

HttpRequestGet.cs

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.Networking;
5
6 public class HttpRequestGet : MonoBehaviour
7 {
8     [SerializeField] private string targetIP;
9     [SerializeField] private string pathName;
10    [SerializeField] private string queryName;
11    [SerializeField] private string queryNumber;
12
13    // Start is called before the first frame update
14    void Start()
15    {
16        //非同期処理で実施
17        StartCoroutine(GetRequest("http://" + targetIP + "/" + pathName + "?" + queryName + "=" + queryNumber));
18    }
19
20    // Update is called once per frame
21    void Update()
22    {
23    }
24
25    IEnumerator GetRequest(string uri)
26    {
27        //データ送信
28        UnityWebRequest request = UnityWebRequest.Get(uri);
29
30        //返信待ち
31        yield return request.SendWebRequest();
32
33        //エラー判定
34        if (request.result == UnityWebRequest.Result.ConnectionError ||
35            request.result == UnityWebRequest.Result.ProtocolError)
36        {
37            Debug.LogError(request.error);
38        }
39        else
40        {
41            Debug.Log(request.downloadHandler.text);
42        }
43    }
44 }
45
46
```

HttpRequestPost.cs

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.Networking;
5  using System.Text;
6
7  public class HttpRequestPost : MonoBehaviour
8  {
9      [SerializeField] private string targetIP;
10     [SerializeField] private string pathName;
11     [SerializeField] private int val1;
12     [SerializeField] private int val2;
13
14     // Start is called before the first frame update
15     void Start()
16     {
17         //非同期処理で実施
18         StartCoroutine(PostRequest("http://" + targetIP + "/" + pathName, val1, val2));
19     }
20
21     // Update is called once per frame
22     void Update()
23     {
24
25     }
26
27     IEnumerator PostRequest(string uri, int val1, int val2)
28     {
29         //JSONデータの作成
30         string jsonData = JsonUtility.ToJson(new PostData { val1 = val1, val2 = val2 });
31         byte[] postData = Encoding.UTF8.GetBytes(jsonData);
32
33         //POSTで送るよー、って指定
34         UnityWebRequest request = new UnityWebRequest(uri, "POST");
35         request.uploadHandler = new UploadHandlerRaw(postData);
36
37         request.downloadHandler = new DownloadHandlerBuffer();
38         request.SetRequestHeader("Content-Type", "application/json");
39
40         //返信待ち
41         yield return request.SendWebRequest();
42
43         //エラー判定
44         if (request.result == UnityWebRequest.Result.ConnectionError ||
45             request.result == UnityWebRequest.Result.ProtocolError)
46         {
47             Debug.LogError(request.error);
48         }
49         else
50         {
51
```

```
52         Debug.Log(request.downloadHandler.text);
53     }
54 }
55 }
56
57 //JSONデータ作成用のクラス
58 [System.Serializable]
59 public class PostData
60 {
61     public int val1;
62     public int val2;
63 }
64
```

HttpRequestPost.cs

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.Networking;
5  using System.Text;
6
7  public class HttpRequestPost : MonoBehaviour
8  {
9      [SerializeField] private string targetIP;
10     [SerializeField] private string pathName;
11     [SerializeField] private int val1;
12     [SerializeField] private int val2;
13
14     // Start is called before the first frame update
15     void Start()
16     {
17         //非同期処理で実施
18         //StartCoroutine(PostRequest("http://" + targetIP + "/" + pathName, val1, val2));
19         StartCoroutine(PostRequest("http://" + targetIP + "/api2"));
20     }
21
22     // Update is called once per frame
23     void Update()
24     {
25
26     }
27
28
29     //IEnumerator PostRequest(string uri, int val1, int val2)
30     IEnumerator PostRequest(string uri)
31     {
32         //JSONデータの作成
33         //string jsonData = JsonUtility.ToJson(new PostData { val1 = val1, val2 = val2 });
34         //byte[] postData = Encoding.UTF8.GetBytes(jsonData);
35         RequestBody requestBody = new RequestBody();
36         requestBody.user = new UserInfo { name = "Alice", age = 25 };
37         requestBody.score = 100;
38         string json = JsonUtility.ToJson(requestBody);
39         byte[] postData = Encoding.UTF8.GetBytes(json);
40
41         //POSTで送るよー、って指定
42         UnityWebRequest request = new UnityWebRequest(uri, "POST");
43         request.uploadHandler = new UploadHandlerRaw(postData);
44
45         request.downloadHandler = new DownloadHandlerBuffer();
46         request.SetRequestHeader("Content-Type", "application/json");
47
48         //返信待ち
49         yield return request.SendWebRequest();
50
51         //エラー判定
```

```
52     if (request.result == UnityWebRequest.Result.ConnectionError ||
53         request.result == UnityWebRequest.Result.ProtocolError)
54     {
55         Debug.LogError(request.error);
56     }
57     else
58     {
59         Debug.Log(request.downloadHandler.text);
60     }
61 }
62 }
63
64 //JSONデータ作成用のクラス
65 [System.Serializable]
66 public class PostData
67 {
68     public int val1;
69     public int val2;
70 }
71
72 //JSONデータ作成用のクラス
73 [System.Serializable]
74 public class RequestBody
75 {
76     public UserInfo user;
77     public int score;
78 }
79
80 //JSONデータ作成用のクラス
81 [System.Serializable]
82 public class UserInfo
83 {
84     public string name;
85     public int age;
86 }
87
```