

회원 정보 페이지

1. 로그인/ 로그아웃

```
@PostMapping("login")
public ResponseEntity<String> login(@RequestBody User user, HttpSession session) throws Exception {
    User userInfo = uService.login(user);
    if (userInfo != null) {
        session.setAttribute("userInfo", userInfo);
        return ResponseEntity.ok("success");
    } else {
        return ResponseEntity.status(HttpStatus.UNAUTHORIZED).body("failure");
    }
}
```

METHOD: POST SCHEME: // HOST: localhost PORT: 80 PATH: /user/login QUERY: length: 30 bytes

HEADERS: Content-Type: application/json

BODY: { "id": "ssafy", "pw": 1234 }

Response: 200 Cache Deleted - Elapsed Time: This

HEADERS: Content-Type: text/plain; charset=UTF-8 Content-Length: 7 bytes

BODY: success

```
@GetMapping("logout")
public ResponseEntity<String> logout(HttpSession session) throws Exception{
    session.invalidate();
    return ResponseEntity.ok("success");
}
```

METHOD: GET SCHEME: // HOST: localhost PORT: 80 PATH: /user/logout QUERY: length: 31 bytes

HEADERS: XHR does not allow payloads for GET request.

Response: 200 Cache: [

HEADERS: Content-Type: text/plain; charset=UTF-8 Content-Length: 7 bytes Date: Tue, 02 May 2023 05:53:37 GMT Keep-Alive: timeout=60 Connection: keep-alive

BODY: success

2. 회원 정보 가입

```
@PostMapping("signup")
public ResponseEntity<String> signup(@RequestBody User user) {
    try {
        uService.signupUser(user);
        return new ResponseEntity<>("success", HttpStatus.OK);
    } catch (Exception e) {
        return new ResponseEntity<>("error", HttpStatus.INTERNAL_SERVER_ERROR);
    }
}
```

The screenshot shows a REST client interface with the following details:

- METHOD:** POST
- URL:** http://localhost:80/user/signup
- HEADERS:** Content-Type: application/json
- BODY:** A JSON object: {"id": "ssafy0", "pw": "1234", "name": "싸피0", "email": "ssafy0@ssafy.com"}
- Response:** 200 OK
- Response Headers:** Content-Type: text/plain; charset=utf-8, Content-Length: 7 bytes
- Response Body:** success

3. 회원 정보 조회

```
@GetMapping("getUser/{userid}")
public ResponseEntity<?> userInfo(@PathVariable("userid") String userId) throws Exception {
    User userInfo = uService.getUser(userId);
    if(userInfo != null) {
        return new ResponseEntity<User>(userInfo, HttpStatus.OK);
    } else {
        return new ResponseEntity<>(HttpStatus.NOT_FOUND);
    }
}
```

The screenshot shows a REST client interface with the following details:

- METHOD:** GET
- URL:** http://localhost:80/user/getUser/ssafy2
- HEADERS:** Content-Type: application/json, Transfer-Encoding: chunked, Date: Tue, 02 May 2023 06:40:50 GMT, Keep-Alive: timeout=60, Connection: keep-alive
- BODY:** A JSON object: {"id": "ssafy2", "pw": "12345", "name": "싸피2", "email": "ssafy@ssafy.com"}
- Response:** 200 OK

4. 회원 정보 수정

```
@PostMapping("update")
public ResponseEntity<String> updateUser(@RequestBody User user) {
    try {
        userService.editUser(user);
        return new ResponseEntity<>("회원 정보가 수정되었습니다.", HttpStatus.OK);
    } catch (Exception e) {
        e.printStackTrace();
        return new ResponseEntity<>("회원 정보 수정에 실패하였습니다.", HttpStatus.INTERNAL_SERVER_ERROR);
    }
}
```

METHOD: PUT SCHEME: // HOST: localhost PORT: 80 PATH: /user/update QUERY: []

HEADERS: Content-Type: application/json

BODY: { "id": "ssafy2", "pw": 12345, "name": "사사2", "email": "ssafy@ssafy.com" }

Response: 200

HEADERS: Content-Type: text/plain; charset=UTF-8 Content-Length: 39 bytes

BODY: 회원 정보가 수정되었습니다.

5. 회원 정보 삭제

```
@DeleteMapping("delete")
public ResponseEntity<Void> delete(HttpSession session) throws Exception {
    User userInfo = (User) session.getAttribute("userInfo");
    userService.deleteUser(userInfo);
    session.invalidate();
    return ResponseEntity.noContent().build();
}
```

METHOD: DELETE SCHEME: // HOST: localhost PORT: 80 PATH: /user/delete QUERY: []

HEADERS: XHR does not allow payloads for DELETE request.

Response: 204

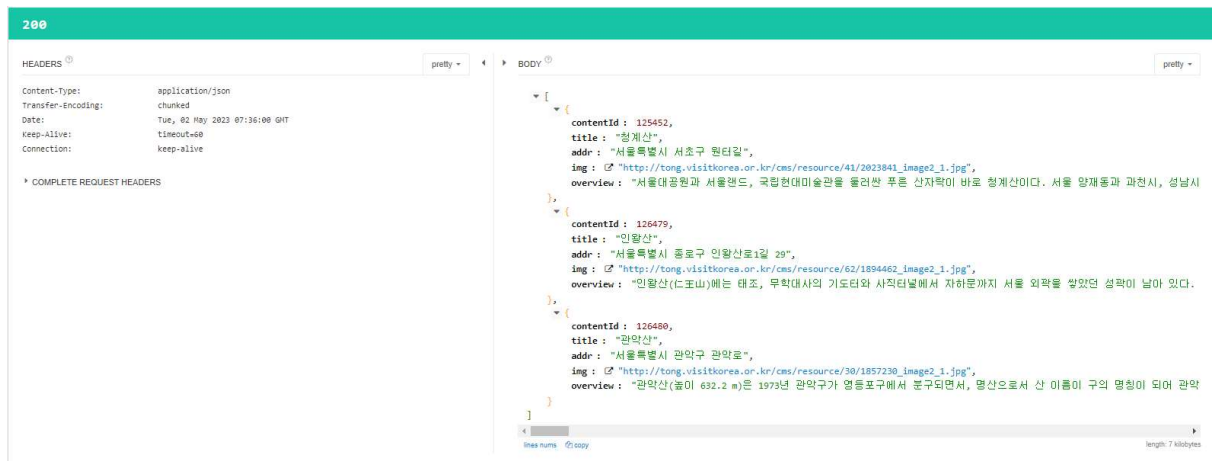
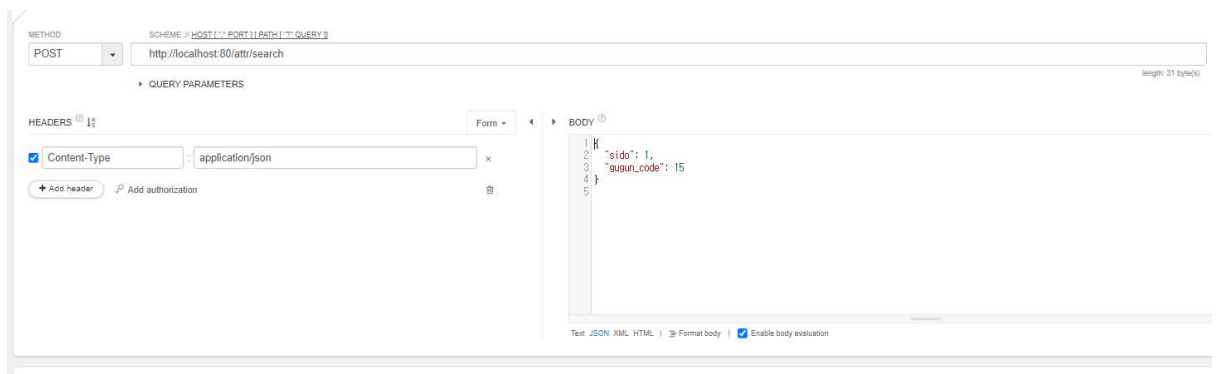
HEADERS: Date: Tue, 02 May 2023 06:00:53 GMT Keep-Alive: timeout=60 Connection: keep-alive Set-Cookie: JSESSIONID=33 B, session, HttpOnly

BODY: No Content

관광지 정보 관리

1. 지역별 관광지 검색

```
@PostMapping("/search")
public ResponseEntity<List<Attraction>> search(@RequestBody Search search) {
    try {
        List<Attraction> list = attrService.search(search);
        logger.info(list.toString());
        return ResponseEntity.ok().body(list);
    } catch (Exception e) {
        e.printStackTrace();
        return ResponseEntity.status(HttpStatus.INTERNAL_SERVER_ERROR).build();
    }
}
```



2. Detail 검색

```
@GetMapping("/detail/{contentId}")
public ResponseEntity<Attraction> detail(@PathVariable String contentId) {
    try {
        Attraction attraction = attrService.detail(contentId);
        return ResponseEntity.ok().body(attraction);
    } catch (Exception e) {
        e.printStackTrace();
        return ResponseEntity.status(HttpStatus.INTERNAL_SERVER_ERROR).build();
    }
}
```

METHOD: GET SCHEME // HOST [":" PORT] [PATH ["?" QUERY]] length: 35 byte(s)

Send

QUERY PARAMETERS

HEADERS

Form

BODY

XHR does not allow payloads for GET request.

Response

Cache Detected - Elapsed Time: 8ms

200

HEADERS

pretty

BODY

```
{
  "contentId": 125266,
  "title": "국립 청대산자연휴양림",
  "addr": "강원도 횡성군 둔내면 청대산로 610",
  "img": "http://tong.visitkorea.or.kr/cms/resource/21/2657021_image2_1.jpg",
  "overview": "해발 1,200m의 청대산을 주봉으로 하여 인공림과 천연림이 잘 조화된 웅장한 산림을 바탕으로 한 국유림 경영
```

3. 콘텐츠별 관광지 검색

METHOD: POST SCHEME // HOST [":" PORT] [PATH ["?" QUERY]] length: 31 byte(s)

Send

QUERY PARAMETERS

HEADERS

Form

BODY

Content-Type: application/json

```
[
  {
    "sido": 1,
    "gugun": 15,
    "type": 12
  },
  {
    "contentId": 125452,
    "title": "청계산",
    "addr": "서울특별시 서초구 원터길",
    "img": "http://tong.visitkorea.or.kr/cms/resource/41/2023841_image2_1.jpg",
    "overview": "서울대공원과 서울랜드, 국립현대미술관을 둘러싼 푸른 산자락이 바로 청계산이다. 서울 양재동과 과천시, 강남시"
  },
  {
    "contentId": 126530,
    "title": "서울 현릉(대릉,왕경왕후)과 인릉(순조,순원왕후) [유네스코 세계문화유산]",
    "addr": "서울특별시 서초구 현인로길 36-10",
    "img": "http://tong.visitkorea.or.kr/cms/resource/42/1985242_image2_1.jpg",
    "overview": "조선 제3대 태종과 그의 비 왕경왕후의 묘가 나란히 모셔져 있는 현릉,그리고 제23대 순조와 그의 비 순원왕후가"
  },
  {
    "contentId": 250443,
    "title": "왕금문천",
    "addr": "서울특별시 서초구 서원대로 143",
    "img": "http://tong.visitkorea.or.kr/cms/resource/06/2024206_image2_1.jpg",
    "overview": "왕금문천은 서울 강남의 한복판에서 온천수를 즐길 수 있는 곳이다. 최고의 수질은 물론 절경과 한강, 사우나"
  }
]
```

Response

Cache Detected - Elapsed Time: 5ms

200

HEADERS

pretty

BODY

COMPLETE REQUEST HEADERS