

# [파이썬 트랙] 4회차 월말평가 – REST API



## 시험 과목

- ✓ Django Framework
- ✓ Django REST API

## 시험 목적

- ✓ API Service를 구현할 수 있다.
- ✓ 에러 상황을 수정할 수 있다.

## 시험 유의 사항

- 1) 성실하게 테스트에 임할 것 (부정 행위 적발 시 강력 조치)
- 2) 코드 유사도 판단 프로그램 기준, 부정 행위로 판단 될 시  
0점 처리 및 학사 기준에 의거 조치
- 3) 주어진 DB를 이용하여 문제를 풀 것 (migration 필요 없음)
- 4) 문제에 명시된 파일이 있는 경우, 반드시 해당 파일에서 수정할 것
- 5) 각 문제에 명시된 URL은 <http://127.0.0.1:8000> 이 생략되어 있음

## 시험 코드 작성 유의 사항

**최종 제출 코드가 다음 항목에 해당하는 경우, 감점 혹은 0점 처리 될 수 있음**

- 1) Syntax Error 혹은 기타 이유로 인한 서버 실행이 불가능한 경우
- 2) 출력 결과에 정답과 무관하거나 불필요한 내용이 있는 경우
- 3) 코드 수정 및 추가가 금지된 영역에 코드가 작성된 경우

# [파이썬 트랙] 4회차 월말평가 – REST API



## 시험 환경

- 개발환경 : python 3.9+, Django 4.2+, vscode, Postman
- 가상환경을 생성하여 주어진 requirements.txt 를 이용하여 환경 구축  
(이 때, 제시된 requirements.txt 이외의 패키지 설치 불가)
- 필요하다면 테스트를 위한 데이터 생성 가능
- 필드 정보는 surveys/models.py 를 참고
- Postman 요청시 URL 마지막에 `/` 를 반드시 붙여야 함

## 정답 제출 안내

**제출 안내 사항 미 준수 시, 감점 혹은 0점 처리 될 수 있음**

### **1) 반드시 서버가 정상적으로 동작하는 코드인지 확인 (중요)**

- 문제를 풀지 못하면 주석 처리하여 서버 실행에 이상없도록 만들 것

### **2) 압축 파일 이름**

- 지역0반\_홍길동  
ex) 서울1반\_홍길동 / 부울경2반\_김싸피

### **3) 압축시 주의 사항**

- 서술형 마크다운 파일도 같이 압축
- **venv/** 폴더는 제외 후 압축

제출 마감 직전, 서버 요청이 집중될 수 있으므로, 미리 제출하는 것을 권장.

**(마감 시간 이후 제출 불가)**

# [파이썬 트랙] 4회차 월말평가 – REST API



## 문제 01. TopicListSerializer 필드 제한 하기 (10점)

- ❖ Request : **GET** /api/v1/
- ❖ 아래 이미지와 같이 title 필드만 나타나도록 수정하시오.
- ❖ 수정 파일 : **surveys/serializers.py**
- ❖ **결과 이미지)**

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

- Method:** GET
- URL:** http://127.0.0.1:8000/api/v1/
- Send Button:** A blue button labeled "Send".
- Tabs:** Params, Auth, Headers (6), Body, Pre-req., Tests, Settings. The "Params" tab is selected.
- Query Params:** A table with columns KEY, VALUE, DESCRIPTION, and Bulk Edit. The first row shows "Key" and "Value".
- Body:** A dropdown menu is open, showing "Body" selected. The response is displayed in the "Pretty" tab.
- Response Status:** 200 OK, 3 ms, 378 B. A "Save Response" button is visible.
- JSON Response:**

```

1  [
2    {
3      "title": "Music"
4    },
5    {
6      "title": "Movie"
7    },
8    {
9      "title": "Drama"
10   }
11 ]

```

# [파이썬 트랙] 4회차 월말평가 – REST API

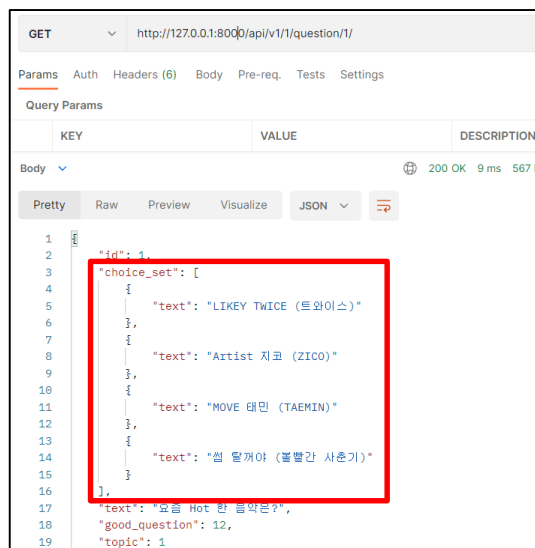


## 문제 02. Question 생성시 require field 문제 해결하기 (10점)

- ❖ Request : **POST** /api/v1/<int:topic\_pk>/question/
- ❖ topic 필드의 " **This field is required.**" 에러를 해결하여 Question이 정상적으로 생성되도록 관련 serializer를 수정하시오.
- ❖ 수정 파일 : **surveys/serializers.py**

## 문제 03. Question Detail 응답에 새로운 필드 추가하기 1 (10점)

- ❖ Request : **GET** /api/v1/<int:topic\_pk>/question/<int:question\_pk>/
- ❖ 아래 이미지와 같이 내부에 작성된 QuestionChoiceSerializer 를 이용하여 choice\_set 필드를 추가하시오.
- ❖ 수정 파일 : **surveys/serializers.py**
- ❖ 결과 이미지)



# [파이썬 트랙] 4회차 월말평가 – REST API



## 문제 04. Question Detail 응답에 새로운 필드 추가하기 2 (10점)

- ❖ Request : **GET** /api/v1/<int:topic\_pk>/question/<int:question\_pk>/
- ❖ 아래 이미지와 같이 answer의 개수를 보여주는 새로운 필드 total\_answer 를 추가하시오.  
(필드 이름은 반드시 total\_answer 로 작성)
- ❖ 수정 파일 : communities/serializers.py
- ❖ 결과 이미지)

```

1  {
2    "id": 1,
3    "choice_set": [
4      {
5        "text": "LIKEY TWICE (트와이스)"
6      },
7      {
8        "text": "Artist 지코 (ZICO)"
9      },
10     {
11       "text": "MOVE 태민 (TAEMIN)"
12     },
13     {
14       "text": "싫 땀겨야 (불빨간 사춘기)"
15     }
16   ],
17   "total_answer": 67,
18   "question": "Q. 어떤 노래를 좋아하나요?",
19   "good_question": 12,

```

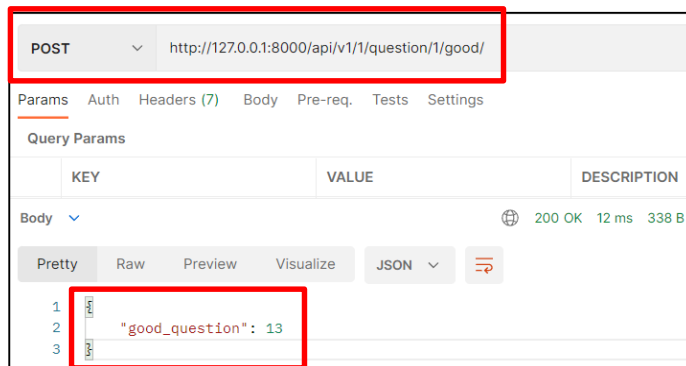
## 문제 05. Question Delete 동작 후 적절한 status code 응답하기 (10점)

- ❖ Request : **DELETE** /api/v1/<int:topic\_pk>/question/<int:question\_pk>/
- ❖ Question 삭제 메시지 정보와 함께 삭제 했음을 나타낼 수 있는 status code 가  
응답되도록 코드를 수정하시오.
- ❖ 수정 파일 : surveys/views.py

# [파이썬 트랙] 4회차 월말평가 – REST API

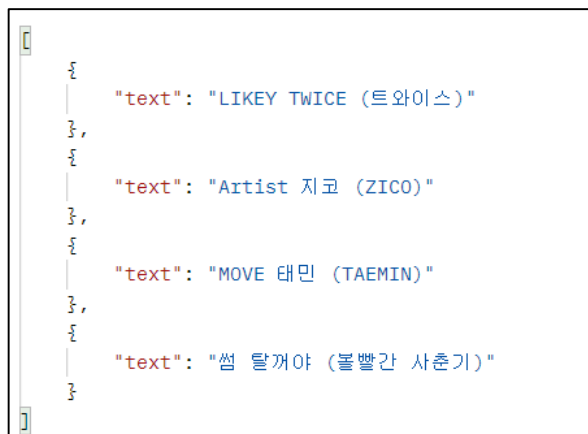
## 문제 06. Good Question 추천 동작 구현하기 (5점)

- ❖ Request : **POST** /api/v1/<int:topic\_pk>/question/<int:question\_pk>/good/
- ❖ 위 주소로 POST요청이 오면 Question 모델의 good\_question 필드의 값이 1 증가하고 DB에 저장되도록 코드를 작성하시오. (body로 전달되는 value는 없다.)
- ❖ 아래 이미지처럼 응답되는 필드는 Question 모델의 good\_question 필드 값을 이용한다.
- ❖ 수정 파일 : **surveys/views.py**
- ❖ **결과 이미지)**



## 문제 07. 특정 Question의 Choice만 응답하기 (5점)

- ❖ Request : **GET** /api/v1/<int:question\_pk>/choice/
- ❖ 요청한 question\_pk 의 선택지(choice)만 응답 결과로 나타날 수 있도록 코드를 수정하시오.
- ❖ 수정 파일 : **surveys/views.py**
- ❖ **예시 이미지) 1번 question – choice (4개) 인 경우**



# [파이썬 트랙] 4회차 월말평가 – REST API



## 문제 08. Choice 생성시 에러 해결하기 (10점)

- ❖ Request : **POST** /api/v1/<int:question\_pk>/choice/
- ❖ 선택지(choice)를 추가할 때, NOT NULL constraint failed 에러가 발생하고 있다.
- ❖ 정상적으로 선택지 (choice) 가 생성되도록 코드를 수정하십시오.
- ❖ 수정 파일 : **surveys/views.py**
- ❖ **에러 이미지)**

```
IntegrityError at /api/v1/10/choice/
NOT NULL constraint failed: surveys_choice.question_id

Request POST
Method:
Request http://127.0.0.1:8000/api/v1/10/choice/
URL:
Django 4.2.11
Version:
Exception IntegrityError
Type:
Exception NOT NULL constraint failed: surveys_choice.question_id
Value:
```

## 문제 09. Answer List 요청시 에러 해결하기 (10점)

- ❖ Request : **GET** /api/v1/<int:choice\_pk>/answer/
- ❖ Answer List를 GET으로 요청하게 되면 'AttributeError'가 발생하게 된다.
- ❖ 이를 수정하여 정상적으로 답변 리스트가 나올 수 있도록 코드를 수정하십시오.
- ❖ 수정 파일 : **surveys/views.py**
- ❖ **에러 이미지)**

```
AttributeError at /api/v1/6/answer/

Got AttributeError when attempting to get a value for field `comment`
on serializer `AnswerListSerializer`.
The serializer field might be named incorrectly and not match any
attribute or key on the `QuerySet` instance.
Original exception text was: 'QuerySet' object has no attribute
'comment'.

Request GET
Method:
Request http://127.0.0.1:8000/api/v1/6/answer/
URL:
Django 4.2.11
Version:
```

# [파이썬 트랙] 4회차 월말평가 – REST API



## 문제 10. Answer 생성 기능 구현하기 (10점)

- ❖ Request : **POST** /api/v1/<int:choice\_pk>/answer/
- ❖ 답변(Answer) 생성 기능이 아직 추가되지 않았다.
- ❖ AnswerSerializer를 이용하여 답변이 정상적으로 생성될 수 있도록 코드를 완성하시오.
- ❖ 수정 파일 : **surveys/views.py**

## 문제 11. Answer 수정 기능 보완하기 (10점)

- ❖ Request : **PUT** /api/v1/<int:choice\_pk>/answer/<int:answer\_pk>/
- ❖ Answer의 comment 필드만 수정하려고 하면 choice 필드가 " **This field is required.**" 라고 메시지가 전달되어 수정되지 않고 있다.
- ❖ Answer는 comment 필드 단독으로 수정이 가능해야 하며, choice 필드 또한 단독으로 수정 가능해야 한다.
- ❖ 각 필드들이 단독으로 수정될 수 있도록 코드를 보완하시오.
- ❖ 수정 파일 : **surveys/views.py**