



APICloud

삼성청년SW아카데미 7기 대전 2반 5팀

자율 프로젝트 (22/10/11 ~ 22/11/21)

시연 시나리오

담당 컨설턴트: 정용기, 서성수

신영제(팀장), 김우원, 김은경, 김은혜, 정세미, 조유진

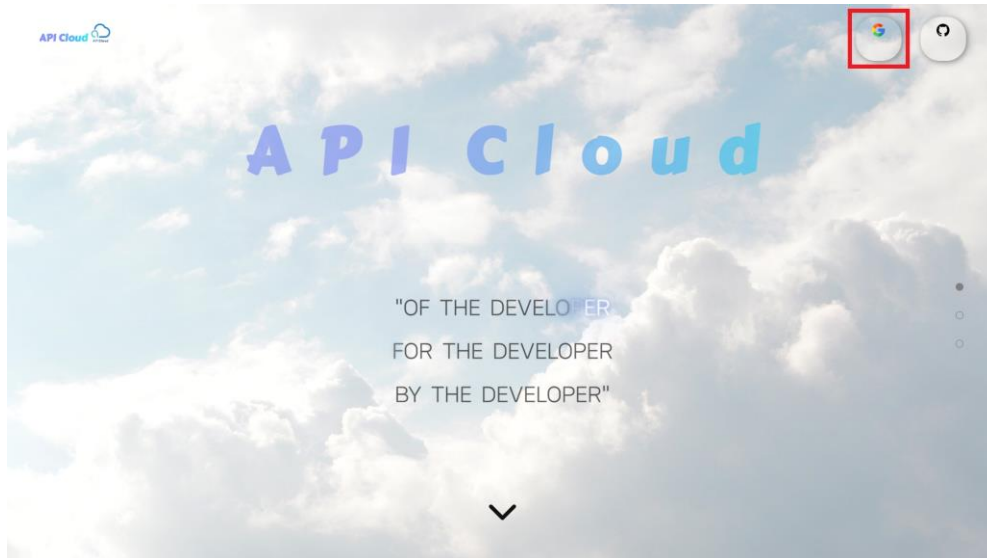
목차

| | |
|-------------------------|----|
| 1. 웰컴 페이지 | 3 |
| 2. 메인 페이지 | 3 |
| 3. API Doc 작성 페이지 | 5 |
| 4. 프로젝트 추출..... | 7 |
| 5. 프로젝트 동기화..... | 7 |
| 6. API Doc 조회 페이지 | 9 |
| 7. API Test 페이지 | 10 |

1. 웰컴 페이지

A. 소셜로그인

- i. Google 소셜 로그인 버튼을 눌러 로그인



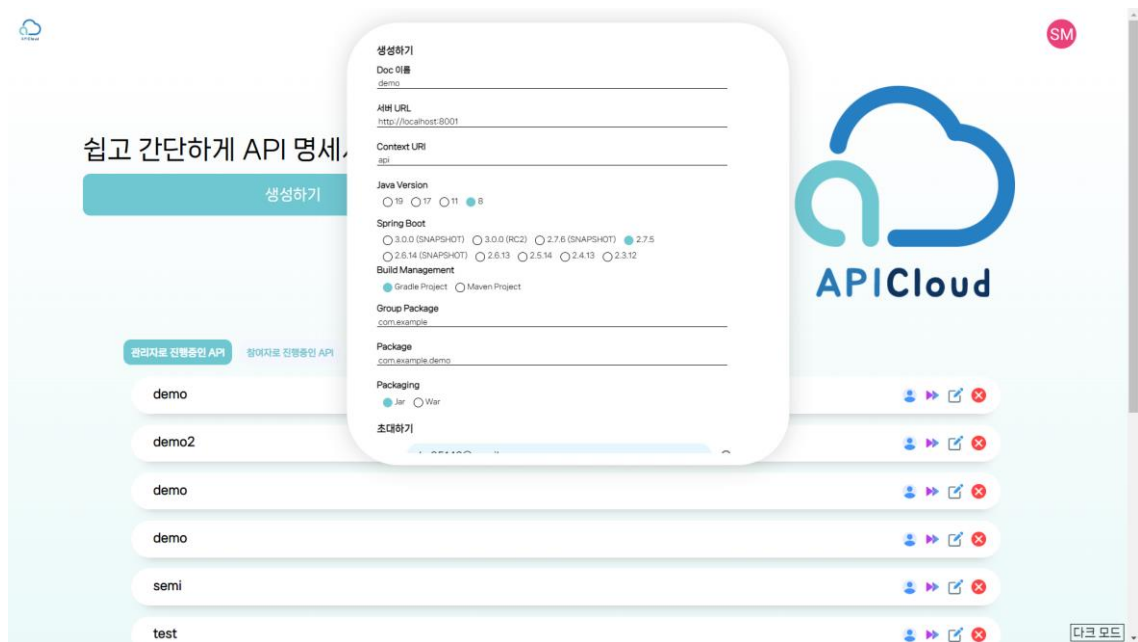
- ii. 로그인 후 자동으로 메인페이지로 이동

2. 메인 페이지

A. API Doc 생성



i. 서버 정보 입력



ii. 유저 초대

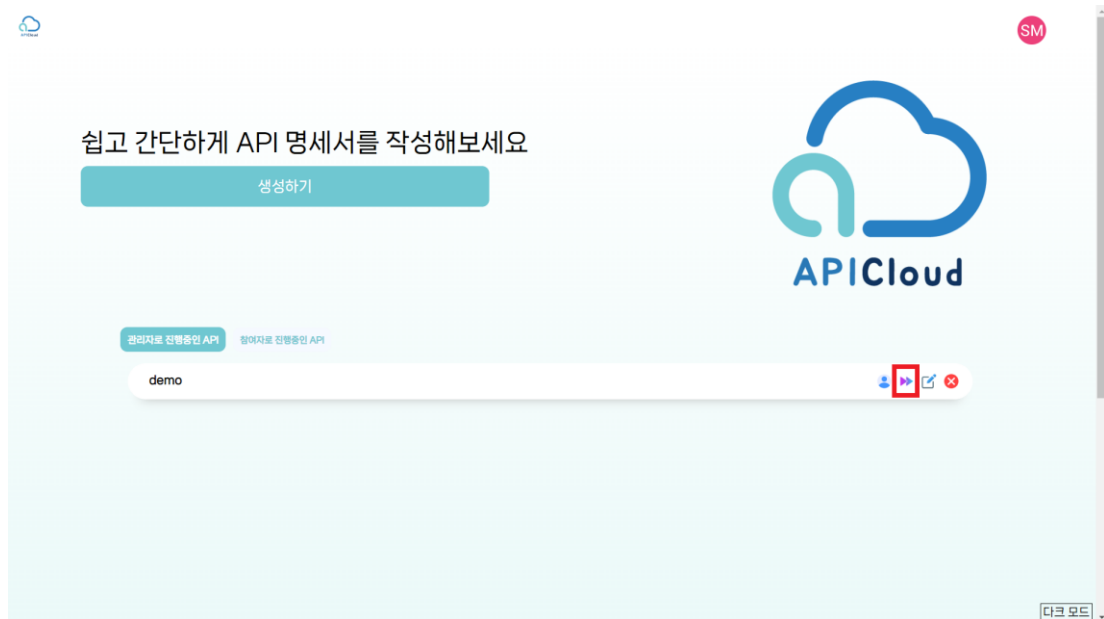


iii. 문서 생성



3. API Doc 작성 페이지

A. 작성 페이지로 이동



B. 컨트롤러 생성



C. api 생성



D. parameter, query, requestBody, responses 수정



4. 프로젝트 추출

A. 프로젝트 추출



5. 프로젝트 동기화

A. 프로젝트 수정 후 동기화

The screenshots illustrate the steps to synchronize a project in APICloud:

- 동기화 버튼 클릭** (Click Sync button): The first screenshot shows the 'demo' project interface. The '동기화' (Sync) button is highlighted with a red box.
- 동기화 버튼 클릭** (Click Sync button): The second screenshot shows a 'Controller 유효성 검사 결과' (Controller Validation Result) dialog box. The '동기화' (Sync) button is highlighted with a red box.
- 변경 사항 확인하기** (Check for Changes): The third screenshot shows the 'apicloud' project interface. The '변경 사항 확인하기' (Check for Changes) dialog box is highlighted with a red box.

- 변경사항 확인

B. Api 문서 수정 후 동기화

The top screenshot shows the APICloud interface for a demo API. A dialog box titled "Controller 유효성 검사 결과" (Controller Validity Check Result) is displayed, asking "계속 진행하시겠습니까?" (Do you want to continue?). The dialog lists several validation errors: "properties 필수값 부족" (0), "필수값 부족" (0), "타입 설정 오류" (1), and "중복된 이름" (0). The "동기화" (Synchronize) button is highlighted with a red box.

The bottom screenshot shows the same interface with the "responses" tab selected. The "User" model is selected, and the "code" tab is active, displaying the following Java code:

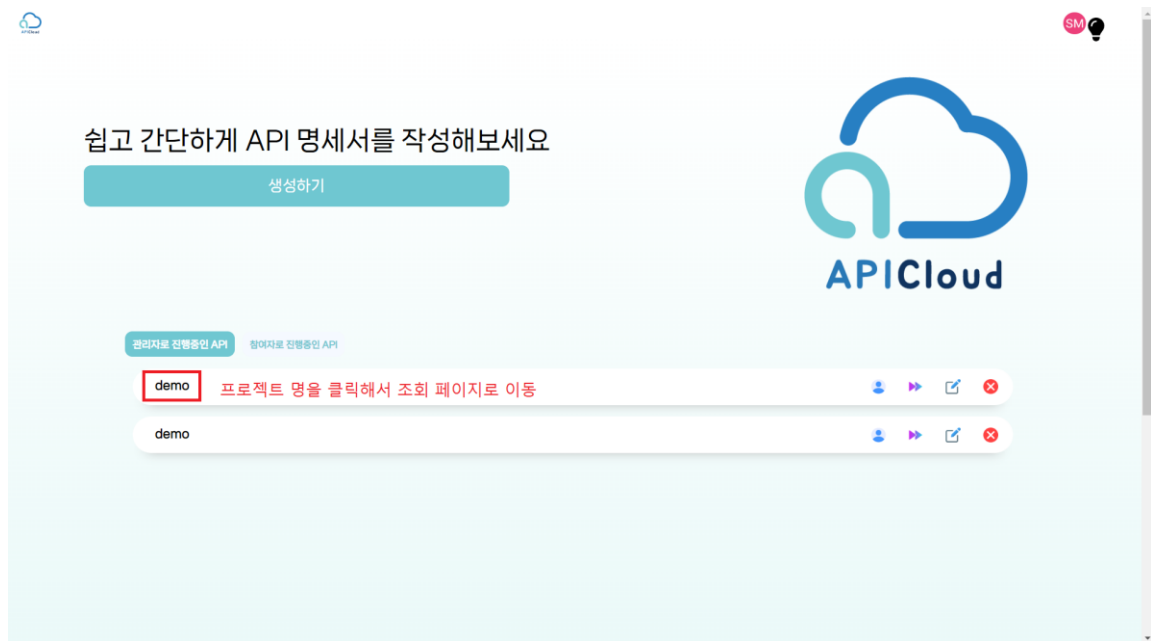
```

1 package com.example.demo.dto.a;
2
3 /** APICloud를 통해 추가된 import 문록입니다 */
4 import java.util.List;
5
6 public class User {
7
8     private List<String> name;
9     public String getName() {
10         return this.name;
11     }
12     public void setName(String name) {
13         this.name = name;
14     }
15     public String getQuery() {
16         return this.query;
17     }
18     public void setQuery(String query) {
19         this.query = query;
20     }
21     public Integer getAge() {
22         return this.age;
23     }
24     public void setAge(Integer age) {
25         this.age = age;
26     }
27 }
    
```

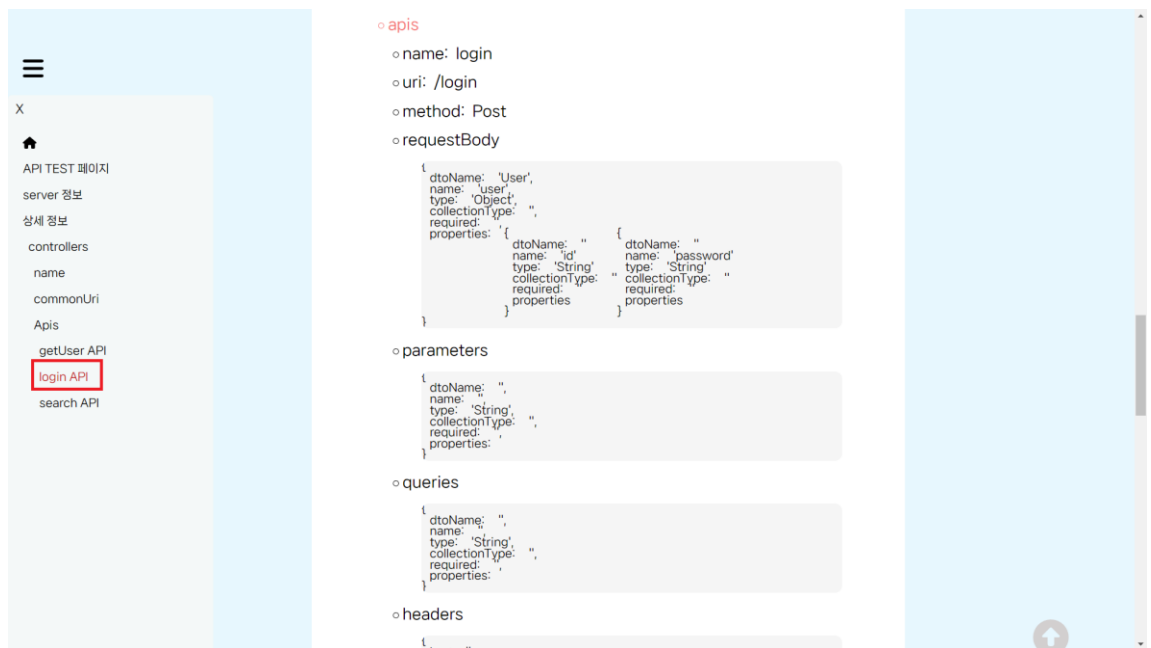
- 수정된 코드 확인 가능

6. API Doc 조회 페이지

A. 메인 페이지에서 이동



B. API 조회



- 왼쪽 사이드바의 API를 클릭하면 해당 API 정보 위치로 이동

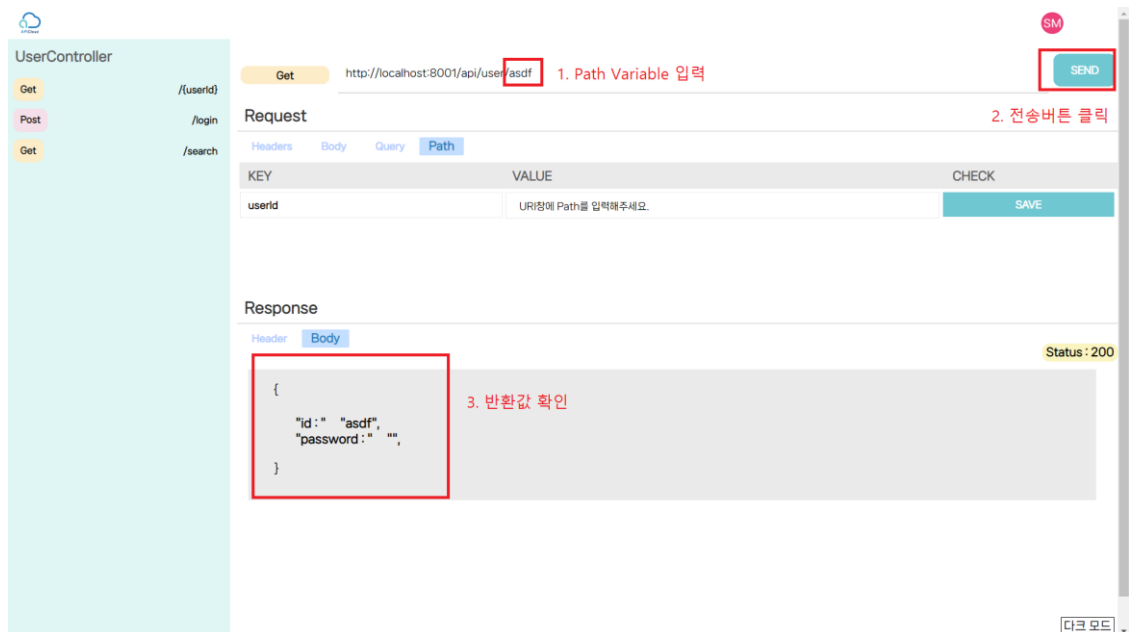
7. API Test 페이지

A. 테스트 페이지 이동



B. API 요청 보내기

i. Parameter



ii. Request Body

The screenshot shows the APICloud interface for a project named 'UserController'. On the left, a list of endpoints is shown: `/userid` (GET), `/login` (POST), and `/search` (GET). The main area displays a POST request to `http://localhost:8001/api/user/login`. The 'Request' tab is active, showing a JSON body: `{ "id": "asdf", "password": "asdfd" }`. A red box highlights this body with the annotation '1. Request Body 입력'. The 'Response' tab shows a JSON body: `{ "id": "asdf", "password": "asdfd", }`, with a red box highlighting it and the annotation '3. 반환값 확인'. The status is 'Status: 200'. A 'SEND' button is in the top right, with a red box around it and the annotation '2. 전송버튼 클릭'.

iii. Query

The screenshot shows the APICloud interface for the same 'UserController' project. The main area displays a GET request to `http://localhost:8001/api/user/search?userId=qwerf`. The 'Request' tab is active, showing a table with query parameters:

| KEY | VALUE | CHECK |
|--------|-------|--------------------------|
| userId | | <input type="checkbox"/> |

A red box highlights the URL's query parameter `?userId=qwerf` with the annotation '1.1. ? 입력 시 자동으로 필요한 쿼리 key 생성'. Another red box highlights the value `qwerf` with the annotation '1.2. 보낼 쿼리 value 작성'. The 'Response' tab shows a JSON body: `{ "id": "qwer", "password": "" }`, with a red box highlighting it and the annotation '3. 반환값 확인'. The status is 'Status: 200'. A 'SEND' button is in the top right, with a red box around it and the annotation '2. 전송버튼 클릭'.