

# API MANAGER

## 간단 설명

여러분들의 시간은 소중한니까  
10분안에 끝낼게요

# INDEX

01

이거 뭐예요

02

이거 왜 만들었어요

03

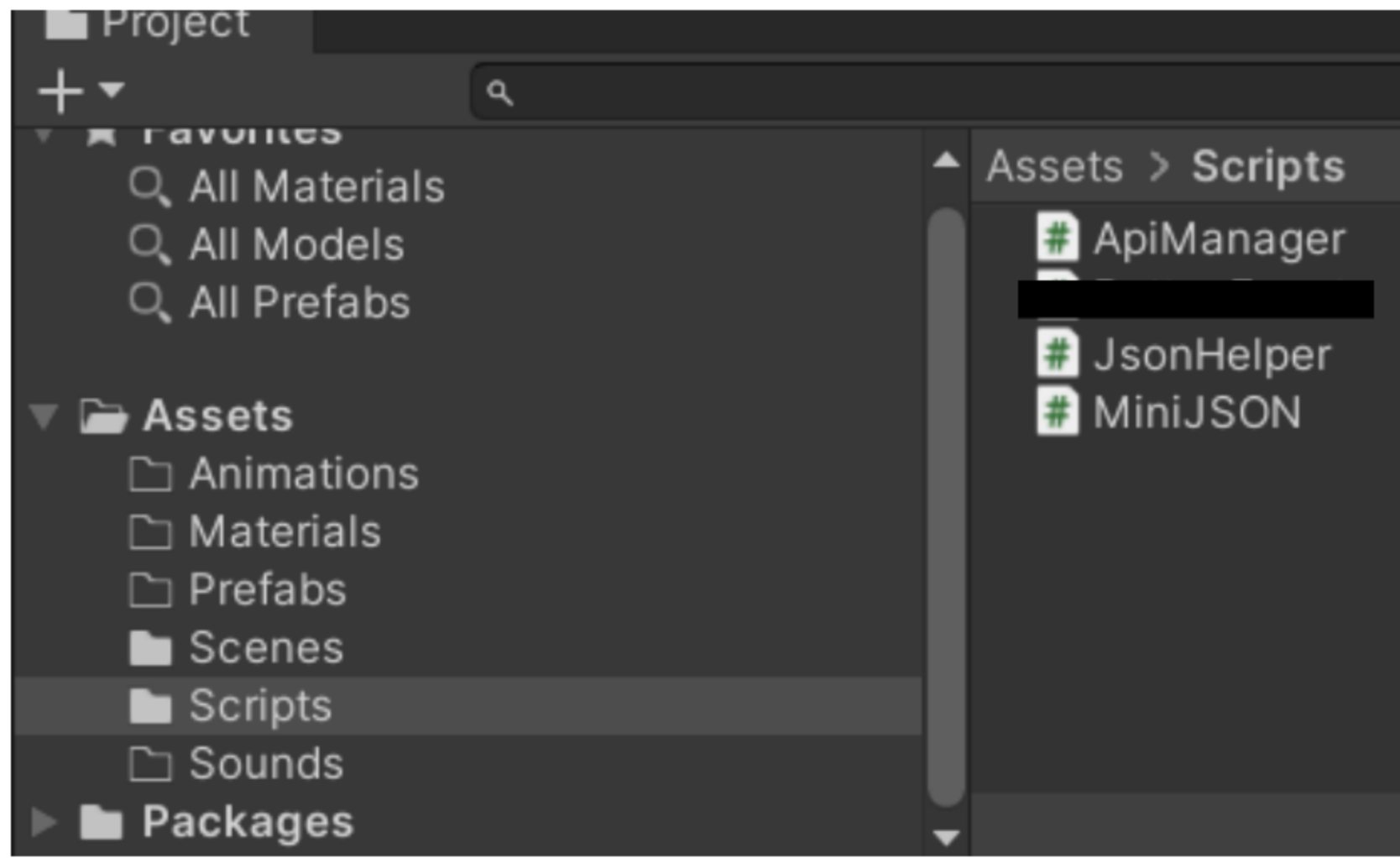
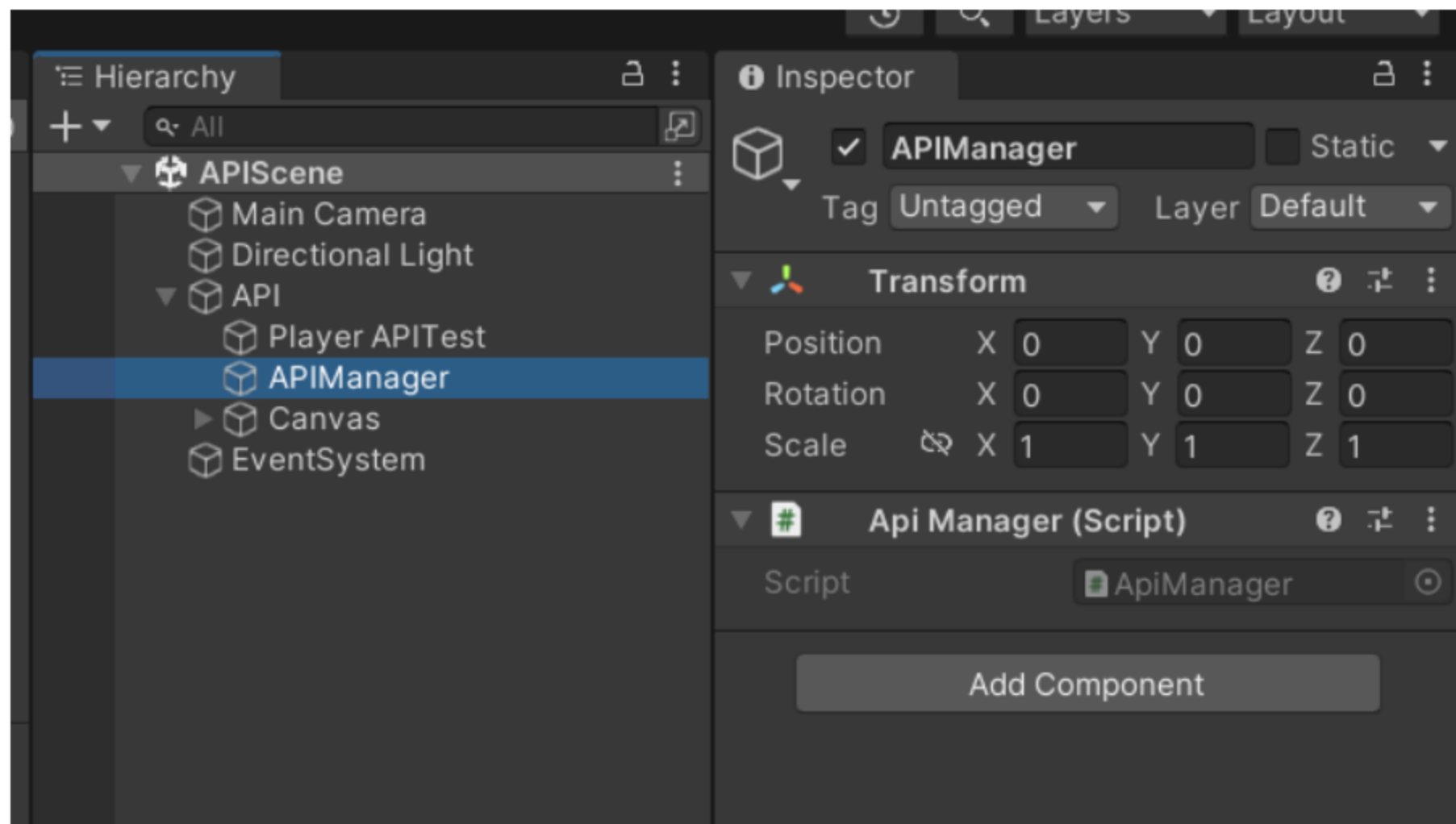
어떻게 쓰는거예요

04

작동원리

# 01 이거 뭐예요

유니티에서 API 통신을 쉽게 하기 위해서  
복잡하고 번거롭고 귀찮은 과정을 대신 해주는  
도구입니다



## 02 이거 왜 만들었어요

이거 없으면 아래의 것들을 매번 직접 해줘야합니다

1. Header에 액세스 토큰값 지정 <- 유저 식별용
2. 토큰 만료되면 리프레시 요청
3. 요청에 필요한 값 json 형태로 변환
4. 각 메서드에 맞게 요청값 변환

## 02 이거 왜 만들었어요

그리고 Coroutine, UnityWebRequest 같은  
알아도 별 쓸모없는 것들을 여럿이서 배워야 합니다

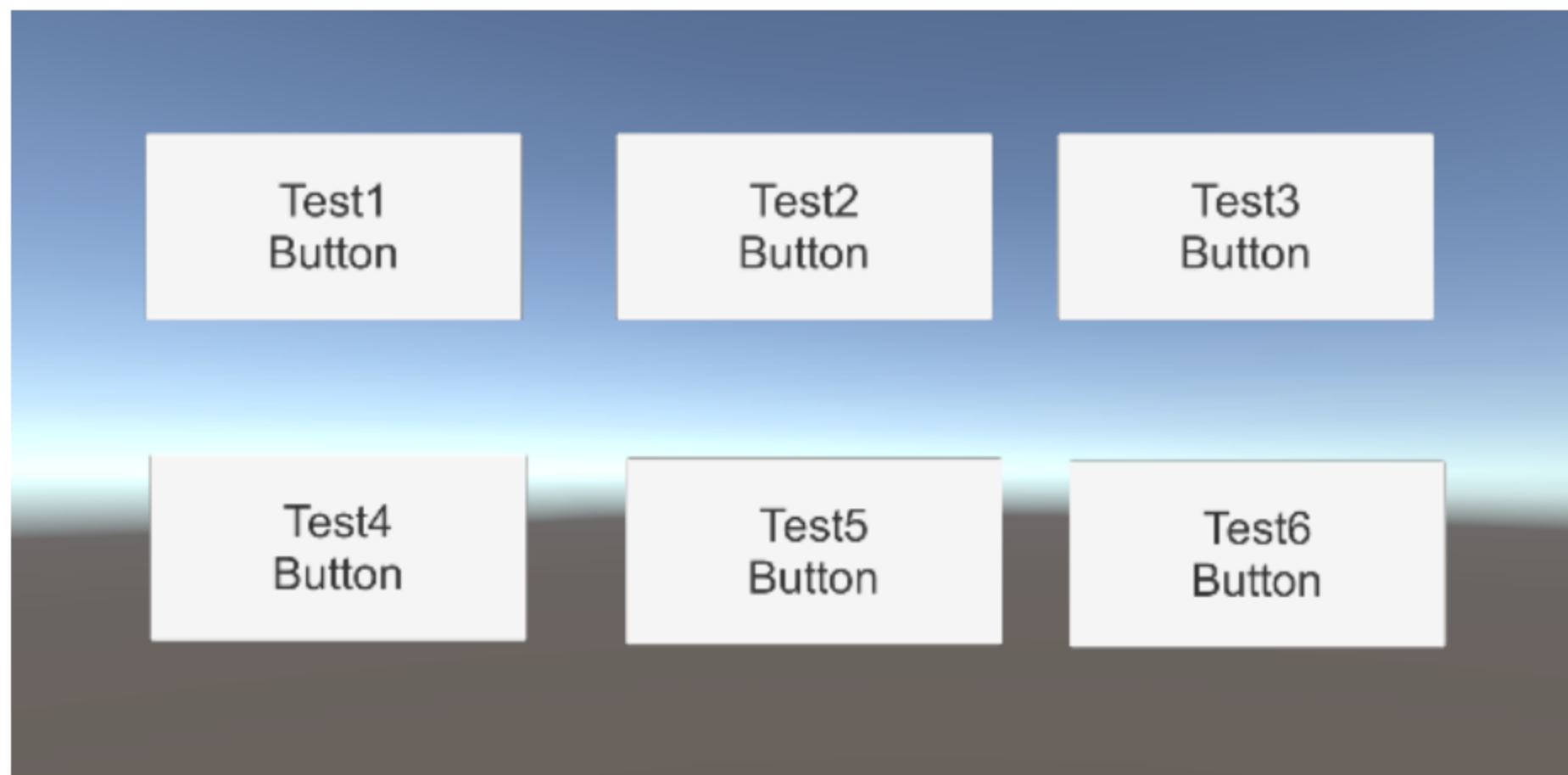
## 03 어떻게 쓰는 거예요

Git Branch - feat/unity/apimanager에 소스가 있습니다  
scripts 폴더의  
1. ApiManager  
2. MiniJSON  
3. JsonHelper를 챙기시면 됩니다  
나머진 버리세요

## 03 어떻게 쓰는 거예요

예시를 보여드리며 진행해보겠습니다

ButtonEvent → ApiManager





## 03 어떻게 쓰는 거예요

Get 요청은 이렇게 해주세요

메서드가 get이면 이곳에 GET 작성

참조 0개

```
public void OnClickTest1()
{
    ApiManager.Instance.GET("test/1", null, delegate (UnityWebRequest request)
    {
        Debug.Log(request.downloadHandler.text);
    });
}
```

요청할 주소

ex) api주소가 http://localhost/api/test/1이면  
test/1만 작성

사용할 수 있는 api 주소는

팀 google sheet를 참고해주세요  
계속 갱신하겠습니다

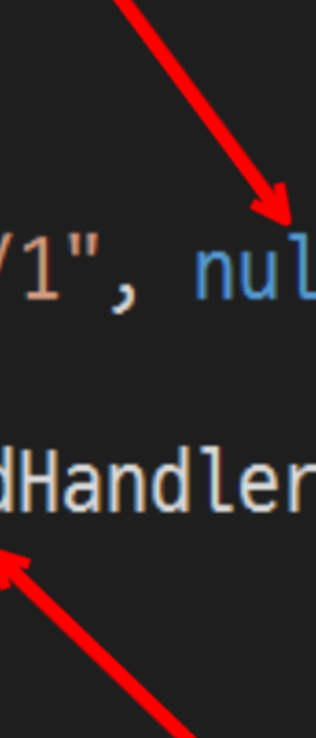


## 03 어떻게 쓰는 거예요

Get 요청은 이렇게 해주세요      parameter 혹은 body를 넣는 곳. 자세한건 후술

참조 0개

```
public void OnClickTest1()
{
    ApiManager.Instance.GET("test/1", null, delegate (UnityWebRequest request)
    {
        Debug.Log(request.downloadHandler.text);
    });
}
```



반환받은 값을 출력

## 03 어떻게 쓰는 거예요

Get 요청은 이렇게 해주세요 (반환값이 일반적인 String인 경우)

참조 0개

```
public void OnClickTest1()
{
    ApiManager.Instance.GET("test/1", null, delegate (UnityWebRequest request)
    {
        Debug.Log(request.downloadHandler.text);
    });
}
```

```
return new ResponseEntity<>(body: "api test 1 ", HttpStatus.OK);
```



[01:44:11] api test 1

UnityEngine.Debug:Log (object)

## 03 어떻게 쓰는 거예요

Get 요청은 이렇게 해주세요 (json으로 요청과 반환을 하는 경우)

```
var str = new Dictionary<string, object>();  
str.Add("nickname", "unity");  
str.Add("level", 1);  
var data = Json.Serialize(str);
```

dictionary 생성

dictionary에 값 추가  
애들이 ->  
?nickname=unity&level=1  
로 됩니다

json으로 직렬화

## 03 어떻게 쓰는 거예요

Get 요청은 이렇게 해주세요

생성한 parameter를 넣는 곳

```
var data = Json.Serialize(str);

ApiManager.Instance.GET("test/3", data, delegate (UnityWebRequest request)
{
    /*Debug.Log(request.downloadHandler.text);*/
    var dict = Json.Deserialize(request.downloadHandler.text) as Dictionary<string, object>;
    Debug.Log(dict["result"]);
    Debug.Log(dict["nickname"]);
    Debug.Log(dict["level"]);
});
```

반환받은 값(문자열)을 사전 형식으로 만들어  
값을 확인할 수 있음

## 03 어떻게 쓰는 거예요

Get 요청은 이렇게 해주세요

```
{"result":[  
  {"pk":1,"intCol":1,"strCol":"one"},  
  {"pk":4,"intCol":1,"strCol":"string"},  
  {"pk":5,"intCol":1,"strCol":"string"},  
  {"pk":8,"intCol":1,"strCol":"1"},  
  {"pk":9,"intCol":1,"strCol":"12"}  
]
```

이런식으로 값이 들어온다면 ...

## 03 어떻게 쓰는 거예요

Get 요청은 이렇게 해주세요

이전과 다르게 JsonHelper를 이용

```
ApiManager.Instance.GET("test/4", data, delegate (UnityWebRequest request)
{
    Debug.Log("res download text : " + request.downloadHandler.text);

    var TestTableData = JsonHelper.FromJson<TestTable>(request.downloadHandler.text);
    Debug.Log($"animalData = ${TestTableData.Length}");

    for (int i = 0; i < TestTableData.Length; i++)
    {
        Debug.Log($"TestTableData[i].pk = {TestTableData[i].pk}, " +
            $"TestTableData[i].intCol = {TestTableData[i].intCol}, " +
            $"TestTableData[i].strCol = {TestTableData[i].strCol}");
    }
});
```

다음페이지에서  
추가 설명

배열의 내용을 순회하며  
확인할 수 있음



## 03 어떻게 쓰는 거예요

Get 요청은 이렇게 해주세요

이와 같이 받아올 자료의 구조체를  
미리 선언해줘야합니다

```
[System.Serializable]
참조 1개
public class TestTable
{
    public int pk;
    public int intCol;
    public string strCol;
}
```

변수 타입은  
swagger, google sheet, mysql workbench  
중 하나를 참고해주세요



## 03 어떻게 쓰는 거예요

POST 요청은 이렇게 해주세요

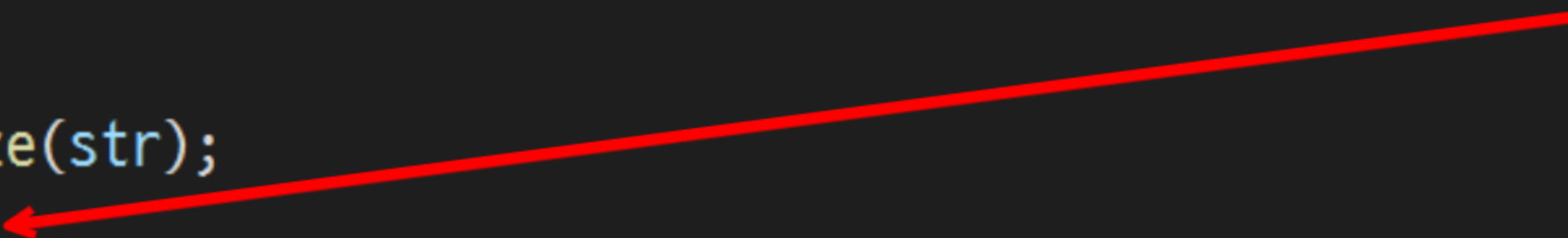
```
var str = new Dictionary<string, object>();  
var res = new Dictionary<string, object>();  
str.Add("strCol", "unity");  
str.Add("intCol", 1);  
  
var data = Json.Serialize(str);  
  
ApiManager.Instance.POST("test/5", data, delegate (UnityWebRequest request)  
{  
    Debug.Log("res download text : " + request.downloadHandler.text);  
});
```

다른점은  
GET ->  
POST ->  
PUT

## 03 어떻게 쓰는 거예요

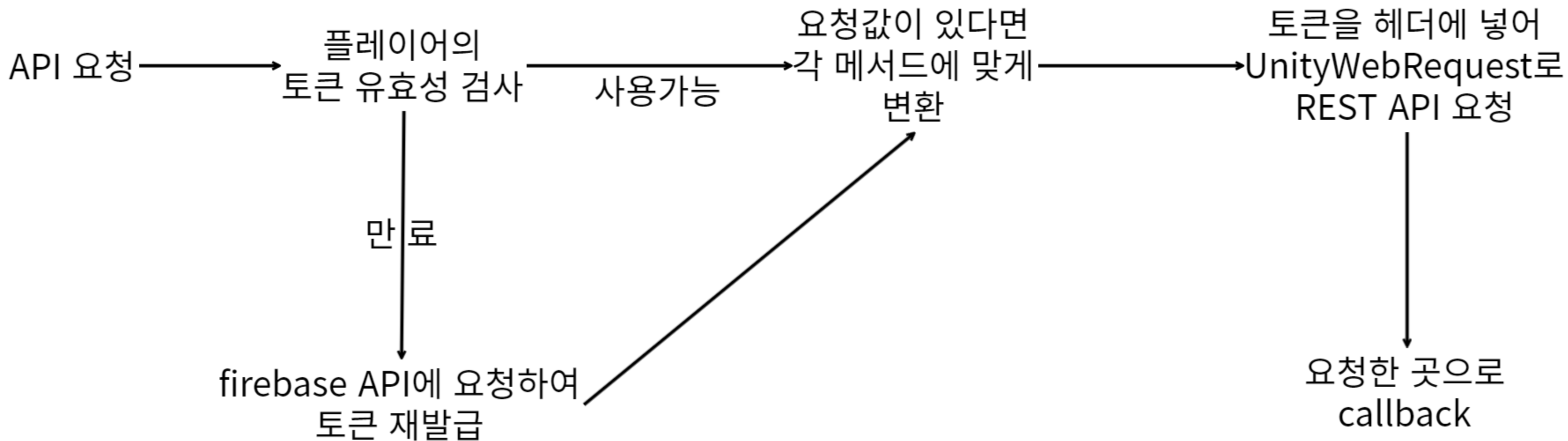
PUT 요청은 이렇게 해주세요

```
var str = new Dictionary<string, object>();  
var res = new Dictionary<string, object>();  
str.Add("hair", 4);  
str.Add("cap", 4);  
  
var data = Json.Serialize(str);  
  
ApiManager.Instance.PUT("test/6", data, delegate (UnityWebRequest request)  
{  
    Debug.Log("res download text : " + request.downloadHandler.text);  
});
```



다른점은  
GET ->  
POST ->  
PUT

## 04 작동원리



# Q&A

궁금한건 언제든지  
안 물어보셔도 됩니다

·  
·  
·

수고하셨습니다