

8 이제 추가한 todo 아이템을 삭제해보자.

```
(venv)$ curl -X 'DELETE' \  
  'http://127.0.0.1:8000/todo/1' \  
  -H 'accept: application/json'
```

윈도우에서는 다음 명령을 사용한다.

```
curl -X DELETE http://127.0.0.1:8000/todo/1 -H 'accept: application/json'
```

실행 결과는 다음과 같다.

```
{  
  "message": "Todo deleted successfully."  
}
```

9 GET 요청을 사용해서 삭제한 아이템이 추출되는지 확인해보자.

```
(venv)$ curl -X 'GET' \  
  'http://127.0.0.1:8000/todo/1' \  
  -H 'accept: application/json'
```

윈도우에서는 다음 명령을 사용한다.

```
curl -X GET http://127.0.0.1:8000/todo/1 -H 'accept: application/json'
```

명령을 실행하면 지정한 ID의 todo가 존재하지 않는다는 응답을 확인할 수 있다.

```
{  
  "message": "Todo with supplied ID doesn't exist."  
}
```

지금까지 배운 내용을 활용해서 CRUD 애플리케이션을 만들었다. 모델을 통해 요청 바디를 검증하여 적절한 데이터를 API에 전송했으며 경로 매개변수를 라우트에 추가해 `todo_list`에서 단일 아이템을 추출하거나 삭제할 수 있게 했다.