

## Logistica クラウド営業キロ程距離 WebApi サービス\_利用手順

## 1. WebAPI の動作確認

### 1.1 WebAPI の動作確認手順

1.1.1 WebAPI を呼び出す URL を作成します。

```
https://haruka01.logistica.jp/lgsWebKiro01/api/K/に付けて
?f=発 JIS 市区町村コードまたは発郵便番号
&t=着 JIS 市区町村コードまたは着郵便番号
&k=WebApiKey 値 を指定します。
```

作成した URL をブラウザのアドレスバーに貼り付けて [Enter]  
ブラウザ画面に結果が表示されます。

13102 東京都中央区 27127 大阪市北区

```
https://haruka01.logistica.jp/lgsWebKiro01/api/K/?f=13102&t=27127&k=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
```

```
{"ShuyouKyorikm":576,"SaitanKyorikm":555,"Error":""}
```

主要道路優先距離 Km:576 一般道路優先距離 Km:555  
{"ShuyouKyorikm":576,"SaitanKyorikm":555,"Error":""}

1.1.2 下記の発着 JIS 市区町村コード、郵便番号でも同様の結果となります。

発 JIS 市区町村コード 13102 東京都中央区 着郵便番号 5300005 大阪府大阪市北区中之島

```
https://haruka01.logistica.jp/lgsWebKiro01/api/K/?f=13102&t=5300005&k=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
{"ShuyouKyorikm":576,"SaitanKyorikm":555,"Error":""}
```

発郵便番号 1048404 東京都中央区築地一丁目 1 番 1 号 着郵便番号 大阪府大阪市北区中之島

```
https://haruka01.logistica.jp/lgsWebKiro01/api/K/?f=1048404&t=5300005&k=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
{"ShuyouKyorikm":576,"SaitanKyorikm":555,"Error":""}
```

### 1.2 WebAPI 呼び出しのプログラム作成手順

1.2.1 ブラウザで WebAPI を呼び出す URL を検索して結果を求める処理を自動化するプログラムを作成します。

WebAPI の呼び出しは、Python, Javascript, Java などあらゆる種類のプログラムで処理できます。

1.2.2 ChatGTP 送信「プログラム言語別に WebAPI を呼び出すサンプルコードを作成してください」で WebAPI を呼び出す言語別のサンプルコードは入手できます。

[上記 Chat のリンク](#)

1.2.3 発着の JIS 市区町村コードまたは郵便番号間のキロ程距離計算を自動化して既存の業務アプリへ組み込みます。

次ページ以降では WebAPI の呼び出しのサンプルコードとして C#を取り上げます。

利用手順として

- (1) VisualStudio2022 を用意します。
  - (2) GitHub クローンから複製してプログラムを作成します。
    - ①用意した DLL で処理
    - ②HttpClient で処理
- ①②のプログラムは GitHub クローンから複製して動作確認できます。

## 2. VS2022 のインストール

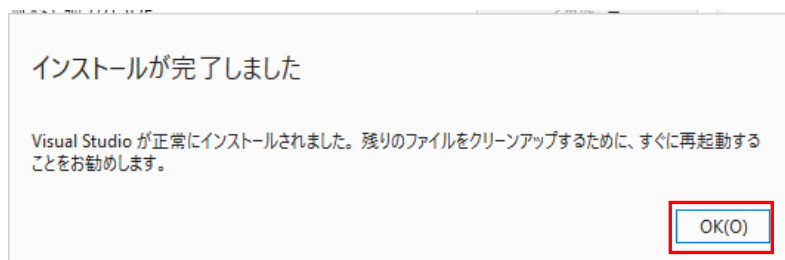
### 2.1 Visual Studio Community からダウンロード

#### 2.1.1 Visual Studio Community

<https://visualstudio.microsoft.com/ja/vs/community/>



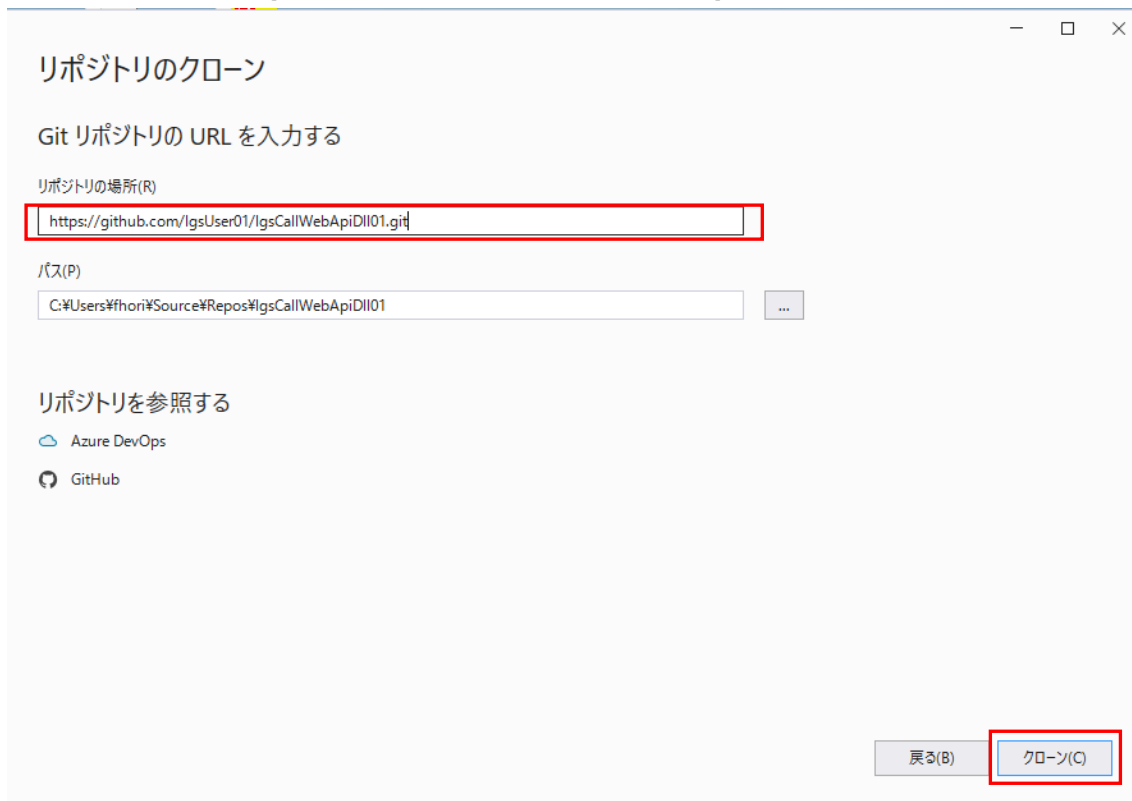
#### 2.1.2 インストールする項目の選択



### 3 WebApi 呼び出し C#サンプル

#### 3.1 WebApi DLL から呼び出し C#サンプル

3.1.1 リポジトリの場所に <https://github.com/lgsUser01/lgsCallWebApiDll01.git> を入力してクローン



3.1.2 Form1.cs で WebApiKey 値と必要ならプロキシ認証の指定をする

```

namespace lgsCallWebApiDll01
{
    3 個の参照
    public partial class Form1 : Form
    {
        ClgsCallWebApi01.ClgsWebKiroTeiKyori01 ClgsWebKiroTei = new ClgsCallWebApi01.ClgsWebKiroTeiKyori01();

        1 個の参照
        public Form1()
        {
            InitializeComponent();
            this.textBoxWebApiUrl.Text = "https://haruka01.logistica.jp/lgsWebKiro01/api/K/";
            this.textBoxWebApiKey.Text = "ここにWebApiKey値をセットする"; // ここにWebApiKey値をセットする

            ClgsWebKiroTei.WebApiUrl = this.textBoxWebApiUrl.Text;
            ClgsWebKiroTei.WebApiKey = this.textBoxWebApiKey.Text;

            // プロキシ認証なし
            ClgsWebKiroTei.ProxyServer = "";
            ClgsWebKiroTei.ProxyUserName = "";
            ClgsWebKiroTei.ProxyPassword = "";

            this.textBox起点.Text = "13102";
            this.textBox終点.Text = "27127";
        }
    }
}

```

3.1.3 デバッグの開始

Form1

WebApi呼び出し

WebApiURL

WebApiKey

起点

終点

主要距離Km

最短距離Km

Error

**3.2 WebApi HttpClient から呼び出し C#サンプル****3.2.1 ChatGPT で「C#で WebAPI を呼び出すサンプルコードを作成してください」を問い合わせた回答です。**

以下は、C#で WebAPI を呼び出すサンプルコードです。この例では、HTTP GET メソッドを使用して API エンドポイントにアクセスし、JSON 応答をシリアル化して.NET オブジェクトにデシリアル化します。

```
using System;
using System.Net.Http;
using System.Net.Http.Headers;
using System.Threading.Tasks;
using Newtonsoft.Json;
public class Program
{
    static HttpClient client = new HttpClient();
    static async Task Main(string[] args)
    {
        await CallWebApi();
    }
    static async Task CallWebApi()
    {
        client.BaseAddress = new Uri("https://api.example.com/");
        client.DefaultRequestHeaders.Accept.Clear();
        client.DefaultRequestHeaders.Accept.Add(new MediaTypeWithQualityHeaderValue("application/json"));

        HttpResponseMessage response = await client.GetAsync("users");
        if (response.IsSuccessStatusCode)
        {
            string jsonString = await response.Content.ReadAsStringAsync();
            User[] users = JsonConvert.DeserializeObject<User[]>(jsonString);

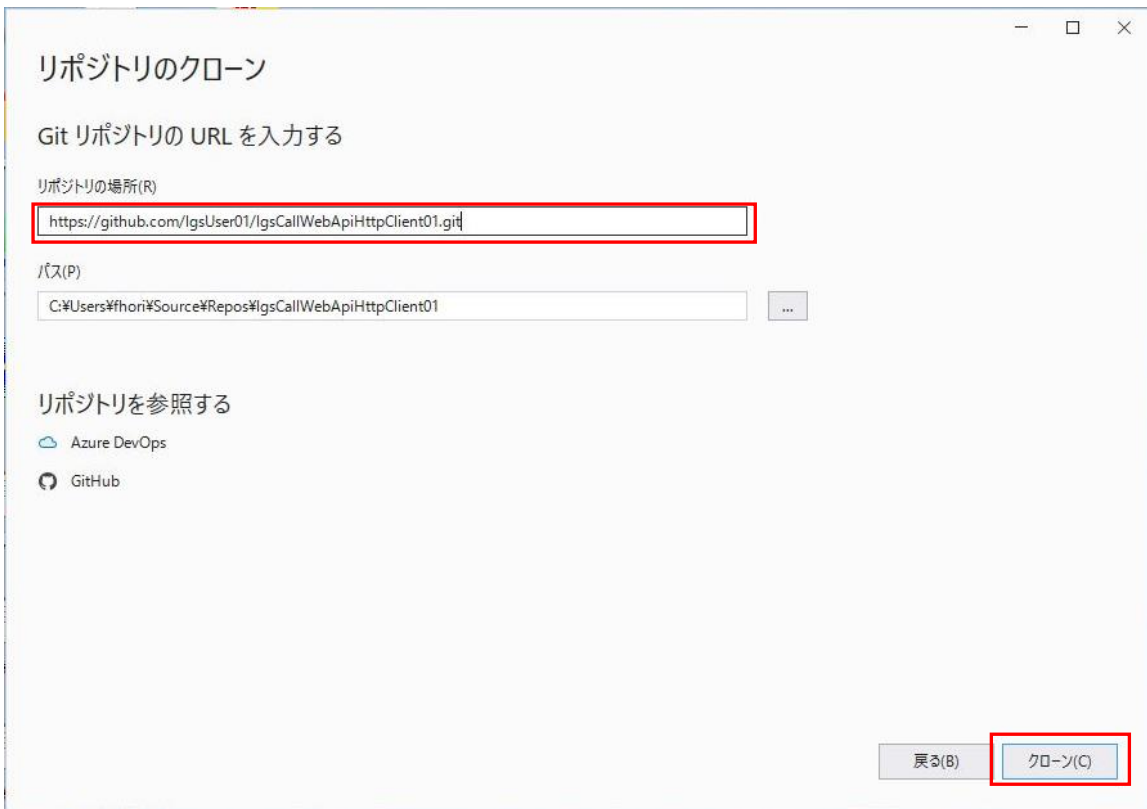
            foreach (User user in users)
            {
                Console.WriteLine($"ID: {user.Id}, Name: {user.Name}");
            }
        }
        else
        {
            Console.WriteLine($"Web API error. Status code: {response.StatusCode}");
        }
    }
}
public class User
{
    public int Id { get; set; }
    public string Name { get; set; }
}
```

この例では、HttpClient クラスを使用して API エンドポイントに HTTP GET リクエストを送信し、レスポンスを取得します。レスポンスが成功した場合は、レスポンス本文を string として取得し、JsonConvert クラスを使用してデシリアライズします。デシリアライズされたデータは、User オブジェクトの配列として取得されます。

注意: リモート API を呼び出す前に、HttpClient を初期化するために必要なセキュリティ/認証/証明書情報を設定する必要があります。また、例外処理も含めて API 呼び出しを実装してください。

7 - 3.2 WebApi HttpClient から呼び出し C#サンプル

3.2.2 リポジトリの場所 <https://github.com/lgsUser01/lgsCallMyWebApiHttpClient01.git> を入力してクローン



3.2.3 Form1.cs で WebApiKey 値と必要ならプロキシ認証の指定をする

```

1 個の参照
public Form1()
{
    InitializeComponent();
    this.textBoxWebApiUrl.Text = "https://haruka01.logistica.jp/lgsWebKiro01/api/K/";
    this.textBoxWebApiKey.Text = "ここにWebApiKey値をセットする"; // ここにWebApiKey値をセットする

    this.textBox起点.Text = "13102";
    this.textBox終点.Text = "27127";
}

1 個の参照
private async void httpClient呼び出しToolStripMenuItem_Click(object sender, EventArgs e) ...

1 個の参照
private async Task<Kyorikm> WebApiKyorikm(string Url)
{
    // 【2】 プロキシ認証の指定
    string strProxyServer = "ProxyServer"; // .Pstring.Empty; // (1)サーバ名
    string strUserName = "ProxyUserName"; // string.Empty; // (2)ユーザ名
    string strPassword = "ProxyPassword"; // string.Empty; // (3)パスワード
}

```

3.2.4 デバッグの開始

Form1

HttpClient呼び出し

WebApiURL

WebApiKey

起点

終点

主要距離Km

最短距離Km

Error