

HttpServiceRequest

Senkron istekte bulunur. Servisin güvenlik prosedürüne göre aşağıdaki gibi örneklenir.

Örn.

```
// Not Authentication.
HttpServiceRequest httpRequest = new HttpServiceRequest();

// Basic Authentication.
HttpServiceRequest httpRequest = new HttpServiceRequest("userName", "password");

// Token Based Authentication.
HttpServiceRequest httpRequest = new HttpServiceRequest("token");

// Token Based Authentication.
HttpServiceRequest httpRequest = new HttpServiceRequest("userName", "password", "tokenUrl");
```

1. httpRequest.GET(string requestUrl, string contentType = "application/json")
Gönderilen Url adresine GET metoduyla istekte bulunur.
Geriye json formatında String Döner.
2. httpRequest.GET<T>(string requestUrl, string contentType = "application/json")
Gönderilen Url adresine GET metoduyla istekte bulunur.
Geriye istenilen obje türünde veri döndürür.
Newtonsoft.Json.dll dosyası referans edilmezse çalışmaz.
3. httpRequest.POST(string requestUrl, object responseObject, string contentType = "application/json")
Gönderilen Url adresine gönderilen obje ile POST metoduyla istekte bulunur.
Geriye json formatında String Döner.
4. httpRequest.POST(string requestUrl, string responseObject, string contentType = "application/json")
Gönderilen Url adresine gönderilen string ile POST metoduyla istekte bulunur.
Geriye json formatında String Döner.
5. httpRequest.POST<T>(string requestUrl, object responseObject, string contentType = "application/json")
Gönderilen Url adresine gönderilen obje ile POST metoduyla istekte bulunur.
Geriye istenilen obje türünde veri döndürür.
Newtonsoft.Json.dll dosyası referans edilmezse çalışmaz.
6. httpRequest.POST<T>(string requestUrl, string responseObject, string contentType = "application/json")
Gönderilen Url adresine gönderilen string ile POST metoduyla istekte bulunur.
Geriye istenilen obje türünde veri döndürür.
Newtonsoft.Json.dll dosyası referans edilmezse çalışmaz.

7. `httpClient.put(string requestUrl, object responseObject, string contentType = "application/json")`
Gönderilen Url adresine gönderilen obje ile PUT metoduyla istekte bulunur.
Geriye json formatında String döner.
8. `httpClient.put(string requestUrl, string responseObject, string contentType = "application/json")`
Gönderilen Url adresine gönderilen string ile PUT metoduyla istekte bulunur.
Geriye json formatında String döner.
9. `httpClient.put<T>(string requestUrl, object responseObject, string contentType = "application/json")`
Gönderilen Url adresine gönderilen obje ile PUT metoduyla istekte bulunur.
Geriye istenilen obje türünde veri döndürür.
[Newtonsoft.Json.dll dosyası referans edilmezse çalışmaz.](#)
10. `httpClient.put<T>(string requestUrl, string responseObject, string contentType = "application/json")`
Gönderilen Url adresine gönderilen string ile PUT metoduyla istekte bulunur.
Geriye istenilen obje türünde veri döndürür.
[Newtonsoft.Json.dll dosyası referans edilmezse çalışmaz.](#)
11. `httpClient.delete(string requestUrl, string contentType = "application/json")`
Gönderilen Url adresine DELETE metoduyla istekte bulunur.
Geriye json formatında String döner.
12. `httpClient.delete<T>(string requestUrl, string contentType = "application/json")`
Gönderilen Url adresine DELETE metoduyla istekte bulunur.
Geriye istenilen obje türünde veri döndürür.
[Newtonsoft.Json.dll dosyası referans edilmezse çalışmaz.](#)

```
public class Request
{
    public string Method { get; set; }
    public string Content_Type { get; set; }
    public string Authorization { get; set; }
    public string[] Headers { get; set; }
    public string Body { get; set; }
}
```

13. `httpClient.request(string requestUrl, Request obj)`
Gönderilen Url adresine Gönderilen "Request" objesindeki verilere göre istekte bulunur.
[Authentication 'ın Manuel girilmesi gerekir örneklemede tanımlanan Authentication bilgilerini kullanmaz !!!!](#)
Geriye json formatında String döner.
14. `httpClient.request<T>(string requestUrl, Request obj)`
Gönderilen Url adresine Gönderilen "Request" objesindeki verilere göre istekte bulunur.
[Authentication 'ın Manuel girilmesi gerekir örneklemede tanımlanan Authentication bilgilerini kullanmaz !!!!](#)
Geriye istenilen obje türünde veri döndürür.

Newtonsoft.Json.dll dosyası referans edilmezse çalışmaz.

HttpServiceAsyncRequest

Asenkron istekte bulunur. Servisin güvenlik prosedürüne göre aşağıdaki gibi örneklenir. Çağırıldığı metod "async" olmalıdır ve "await" anahtar kelimesiyle metodlar çağırılmalıdır.

Örn.

```
// Not Authentication.
HttpServiceAsyncRequest httpServiceAsyncRequest = new HttpServiceAsyncRequest();

// Basic Authentication.
HttpServiceAsyncRequest httpServiceAsyncRequest = new
HttpServiceAsyncRequest("userName", "password");

// Token Based Authentication.
HttpServiceAsyncRequest httpServiceAsyncRequest = new HttpServiceAsyncRequest("token");

// Token Based Authentication.
HttpServiceAsyncRequest httpServiceAsyncRequest = new
HttpServiceAsyncRequest("userName", "password", "tokenUrl");
```

1. `await httpServiceAsyncRequest.GET(string requestUrl, string contentType = "application/json")`
Gönderilen Url adresine GET metoduyla istekte bulunur.
Geriye json formatında String Döner.
2. `await httpServiceAsyncRequest.GET<T>(string requestUrl, string contentType = "application/json")`
Gönderilen Url adresine GET metoduyla istekte bulunur.
Geriye istenilen obje türünde veri döndürür.
Newtonsoft.Json.dll dosyası referans edilmezse çalışmaz.
3. `await httpServiceAsyncRequest.POST(string requestUrl, object responseObject, string contentType = "application/json")`
Gönderilen Url adresine gönderilen obje ile POST metoduyla istekte bulunur.
Geriye json formatında String Döner.
4. `await httpServiceAsyncRequest.POST(string requestUrl, string responseObject, string contentType = "application/json")`
Gönderilen Url adresine gönderilen string ile POST metoduyla istekte bulunur.
Geriye json formatında String Döner.
5. `await httpServiceAsyncRequest.POST<T>(string requestUrl, object responseObject, string contentType = "application/json")`
Gönderilen Url adresine gönderilen obje ile POST metoduyla istekte bulunur.
Geriye istenilen obje türünde veri döndürür.
Newtonsoft.Json.dll dosyası referans edilmezse çalışmaz.
6. `await httpServiceAsyncRequest.POST<T>(string requestUrl, string responseObject, string contentType = "application/json")`
Gönderilen Url adresine gönderilen string ile POST metoduyla istekte bulunur.
Geriye istenilen obje türünde veri döndürür.
Newtonsoft.Json.dll dosyası referans edilmezse çalışmaz.

7. `await httpServiceAsyncRequest.PUT(string requestUrl, object responseObject, string contentType = "application/json")`
Gönderilen Url adresine gönderilen obje ile PUT metoduyla istekte bulunur.
Geriye json formatında String Döner.
8. `await httpServiceAsyncRequest.PUT(string requestUrl, string responseObject, string contentType = "application/json")`
Gönderilen Url adresine gönderilen string ile PUT metoduyla istekte bulunur.
Geriye json formatında String Döner.
9. `await httpServiceAsyncRequest.PUT<T>(string requestUrl, object responseObject, string contentType = "application/json")`
Gönderilen Url adresine gönderilen obje ile PUT metoduyla istekte bulunur.
Geriye istenilen obje türünde veri döndürür.
[Newtonsoft.Json.dll dosyası referans edilmezse çalışmaz.](#)
10. `await httpServiceAsyncRequest.PUT<T>(string requestUrl, string responseObject, string contentType = "application/json")`
Gönderilen Url adresine gönderilen string ile PUT metoduyla istekte bulunur.
Geriye istenilen obje türünde veri döndürür.
[Newtonsoft.Json.dll dosyası referans edilmezse çalışmaz.](#)
11. `await httpServiceAsyncRequest.DELETE(string requestUrl, string contentType = "application/json")`
Gönderilen Url adresine DELETE metoduyla istekte bulunur.
Geriye json formatında String Döner.
12. `await httpServiceAsyncRequest.DELETE<T>(string requestUrl, string contentType = "application/json")`
Gönderilen Url adresine DELETE metoduyla istekte bulunur.
Geriye istenilen obje türünde veri döndürür.
[Newtonsoft.Json.dll dosyası referans edilmezse çalışmaz.](#)

```
public class Request
{
    public string Method { get; set; }
    public string Content_Type { get; set; }
    public string Authorization { get; set; }
    public string[] Headers { get; set; }
    public string Body { get; set; }
}
```

13. `await httpServiceAsyncRequest.REQUEST(string requestUrl, Request obj)`
Gönderilen Url adresine Gönderilen "Request" objesindeki verilere göre istekte bulunur.
[Authentication 'ın Manuel girilmesi gerekir örneklemede tanımlanan Authentication bilgilerini kullanmaz !!!!](#)
Geriye json formatında String Döner.
14. `await httpServiceAsyncRequest.REQUEST<T>(string requestUrl, Request obj)`
Gönderilen Url adresine Gönderilen "Request" objesindeki verilere göre istekte bulunur.
[Authentication 'ın Manuel girilmesi gerekir örneklemede tanımlanan Authentication bilgilerini kullanmaz !!!!](#)
Geriye istenilen obje türünde veri döndürür.
[Newtonsoft.Json.dll dosyası referans edilmezse çalışmaz.](#)

Örn:

```
using System;
using System.Collections.Generic;
using ServiceConnect;
using ServiceConnect.Object;

namespace TestServiceControl
{
    class Program
    {
        static void Main(string[] args)
        {
            FillData();           // Default
            FillDataAsync();      // Async
            FillDataBasic();      // Basic Authentication
            FillDataAsyncBasic();  // Basic Authentication Async
            FillDataToken();      // Token Authentication
            FillDataAsyncToken();  // Token Authentication Async
            FillDataTokenManuel(); // Token Authentication Manuel & REQUEST Method
            FillDataAsyncTokenManuel(); // Token Authentication Async Manuel & Async REQUEST Method
        }

        private static void FillData()
        {
            HttpServiceRequest request = new HttpServiceRequest();
            string json = "{ 'Key':Value, 'Key':Value }";
            object response;

            // GET
            response = request.GET("http://192.168.1.79/ExampleApi/api/Products/Gets");

            // POST
            response = request.POST("http://192.168.1.79/ExampleApi/api/Products/Create", json);

            // PUT
            response = request.PUT("http://192.168.1.79/ExampleApi/api/Products/Update", json);

            // DELETE
            response = request.DELETE("http://192.168.1.79/ExampleApi/api/Products/Delete/5");
        }

        private async static void FillDataAsync()
        {
            HttpServiceAsyncRequest asyncRequest = new HttpServiceAsyncRequest();
            string json = "{ 'Key':Value, 'Key':Value }";
            object response;

            // GET
            response = await asyncRequest.GET("http://192.168.1.79/ExampleApi/api/Products/Gets");

            // POST
            response = await asyncRequest.POST("http://192.168.1.79/ExampleApi/api/Products/Create", json);

            // PUT
            response = await asyncRequest.PUT("http://192.168.1.79/ExampleApi/api/Products/Update", json);

            // DELETE
            response = await asyncRequest.DELETE("http://192.168.1.79/ExampleApi/api/Products/Delete/5");
        }
    }
}
```

```

private static void FillDataBasic()
{
    HttpServiceRequest request = new HttpServiceRequest("test", "1234");
    string json = "{ 'Key':Value, 'Key':Value}";
    object response;

    // GET
    response = request.GET("http://192.168.1.79/ExampleApis/api/Products/Gets");

    // POST
    response = request.POST("http://192.168.1.79/ExampleApis/api/Products/Create", json);

    // PUT
    response = request.PUT("http://192.168.1.79/ExampleApis/api/Products/Update", json);

    // DELETE
    response = request.DELETE("http://192.168.1.79/ExampleApis/api/Products/Delete/5");
}

private async static void FillDataAsyncBasic()
{
    HttpServiceAsyncRequest asyncRequest = new HttpServiceAsyncRequest("test", "1234");
    string json = "{ 'Key':Value, 'Key':Value}";
    object response;

    // GET
    response = await asyncRequest.GET("http://192.168.1.79/ExampleApis/api/Products/Gets");

    // POST
    response = await asyncRequest.POST("http://192.168.1.79/ExampleApis/api/Products/Create", json);

    // PUT
    response = await asyncRequest.PUT("http://192.168.1.79/ExampleApis/api/Products/Update", json);

    // DELETE
    response = await asyncRequest.DELETE("http://192.168.1.79/ExampleApis/api/Products/Delete/5");
}

private static void FillDataToken()
{
    HttpServiceRequest request = new
HttpServiceRequest("test", "1234", "http://192.168.1.79/ExampleApiTokenBased/Token");
    string json = "{ 'Key':Value, 'Key':Value}";
    object response;

    // GET
    response = request.GET("http://192.168.1.79/ExampleApiTokenBased/api/Product/Gets");

    // POST
    response = request.POST("http://192.168.1.79/ExampleApiTokenBased/api/Product/Create", json);

    // PUT
    response = request.PUT("http://192.168.1.79/ExampleApiTokenBased/api/Product/Update", json);

    // DELETE
    response = request.DELETE("http://192.168.1.79/ExampleApiTokenBased/api/Product/Delete/5");
}

private static async void FillDataAsyncToken()
{
    HttpServiceAsyncRequest asyncRequest = new HttpServiceAsyncRequest("test",
"1234", "http://192.168.1.79/ExampleApiTokenBased/Token");
    string json = "{ 'Key':Value, 'Key':Value}";
    object response;

    // GET
    response = await asyncRequest.GET("http://192.168.1.79/ExampleApiTokenBased/api/Product/Gets");

    // POST
    response = await asyncRequest.POST("http://192.168.1.79/ExampleApiTokenBased/api/Product/Create", json);

    // PUT
    response = await asyncRequest.PUT("http://192.168.1.79/ExampleApiTokenBased/api/Product/Update", json);

    // DELETE
    response = await asyncRequest.DELETE("http://192.168.1.79/ExampleApiTokenBased/api/Product/Delete/5");
}

```

```

private static void FillDataTokenManuel()
{
    HttpServiceRequest tokenRequest = new HttpServiceRequest();
    Request requestObject = new Request();

    requestObject.Method = "POST";
    requestObject.Content_Type = "application/x-www-form-urlencoded";
    requestObject.Body = "grant_type=password&username=test&password=1234";

    //REQUEST
    string token = tokenRequest.REQUEST("http://192.168.1.79/ExampleApiTokenBased/Token", requestObject);
    token = ConvertJsonToToken(token);

    HttpServiceRequest request = new HttpServiceRequest(token);
    string json = "{ 'Key':Value, 'Key':Value}";
    object response;

    // GET
    response = request.GET("http://192.168.1.79/ExampleApiTokenBased/api/Product/Gets");

    // POST
    response = request.POST("http://192.168.1.79/ExampleApiTokenBased/api/Product/Create", json);

    // PUT
    response = request.PUT("http://192.168.1.79/ExampleApiTokenBased/api/Product/Update", json);

    // DELETE
    response = request.DELETE("http://192.168.1.79/ExampleApiTokenBased/api/Product/Delete/5");
}

private static async void FillDataAsyncTokenManuel()
{
    HttpServiceAsyncRequest tokenRequest = new HttpServiceAsyncRequest();
    Request requestObject = new Request();

    requestObject.Method = "POST";
    requestObject.Content_Type = "application/x-www-form-urlencoded";
    requestObject.Body = "grant_type=password&username=test&password=1234";

    //REQUEST
    string token = await tokenRequest.REQUEST("http://192.168.1.79/ExampleApiTokenBased/Token",
requestObject);
    token = ConvertJsonToToken(token);

    HttpServiceAsyncRequest request = new HttpServiceAsyncRequest(token);
    string json = "{ 'Key':Value, 'Key':Value}";
    object response;

    // GET
    response = await request.GET("http://192.168.1.79/ExampleApiTokenBased/api/Product/Gets");

    // POST
    response = await request.POST("http://192.168.1.79/ExampleApiTokenBased/api/Product/Create", json);

    // PUT
    response = await request.PUT("http://192.168.1.79/ExampleApiTokenBased/api/Product/Update", json);

    // DELETE
    response = await request.DELETE("http://192.168.1.79/ExampleApiTokenBased/api/Product/Delete/5");
}

private static string ConvertJsonToToken(string strJson)
{
    strJson = strJson.Replace('{', ' ').Trim().Replace('}', ' ').Trim().Replace('"', ' ').Trim();
    string[] arrayStrJson = strJson.Split(',');
    string token = String.Empty;
    string tokenType = String.Empty;
    foreach (var item in arrayStrJson)
    {
        if (item.Split(':')[0].Trim() == "access_token")
            token = item.Trim().Split(':')[1];
        if (item.Split(':')[0].Trim() == "token_type")
            tokenType = item.Trim().Split(':')[1];
    }
    return tokenType + " " + token;
}
}
}

```