.10.21	<input type="button" value="수정"/> <input type="button" value="삭제"/>
parkssafy	9876	박싸피	20.10.21	<input type="button" value="수정"/> <input type="button" value="삭제"/>
test	1234	테스트	20.10.21	<input type="button" value="수정"/> <input type="button" value="삭제"/>
troment	1234	안효인	20.10.21	<input type="button" value="수정"/> <input type="button" value="삭제"/>
admin	admin	관리자	20.10.04	<input type="button" value="수정"/> <input type="button" value="삭제"/>

REST(Representational State Transfer) API

✓ REST API 예.

▪ list.jsp

```
$(document).on("click", ".delBtn", function() {  
    if(confirm("정말 삭제?")) {  
        let delid = $(this).parents("tr").attr("data-id");  
        $.ajax({  
            url: '${root}/admin/user/' + delid,  
            type: 'DELETE',  
            contentType: 'application/json; charset=utf-8',  
            dataType: 'json',  
            success: function(users) {  
                makeList(users);  
            },  
            error: function(xhr, status, msg){  
                console.log("상태값 : " + status + " Http에러메시지 : "+msg);  
            }  
        });  
    }  
});  
});
```

REST(Representational State Transfer) API

✓ REST API 예.

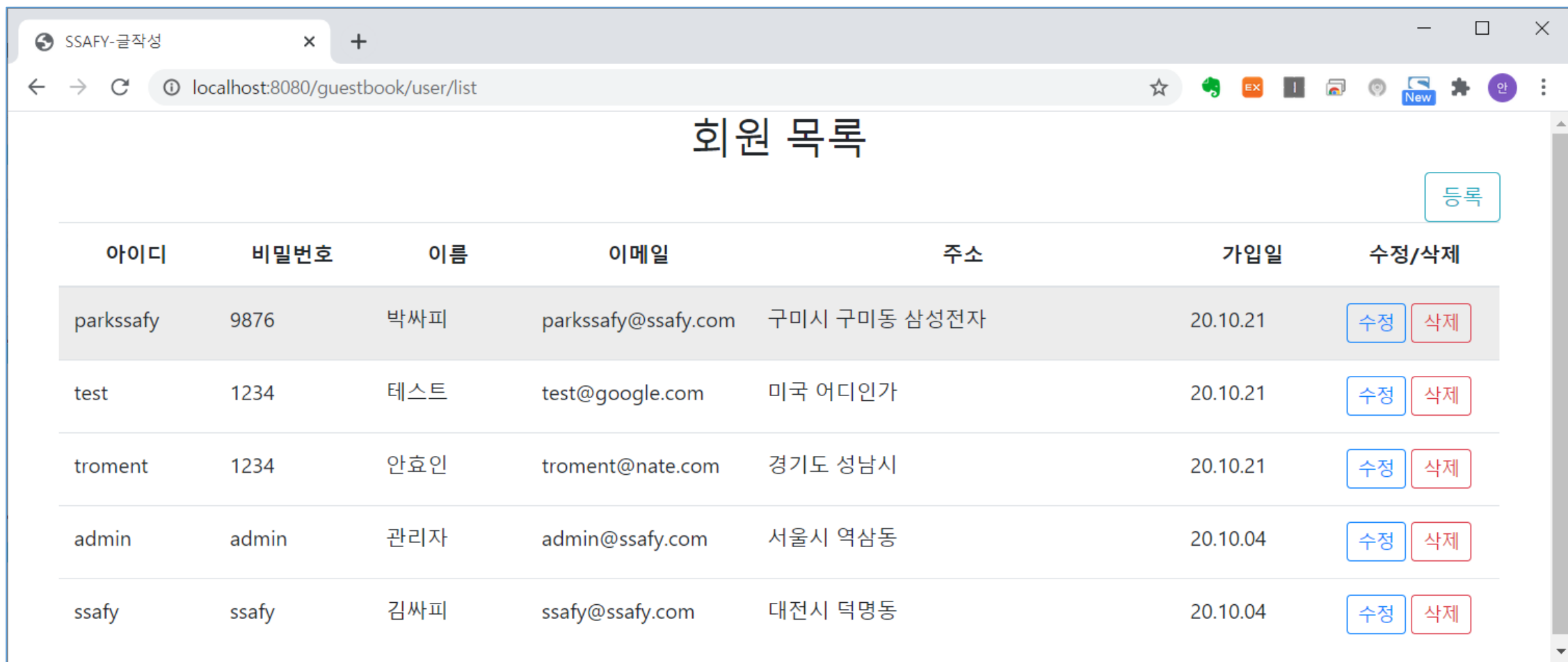
- AdminController.java

```
@RequestMapping(value = "/user/{userid}", method = RequestMethod.DELETE,  
                  headers = { "Content-type=application/json" })  
public List<MemberDto> userDelete(@PathVariable("userid") String userid) {  
    userService.userDelete(userid);  
    return userService.userList();  
}
```

REST(Representational State Transfer) API

✓ REST API 예.

- 회원 삭제 결과.



아이디	비밀번호	이름	이메일	주소	가입일	수정/삭제
parkssafy	9876	박싸피	parkssafy@ssafy.com	구미시 구미동 삼성전자	20.10.21	수정 삭제
test	1234	테스트	test@google.com	미국 어디인가	20.10.21	수정 삭제
troment	1234	안효인	troment@nate.com	경기도 성남시	20.10.21	수정 삭제
admin	admin	관리자	admin@ssafy.com	서울시 역삼동	20.10.04	수정 삭제
ssafy	ssafy	김싸피	ssafy@ssafy.com	대전시 덕명동	20.10.04	수정 삭제

REST(Representational State Transfer) API

✓ REST API 예.

- list.jsp : 회원 목록 테이블 만들기.

```
function makeList(users) {
    $("#userlist").empty();
    $(users).each(function(index, user) {
        let str = "<tr id=\"view_\" + user.userid + \"\" class=\"view\" data-id=\"\" + user.userid + \"\">"
            + "<td>\" + user.userid + "</td>"
            + "<td>\" + user.userpwd + "</td>"
            + "<td>\" + user.username + "</td>"
            + "<td>\" + user.email + "</td>"
            + "<td>\" + user.address + "</td>"
            + "<td>\" + user.joindate + "</td>"
            + "<td><button type=\"button\" class=\"modiBtn btn btn-outline-primary btn-sm\">수정</button> "
            + "<button type=\"button\" class=\"delBtn btn btn-outline-danger btn-sm\">삭제</button></td>"
            + "</tr>"
            + "<tr id=\"mview_\" + user.userid + \"\" data-id=\"\" + user.userid + \"\" style=\"display: none;\">"
            + "<td>\" + user.userid + "</td>"
            + "<td><input type=\"text\" name=\"userpwd\" id=\"userpwd\" + user.userid + \"\" value=\"\" + user.userpwd + \"\"></td>"
            + "<td>\" + user.username + "</td>"
            + "<td><input type=\"text\" name=\"email\" id=\"email\" + user.userid + \"\" value=\"\" + user.email + \"\"></td>"
            + "<td><input type=\"text\" name=\"address\" id=\"address\" + user.userid + \"\" value=\"\" + user.address + \"\"></td>"
            + "<td>\" + user.joindate + "</td>"
            + "<td><button type=\"button\" class=\"modifyBtn btn btn-outline-primary btn-sm\">수정</button> "
            + "<button type=\"button\" class=\"cancelBtn btn btn-outline-danger btn-sm\">취소</button></td>"
            + "</tr>";
        $("#userlist").append(str);
    }); //each
}
```

내일 방송에서 만나요!

삼성 청년 SW 아카데미