

스프링 스타-디

# 포스트맨 사용법



## 포스트맨 사용법

### 포스트맨 이란?

API 개발 및 테스트에 사용

협업 기반의 플랫폼 및 도구

REST API나 GraphQL API를  
테스트하고 디버깅할 때 널리 사용



## 포스트맨 사용법

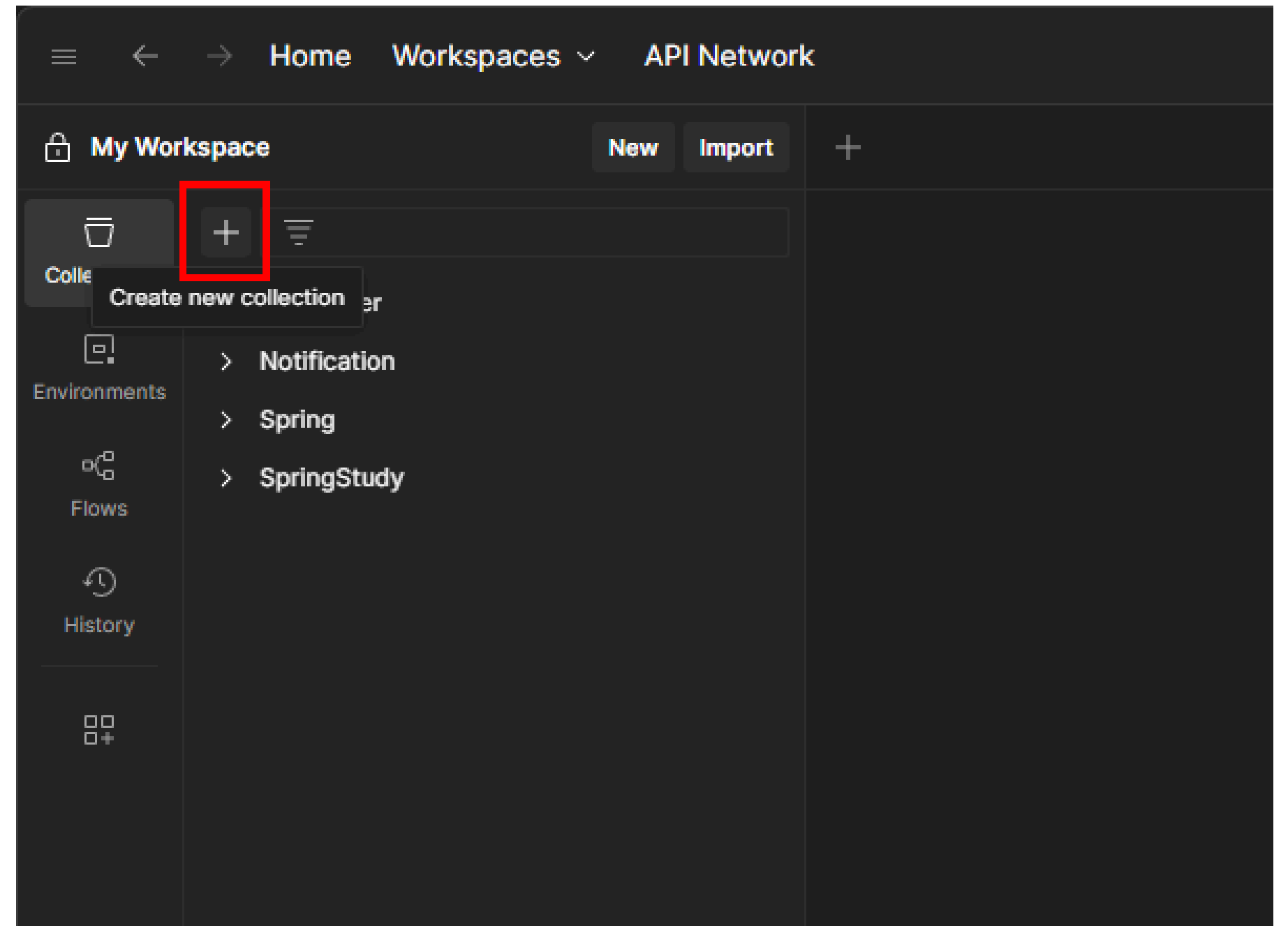
### 포스트맨 새로운 요청 생성하기

새로운 컬렉션을 생성 후 요청 만들기

1. 좌측 상단의 '+' 버튼 클릭
2. Blank collection 클릭
3. 원하는 이름 설정
4. 컬렉션 마우스 호버 후 점 3개 버튼 클릭
5. Add request 클릭하여 새로운 요청 생성

Add folder를 활용하여 요청을  
종류별로 정리 할 수도 있음

1



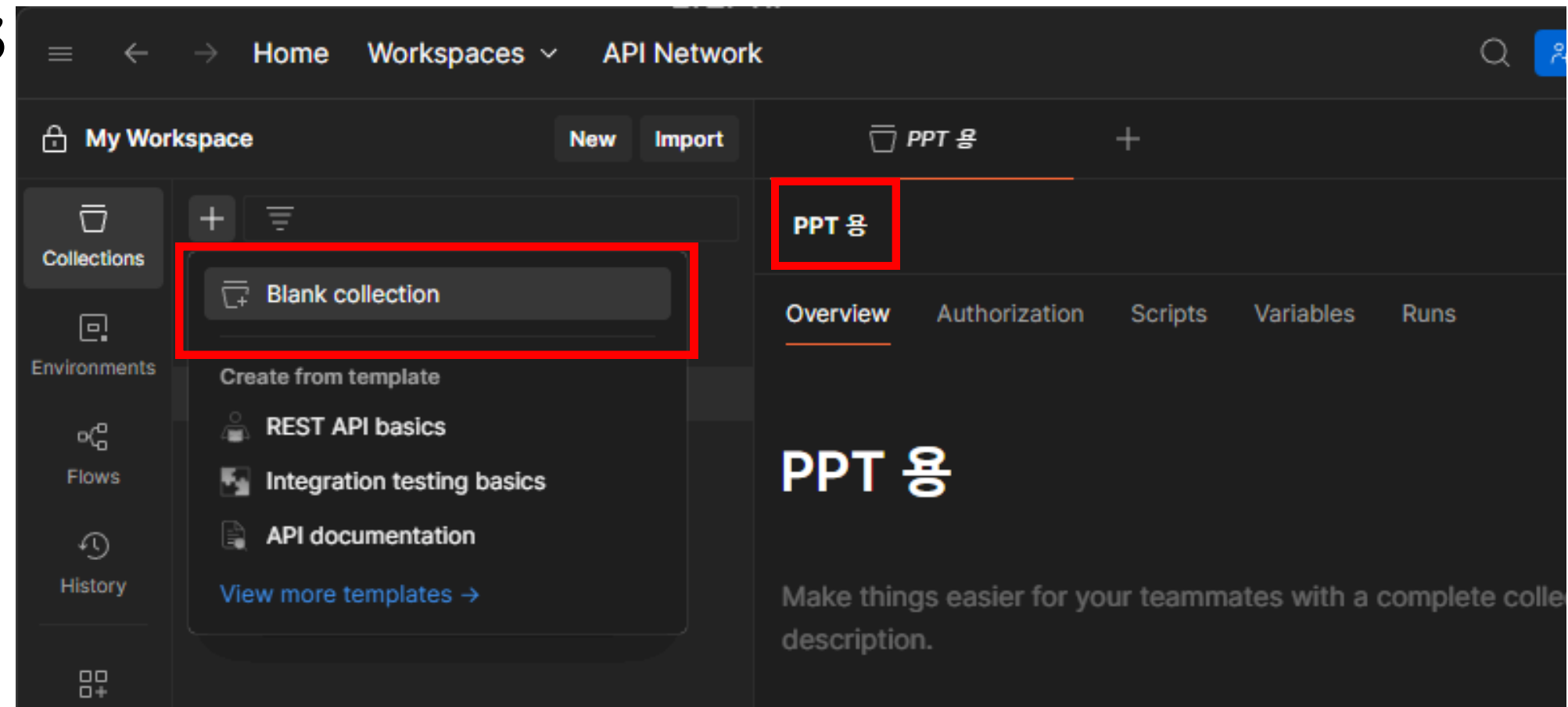
## 포스트맨 사용법

### 포스트맨 새로운 요청 생성하기

새로운 컬렉션을 생성 후 요청 만들기

1. 좌측 상단의 '+' 버튼 클릭
  2. Blank collection 클릭
  3. 원하는 이름 설정
  4. 컬렉션 마우스 호버 후 점 3개 버튼 클릭
  5. Add request 클릭하여 새로운 요청 생성
- Add folder를 활용하여 요청을  
종류별로 정리 할 수도 있음

2, 3



## 포스트맨 사용법

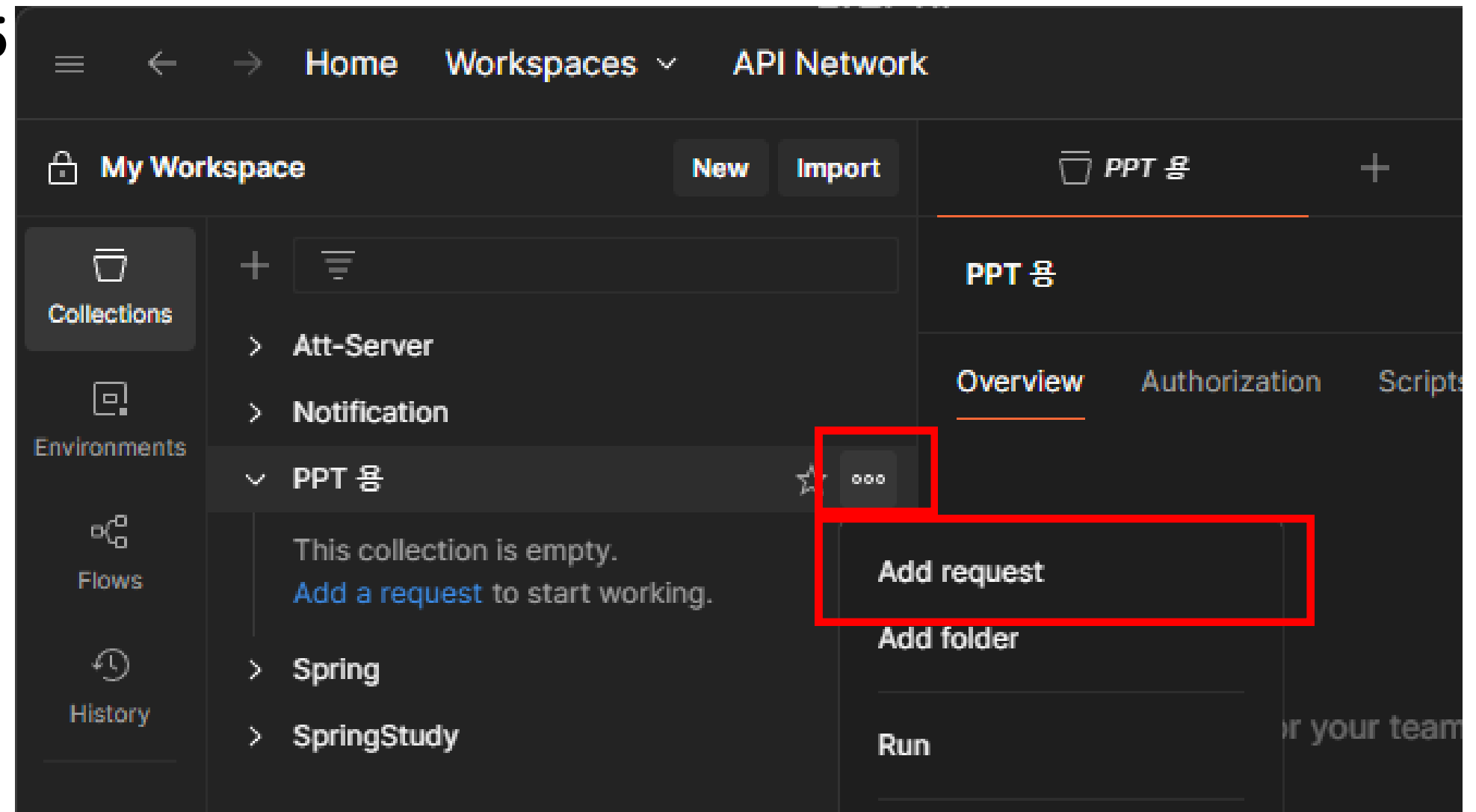
### 포스트맨 새로운 요청 생성하기

새로운 컬렉션을 생성 후 요청 만들기

1. 좌측 상단의 '+' 버튼 클릭
2. Blank collection 클릭
3. 원하는 이름 설정
4. 컬렉션 마우스 호버 후 점 3개 버튼 클릭
5. Add request 클릭하여 새로운 요청 생성

Add folder를 활용하여 요청을  
종류별로 정리 할 수도 있음

4, 5



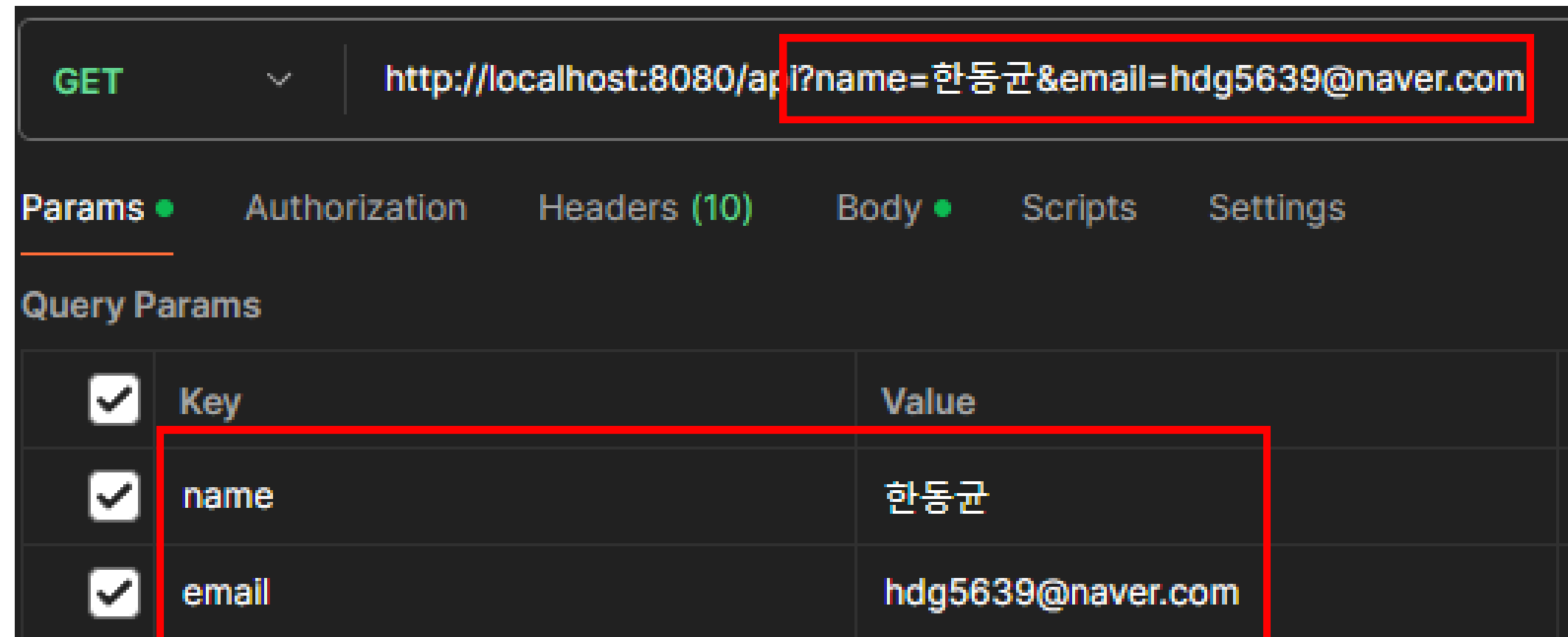
## — 포스트맨 사용법

### 요청 보내는 방법들

#### 요청 기본 화면 설명

##### Params

1. URL에 붙은 **쿼리 파라미터** 설정
2. ?id=123&name=홍길동 이런 느낌
3. **Spring**의 경우 **RequestParam** 변수
4. Key에 변수명, Value에 값을 기입



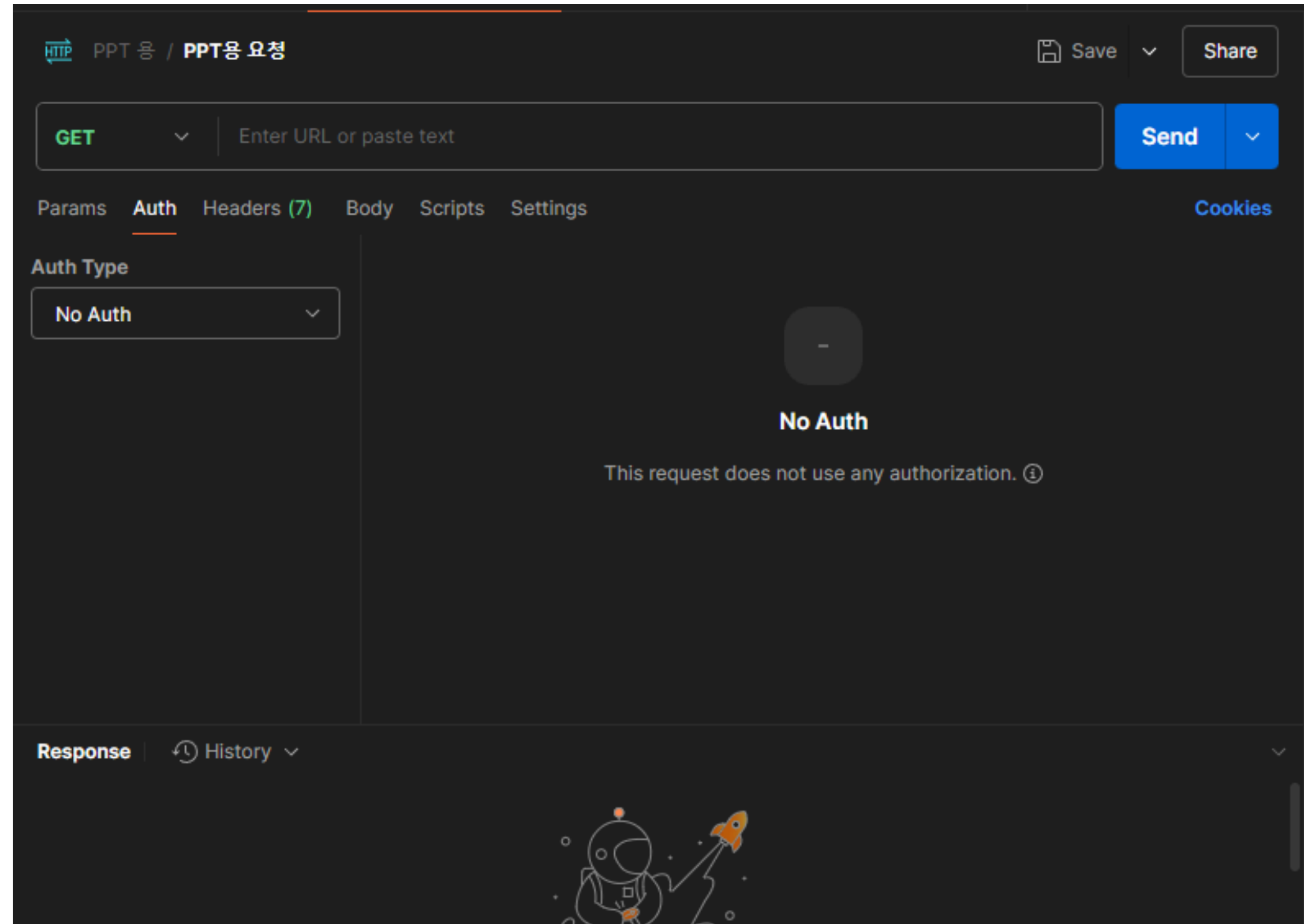
## 포스트맨 사용법

## 요청 보내는 방법들

### 요청 기본 화면 설명

#### Authorization

1. **Auth Type**에 따라 토큰 자동 생성
2. 만들어준 토큰을 **Headers**에 넣어줌
3. 아직까진 저걸 사용한 적이 없긴 함



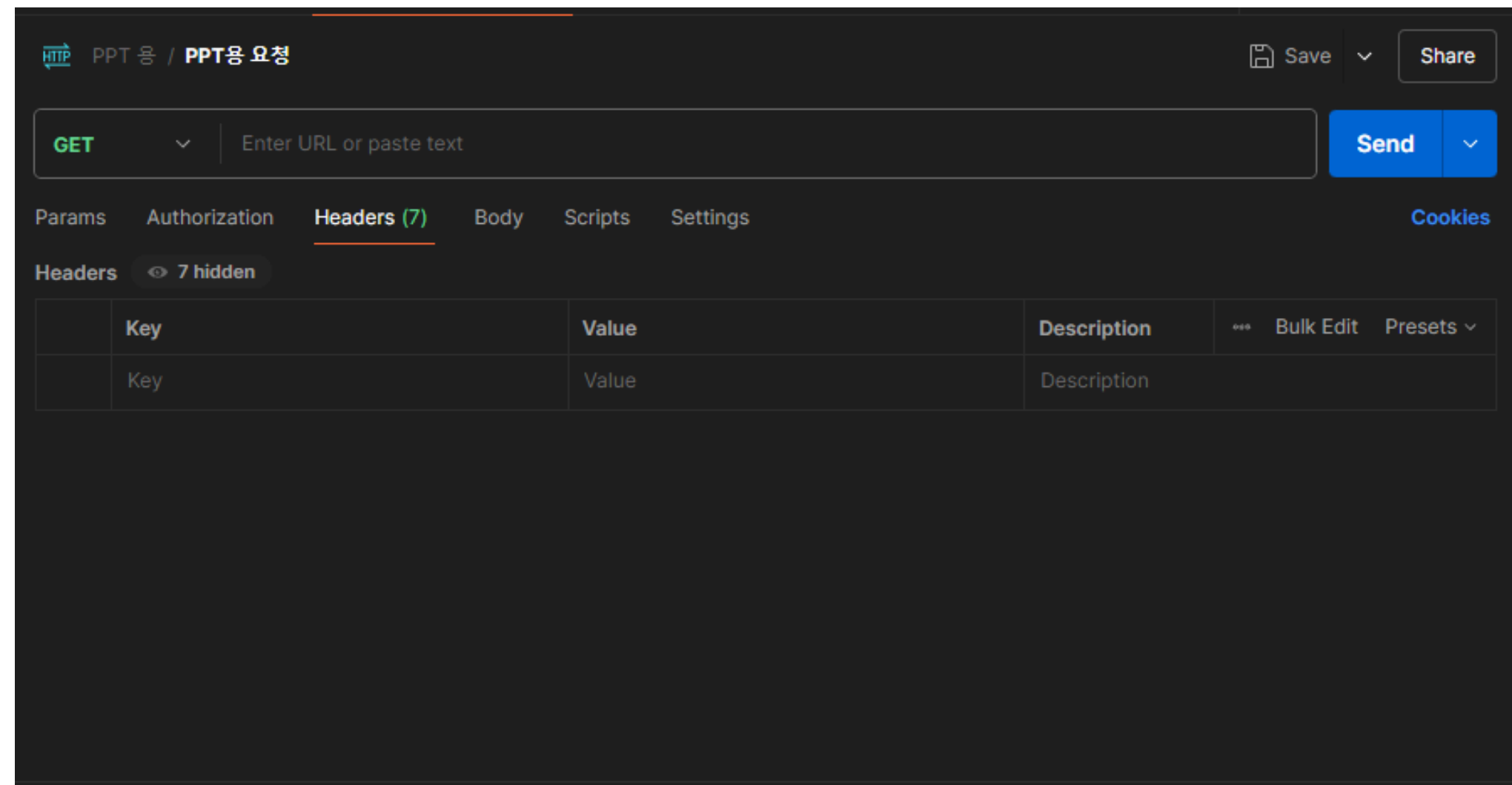
## 포스트맨 사용법

## 요청 보내는 방법들

### 요청 기본 화면 설명

#### Headers

1. 요청에 함께 전송할 **헤더 정보**를 설정
2. **JWT**의 **Bearer Token** 같은 것을 넣어 줄 수 있음
3. 콘텐츠 타입 혹은 인증 보낼 때 자주 사용
4. Key에 어떤 정보를 보낼 지 타입 작성
5. value에 해당 구문에 맞는 내용 작성



	Key	Value
<input checked="" type="checkbox"/>	Authorization	Bearer eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJ...
	Key	Value

이런 느낌



## — 포스트맨 사용법

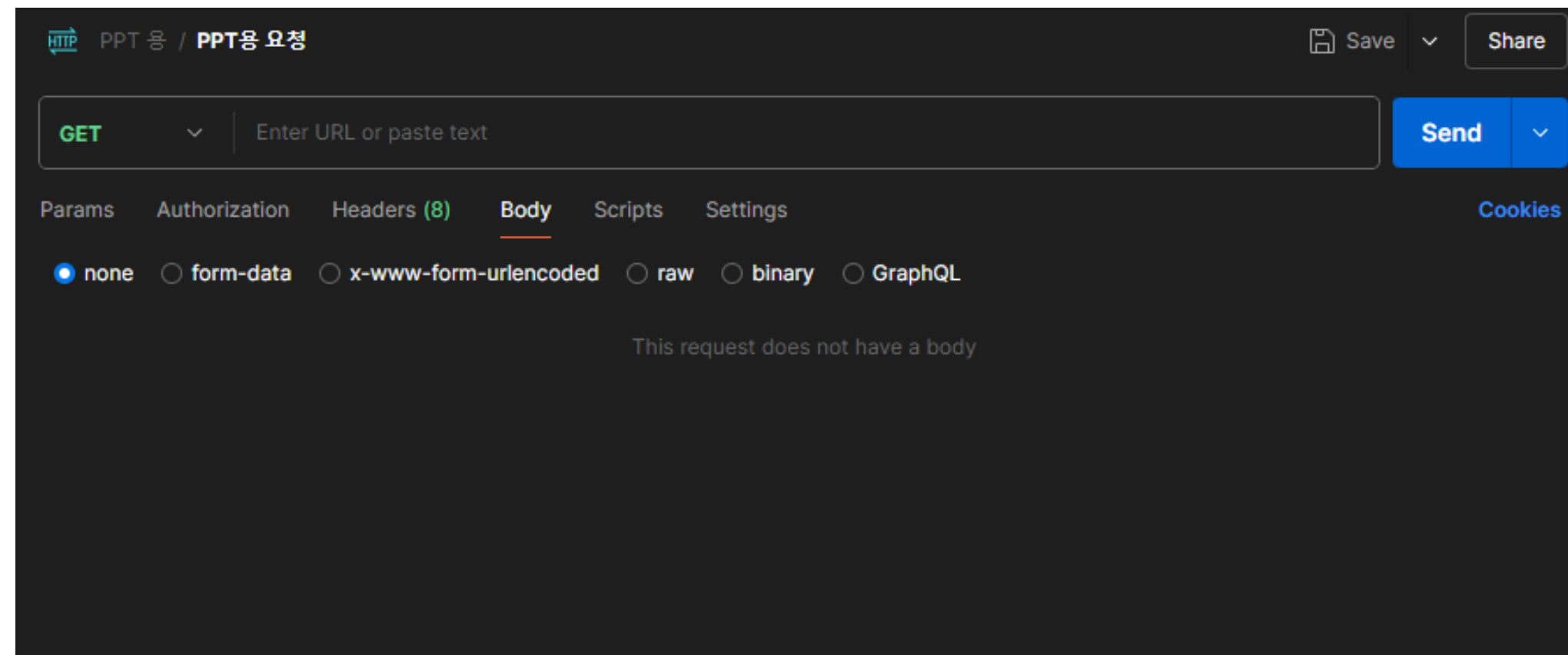
### 요청 보내는 방법들

#### 요청 기본 화면 설명

##### Body

1. 주로 **POST, PUT** 에서 본문을 설정
2. raw JSON, form-data 등 형식 선택
3. 웬만하면 raw를 주로 쓰게 될 것임
4. 이미지 서버를 한다면 form-data도 사용할 때가 생김

Scripts, Settings는 한번도 안써봐서 넘기겠습니다.



## — 포스트맨 사용법

### 요청 JSON 작성법

#### 작성 문법

1. 문자열은 "큰따옴표"로 감싸기
2. 키:값 구조를 { } 안에 작성
3. 마지막 항목에는 쉼표(,) 붙이면 안 됨

```
1 {  
2     "name": "한동균",  
3     "email": "hdg5639@naver.com",  
4     "age": 25  
5 }
```

## 포스트맨 사용법

## 요청 JSON 작성법

### List 형식으로 보내기

1. 객체를 [] 안에 작성
2. 객체를 구분할 때 쉼표(,) 사용
3. 마찬가지로 마지막 항목에는 쉼표 금지

### List<String>일 때

```
1  [
2    "test",
3    "test2",
4    "test3"
5  ]
```

## — 포스트맨 사용법

## 요청 JSON 작성법

List 형식으로 보내기

1. 객체를 [] 안에 작성
2. 객체를 구분할 때 쉼표(,) 사용
3. 마찬가지로 마지막 항목에는 쉼표 금지

List<Entity>일 때

```
1  [  
2      {  
3          "name": "한동균",  
4          "email": "hdg5639@naver.com",  
5          "age": 25  
6      },  
7      {  
8          "name": "박상우",  
9          "email": "babsang@naver.com",  
10         "age": 27  
11     }  
12 ]
```

## — 포스트맨 사용법

### 요청 JSON 작성법

#### Map 형식으로 보내기

1. 객체를 {} 안에 작성
2. Map의 Key는 자료형에 맞게 작성
3. Value는 Key 뒤에 콜론(:)뒤에 시작
4. Value 내용은 동일하게 작성
5. 객체를 구분할 때 쉼표(,) 사용
6. 마찬가지로 마지막 항목에는 쉼표 금지

#### 예시로 가져온 Map<String, String>구조의 JSON

```
{  
  "testkey1": "test",  
  "testkey2": "test2"  
}
```

## — 포스트맨 사용법

## 요청 JSON 작성법

### Map 형식으로 보내기

1. 객체를 {} 안에 작성
2. Map의 Key는 자료형에 맞게 작성
3. Value는 Key 뒤에 콜론(:)뒤에 시작
4. Value 내용은 동일하게 작성
5. 객체를 구분할 때 쉼표(,) 사용
6. 마찬가지로 마지막 항목에는 쉼표 금지

## 예시로 가져온 Map<String, Entity>구조의 JSON

```
1  {
2      "testkey1": {
3          "name": "한동균",
4          "email": "hdg5639@naver.com",
5          "age": 25
6      },
7      "testkey2": {
8          "name": "박상우",
9          "email": "bsang@naver.com",
10         "age": 27
11     }
12 }
```

—  
포스트맨 사용법

## 요청 JSON 작성법

Map에 value 가 List일 때 작성법

1. 기본적인 구조는 일반적인 Map과 동일
2. 콜론(:) 뒤에 []로 객체를 묶어줌
3. 이외 다른 것은 일반적인 List와 동일

예시로 가져온 Map<String, List<String>>구조의 JSON

```
1 {  
2   "testkey1": [  
3     "test",  
4     "test2",  
5     "test3"  
6   ]  
7 }
```

## 포스트맨 사용법

## 요청 JSON 작성법

### Map에 value 가 List일 때 작성법

1. 기본적인 구조는 일반적인 Map과 동일
2. 콜론(:) 뒤에 []로 객체를 묶어줌
3. 이외 다른 것은 일반적인 List와 동일

### 예시로 가져온 Map<String, List<Entity>>구조의 JSON

```
1  {
2      "testkey1": [
3          {
4              "name": "한동균",
5              "email": "hdg5639@naver.com",
6              "age": 25
7          },
8          {
9              "name": "박상우",
10             "email": "babsang@naver.com",
11             "age": 27
12         }
13     ]
14 }
```

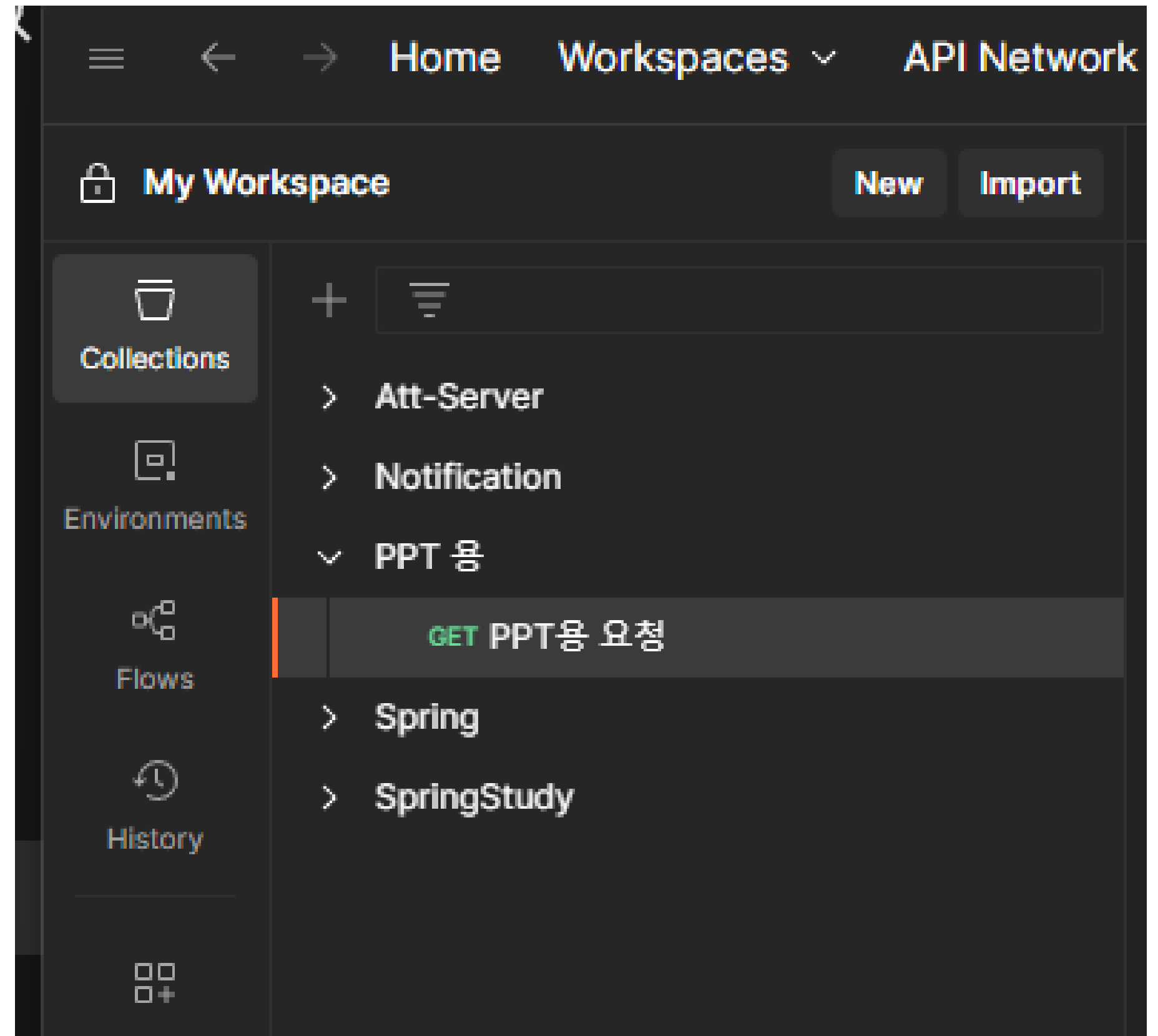


## 포스트맨 사용법

## 환경변수 사용법

포스트맨의 **환경변수**란?

1. API 요청에 **자주 쓰이는 값**을 변수로 저장
2. 예: 도메인 주소, 인증 토큰 등
3. **Collections** 하단 **Environments**에서  
편집 가능



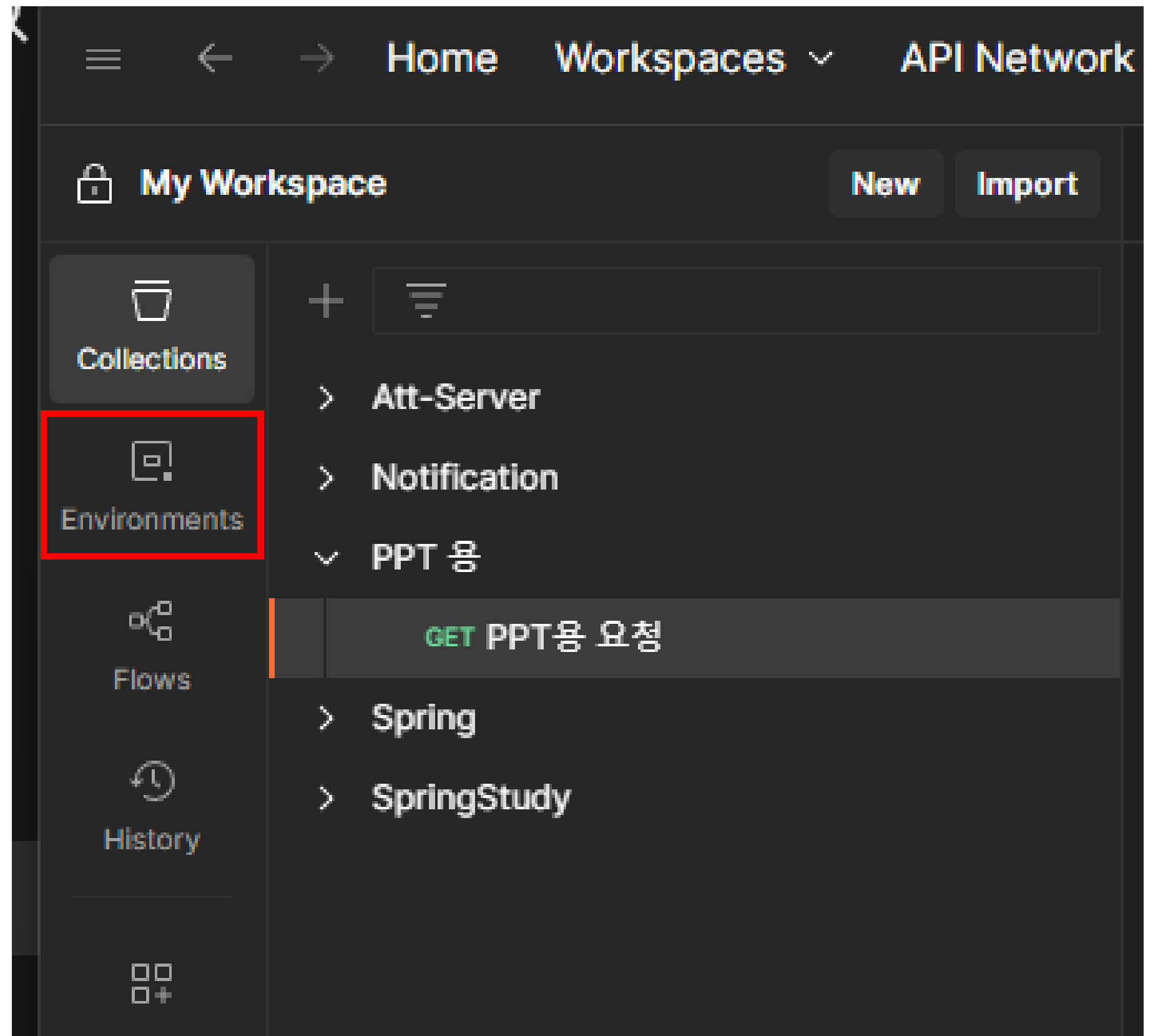
## 포스트맨 사용법

## 환경변수 사용법

### 환경변수 설정법

1. Collections 하단 Environments 클릭
2. 좌측 상단 + 버튼 클릭하면 New Env 생성됨
3. Variable에 이름 설정
4. Type은 default
5. Initial value는 기본 값
6. Current value는 현재 사용하는 실제 값

1



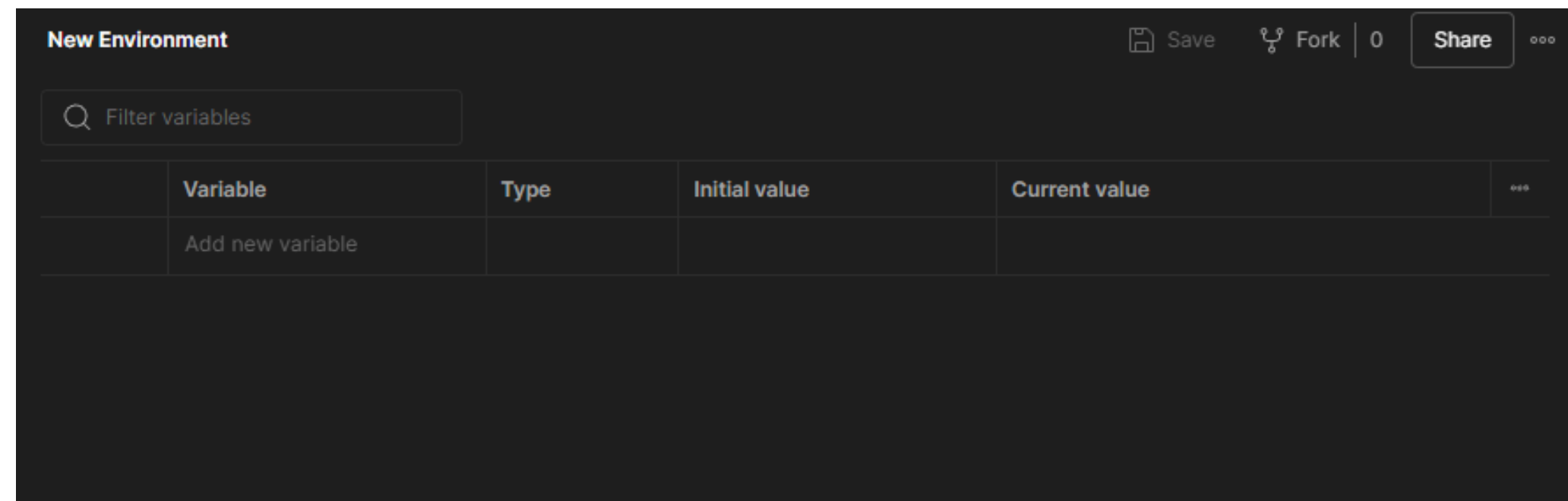
## 포스트맨 사용법

## 환경변수 사용법

### 환경변수 설정법

1. Collections 하단 Environments 클릭
2. 좌측 상단 + 버튼 클릭하면 New Env 생성됨
3. Variable에 이름 설정
4. Type은 default
5. Initial value는 기본 값
6. Current value는 현재 사용하는 실제 값

2



## 포스트맨 사용법

## 환경변수 사용법

### 환경변수 설정법

1. Collections 하단 Environments 클릭
2. 좌측 상단 + 버튼 클릭하면 New Env 생성됨
3. Variable에 이름 설정
4. Type은 default
5. Initial value는 기본 값
6. Current value는 현재 사용하는 실제 값

3~6

New Environment					Save	
Filter variables						
	Variable	Type	Initial value	Current value		
<input checked="" type="checkbox"/>	localhost	default	http://localhost:8080/api	http://localhost:8080/api		
<input checked="" type="checkbox"/>	user server	default	/user	/user		
	Add new variable					

## 포스트맨 사용법

## 환경변수 사용법

### Initial, Current 차이

1. **Initial**은 기본값이고 팀원과 공유됨
2. **Current**는 현재 사용중인 값, 공유 X
3. 민감한 값은 **Initial** 말고 **Current**

New Environment					Save	
<input type="text" value="Filter variables"/>						
	Variable	Type	Initial value	Current value		
<input checked="" type="checkbox"/>	localhost	default	http://localhost:8080/api	http://localhost:8080/api		
<input checked="" type="checkbox"/>	user server	default	/user	/user		
	Add new variable					

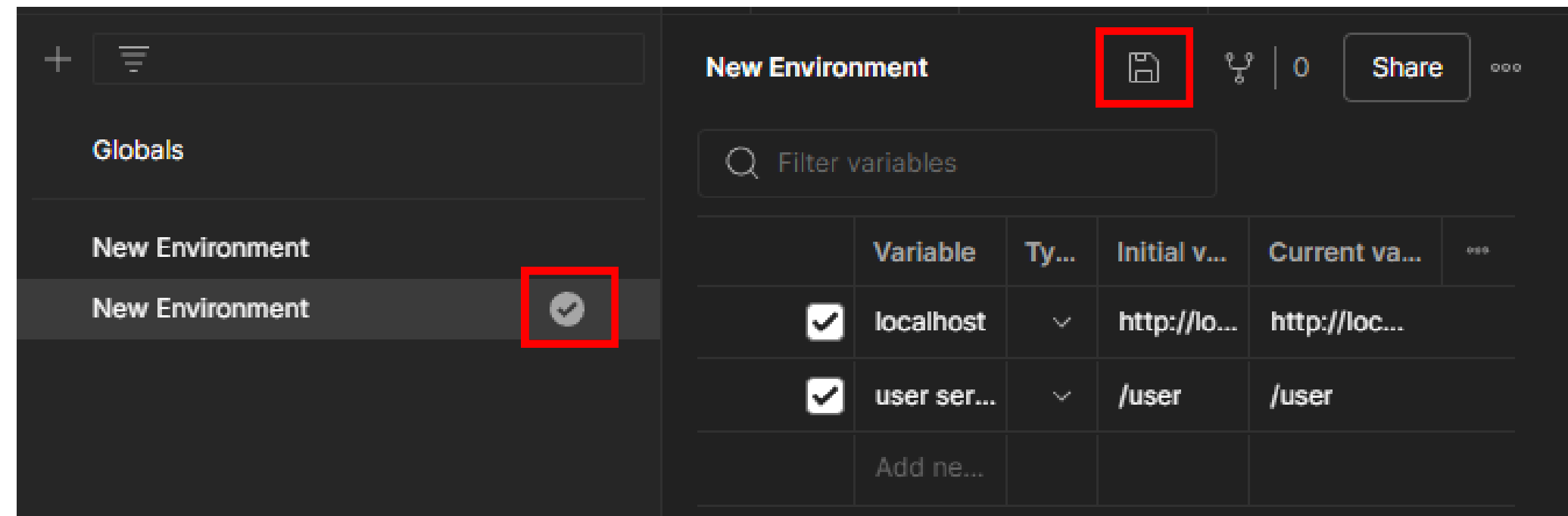
## 포스트맨 사용법

## 환경변수 사용법

### 환경변수 설정법

7. 다 작성 했으면 우측 상단 **save**
8. 디렉토리 우측 **체크 버튼** 클릭하여 활성화
9. 요청에서 사용할 때는 URL에 중괄호 두겁

7, 8



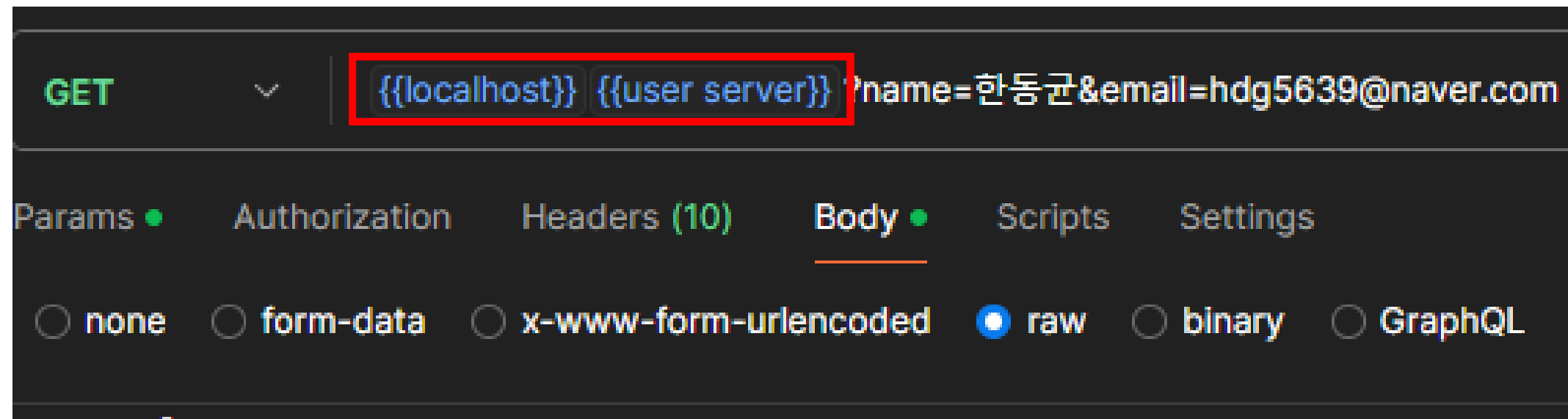
## 포스트맨 사용법

## 환경변수 사용법

### 환경변수 설정법

7. 다 작성 했으면 우측 상단 **save**
8. 디렉토리 우측 **체크 버튼** 클릭하여 활성화
9. 요청에서 사용할 때는 URL에 중괄호 두겹

9



# 감사합니다

스프링 스타-디