

LogisticaTRUCKServer-Ⅱ 距離計算 WebApi 作成手順

1.VS2022 のインストール

1.1 Visual Studio Community からダウンロード

1.1.1 Visual Studio Community

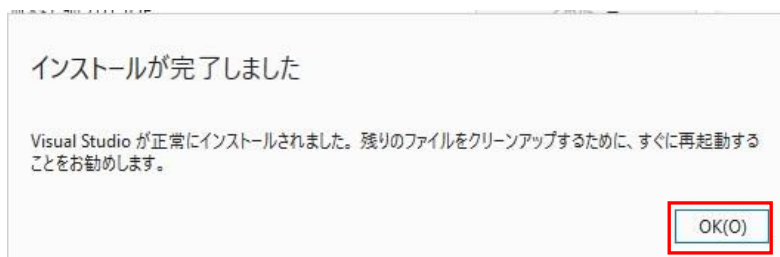
<https://visualstudio.microsoft.com/ja/vs/community>



1.2 インストールする項目の選択



1.3 インストール完了



2. IIS と ASP.NET Core 8 Windows Hosting Bundle のインストール

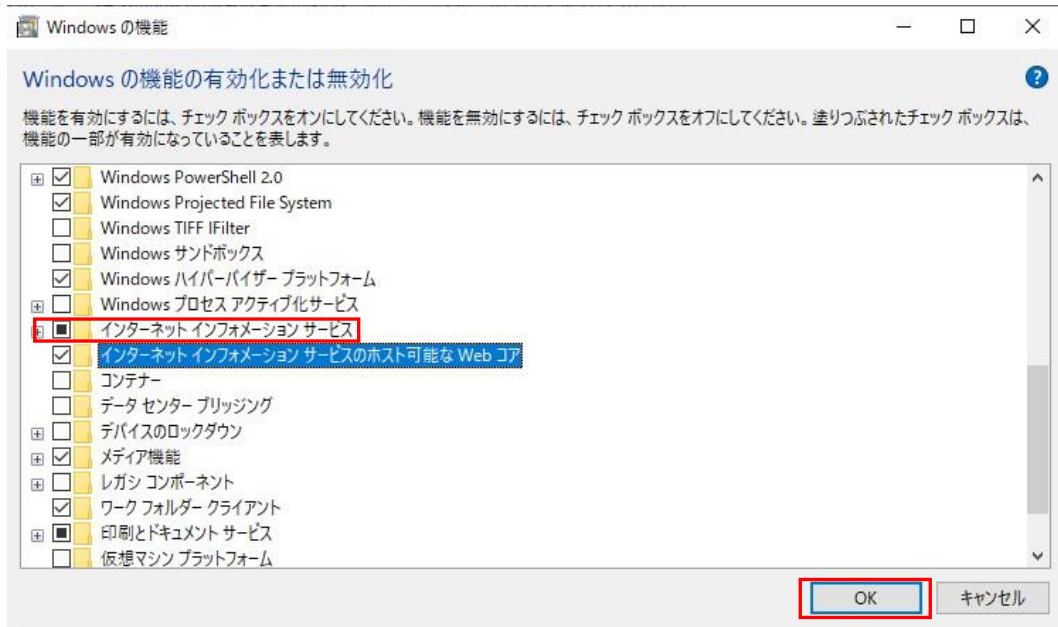
2.1 IIS 機能の有効化

2.1.1. 設定でコントロールパネルを検索 Windows の機能の有効化

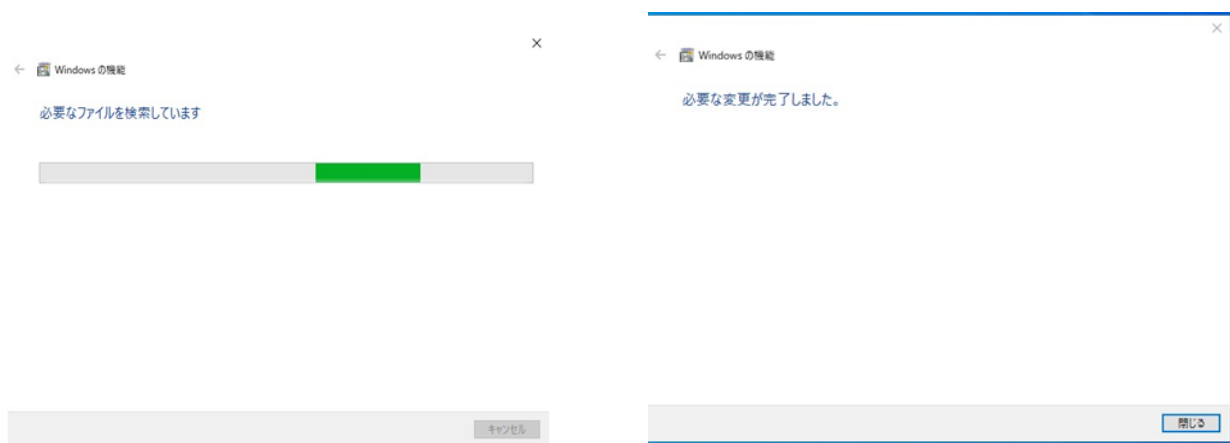
Windows11 では「設定」「Windows 機能の有効化または無効化」を検索します。

Windows Server2022 ではサーバマネージャー → 機能と役割の追加 対象サーバの選択

2.1.2 インターネットインフォメーションサービスのチェックボックスをオンにして OK する。



2.1.3 処理 完了



2.2 ASP.NET Core 8 Windows Hosting Bundle をインストールする

2.2.1 Microsoft .NET8 のダウンロードページ <https://dotnet.microsoft.com/en-us/download/dotnet/8.0> を開きます。
[ASP.NET Core ランタイム 8.0.22] のセクションの [Windows] の [Hosting Bundle] をダブルクリックします。

8.0.22

[リリースノート](#) 最新リリース日 2025年11月11日

アプリのビルド - SDK

SDK 8.0.416

OS	インストーラー	バイナリ
Linux	パッケージマネージャーの手順	Arm32 Arm32 Alpine Arm64 Arm64 Alpine x64 x64 Alpine
macOS	Arm64 x64	Arm64 x64
Windows	x64 x86 Arm64 wingetの手順	x64 x86 Arm64
すべて	dotnet-install scripts	

付加済みランタイム
.NET Runtime 8.0.22
ASP.NET Core ランタイム 8.0.22
.NET デスクトップランタイム 8.0.22

言語サポート
C# 12.0
F# 8.0
Visual Basic 16.9

アプリの実行 - ランタイム

ASP.NET Core ランタイム 8.0.22

ASP.NET Core ランタイムを使用すると、既存の Web/サーバー アプリケーションを実行できます。Windows では、.NET ランタイムと IIS サポートを含むホスティング バンドルをインストールすることをお勧めします。

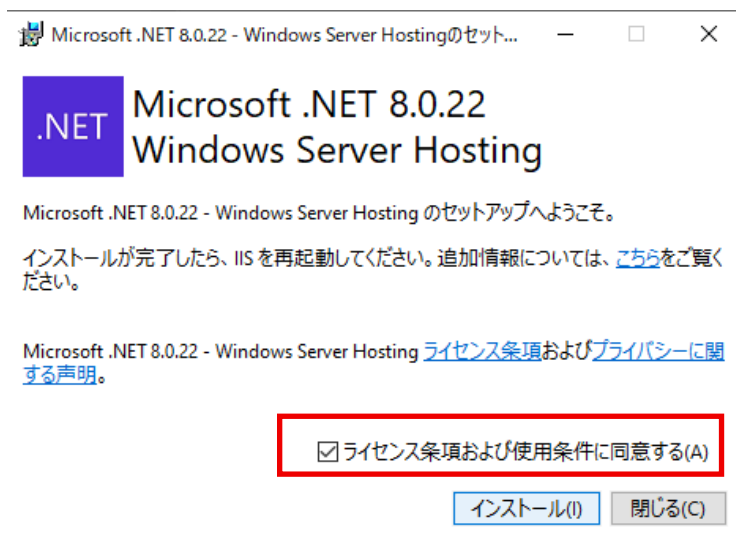
IIS ランタイム サポート (ASP.NET Core モジュール v2)
18.0.25301.22

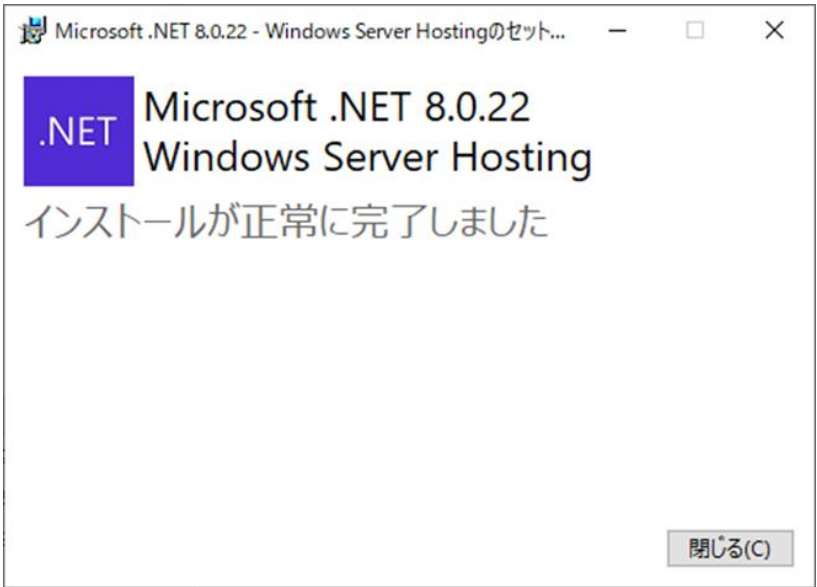
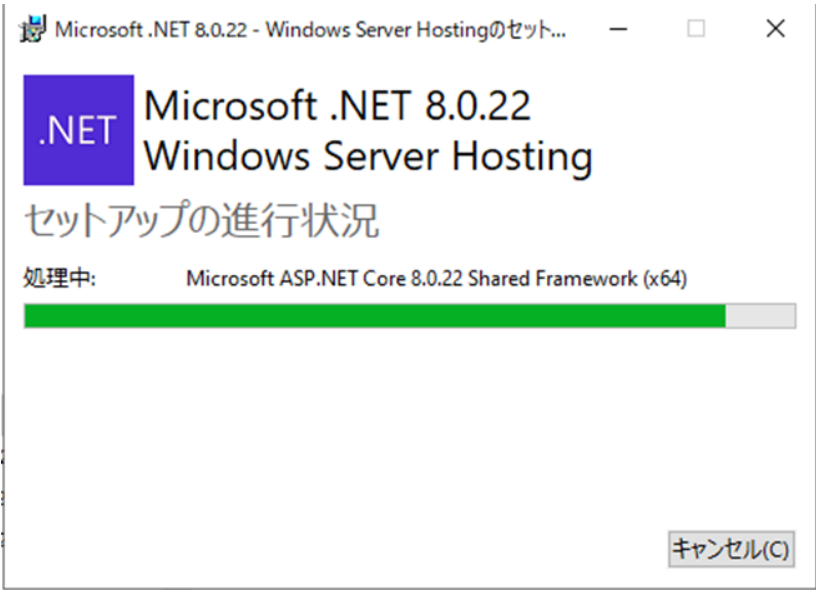
OS	インストーラー	バイナリ
Linux	パッケージマネージャーの手順	Arm32 Arm32 Alpine Arm64 Arm64 Alpine x64 x64 Alpine
macOS		Arm64 x64
Windows	x64 x86 Arm64 Hosting Bundle wingetの手順	x64 x86 Arm64

.NET デスクトップ ランタイム 8.0.22

2.2.2 インストーラファイルがダウンロードされます。

2.2.3 ダウンロードしたインストーラファイル dotnet-hosting-dotnet-hosting-8.0.22-win.exe を実行します。
以降は画面指示に従って ASP.NET Core 8 Windows Hosting Bundle をインストールします。





3. GitHub リポジトリ lgsWebJitsuKyori01 のクローンから WebApi 発行

3.1 VS2022 リポジトリのクローン

3.1.1 リポジトリの場所

<https://github.com/lgsUser01/lgsWebJitsuKyori01.git>



3.1.2 lgsWebJitsuKyori01.sl を開きます。

3.1.3 lgsWebJitsuKyori01 プロジェクトの ¥Contorolers¥KContorollers.cs で 2 か所変更します。

```
68 | | | | TRKSVR2NETX1.ServerComputerName = @"MR8400¥lgstrksvr"; // 【1】 距離計算DBのSQLServerインスタンス
69 | | | | TRKSVR2NETX1.KeisanJouken = 272; // 【2】 デフォルトの計算条件値
70 |
71 |
```

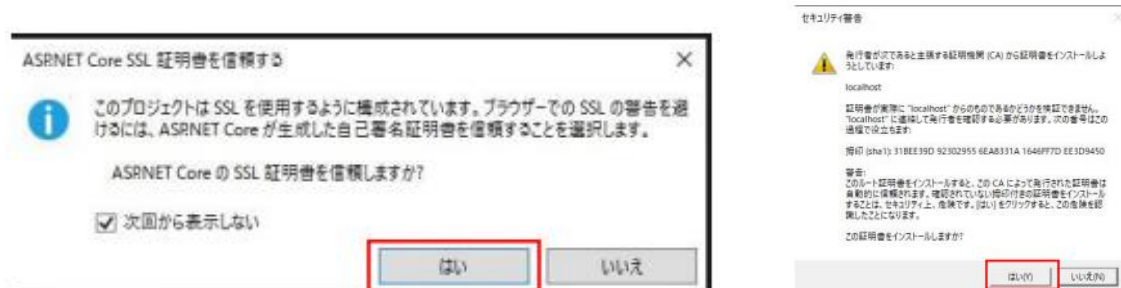
3.1.4 距離計算サーバを起動します。

下記のコマンドプロンプトを実行します。

```
CD ¥lgs¥lgsPAk015S
lgsPAk01M015 Start
```

3.2 デバッグ

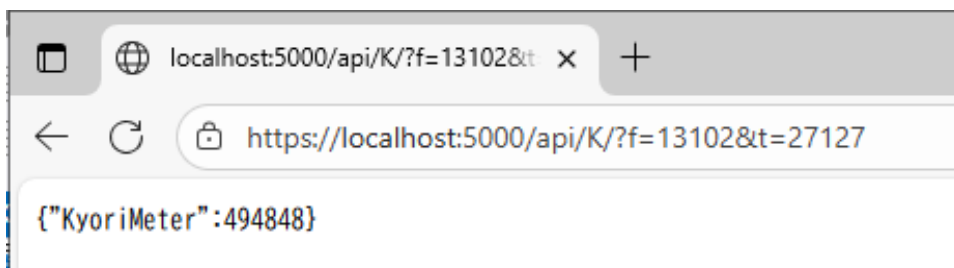
3.2.1 デバッグの開始 証明書・セキュリティ関連のメッセージは「はい」



3.2.2 デバッグ IIS Express Debug

ブラウザから以下を入力します。 (東京都中央区 大阪市北区の実距離計算 494848m)

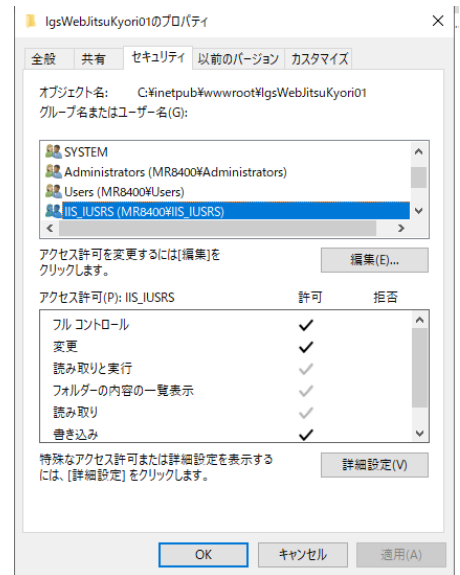
<https://localhost:5000/api/K/?f=13102&t=27127>



3.3 発行・公開

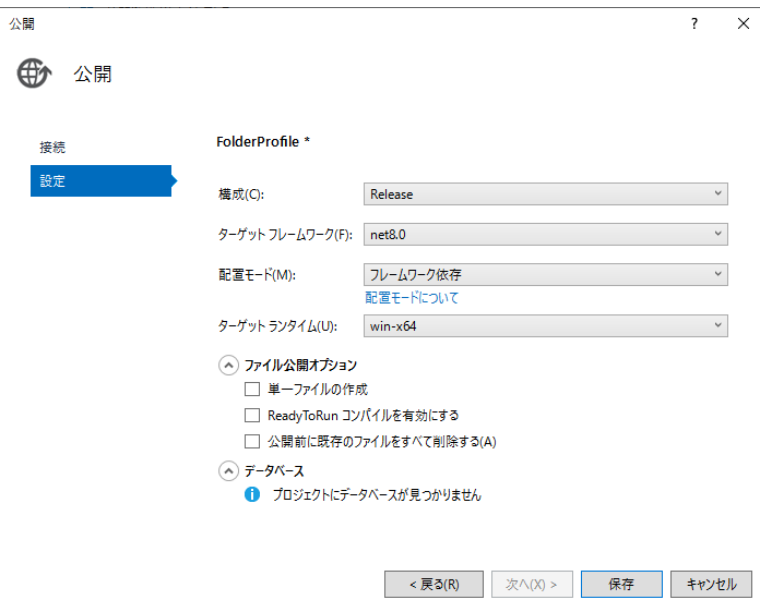
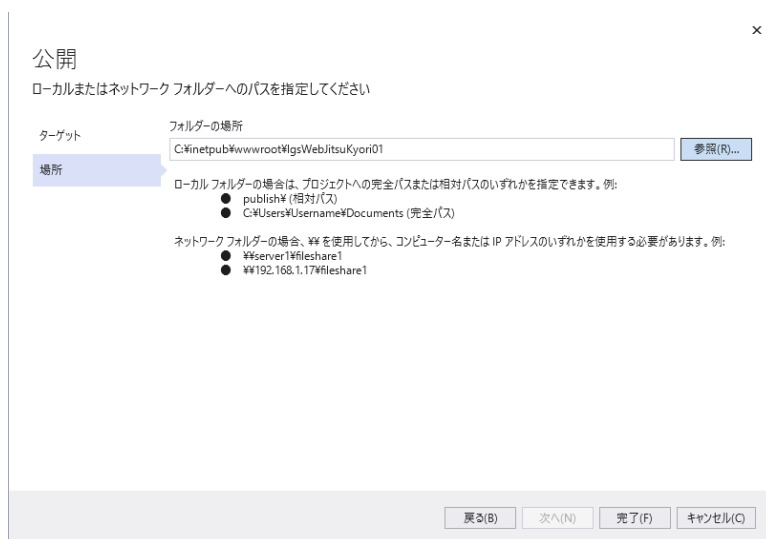
3.3.1 ローカルフォルダ発行

発行先フォルダ C:\inetpub\wwwroot\lgsWebJitsuKyori01 プロパティのセキュリティ Users でフルコントロール書き込み可能に 設定 します。



3.3.2 ビルド → 発行

ターゲット C:\inetpub\wwwroot\lgsWebJitsuKyori01\ で発行・公開します。



4. IIS WebApi のアプリケーション設定

4.1 WebApi のアプリケーション設定

4.1.1 IIS の起動

Windows11 ではスタートボタン→すべてのアプリ→Windows ツール→インターネットインフォメーションサービス (IIS)

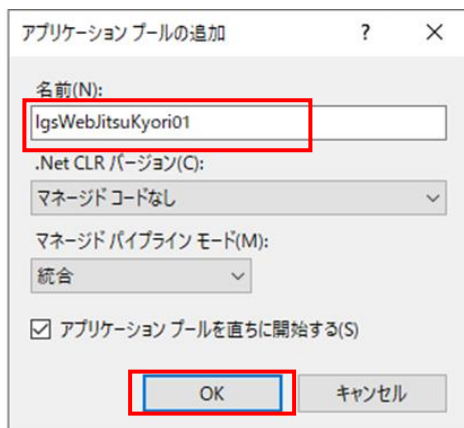
Windows Server2022 ではサーバマネージャー →ツール→インターネットインフォメーションサービス (IIS)

4.1.2 アプリケーションプールの追加

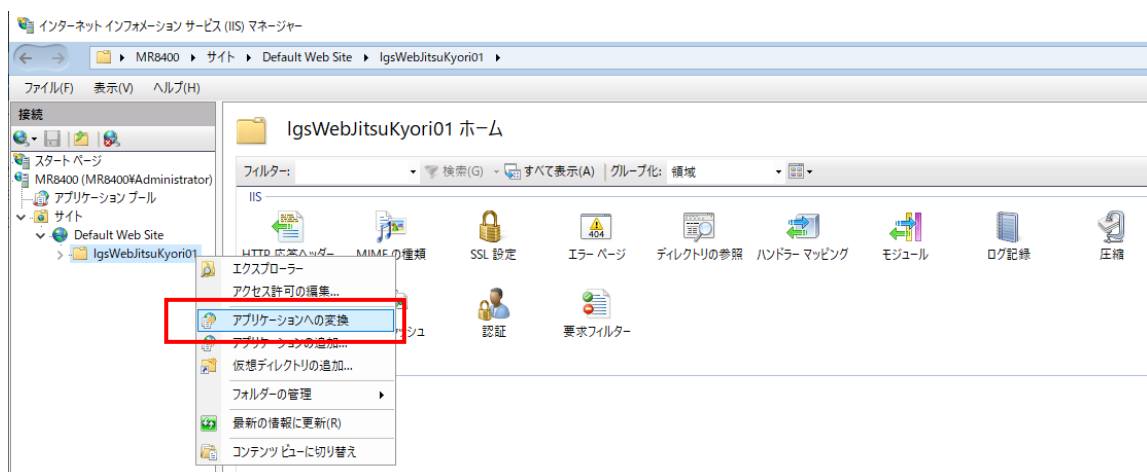


「アプリケーションの追加」画面で

lgsWebJitsuKyori01 マネージコードなし で OK



4.1.3 DefaultWebSite 下のフォルダ lgsWebJitsuKyori01 を右クリックしてメニューからアプリケーションへの変換を選択



4. IIS WebApi のアプリケーション設定

4.1.4 アプリケーションの追加でアプリケーションプールに lgsWebJitsuKyori01 を選択して下図のように設定して OK

アプリケーションの追加

サイト名: Default Web Site
パス: /

エイリアス(A): lgsWebJitsuKyori01 アプリケーション プール(L): DefaultAppPool **選択(E)...**

例: sales

物理パス(P): C:\inetpub\wwwroot\lgsWebJitsuKyori01 ...

パスルー認証
接続(C)... テスト設定(G)...

☐ プリロードの有効化(N)

OK キャンセル

アプリケーション プールの選択

アプリケーション プール(L): **lgsWebJitsuKyori01** ▼

プロパティ:
.Net CLR バージョン: マネージド コードなし
パイプライン モード: 統合

OK キャンセル

アプリケーションの追加

サイト名: Default Web Site
パス: /

エイリアス(A): lgsWebJitsuKyori01 アプリケーション プール(L): lgsWebJitsuKyori01 **選択(E)...**

例: sales

物理パス(P): C:\inetpub\wwwroot\lgsWebJitsuKyori01 ...

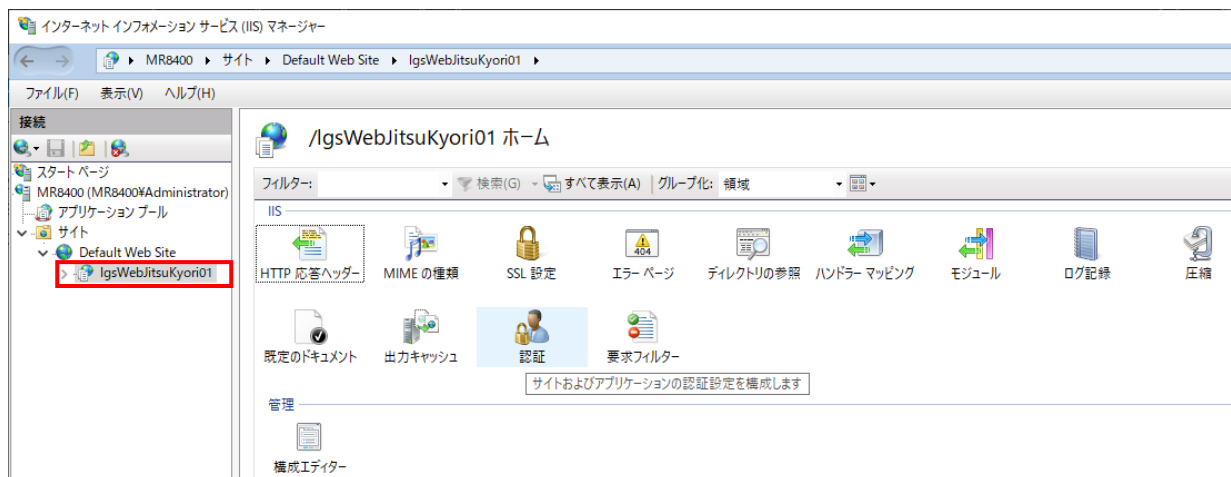
パスルー認証
接続(C)... テスト設定(G)...

☐ プリロードの有効化(N)

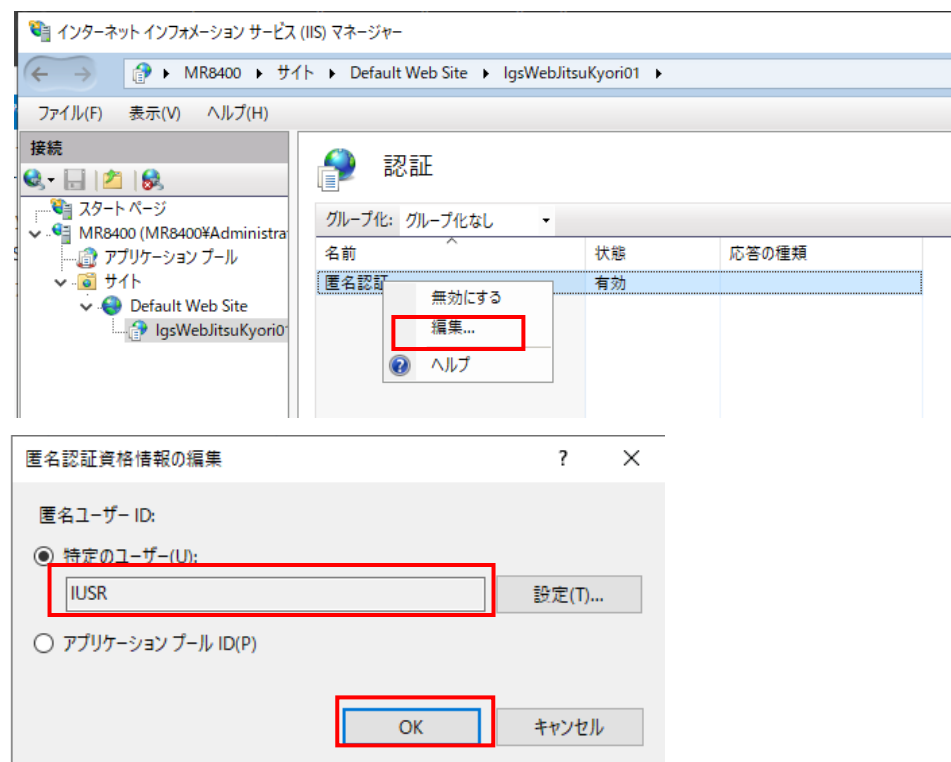
OK キャンセル

4. IIS WebApi のアプリケーション設定

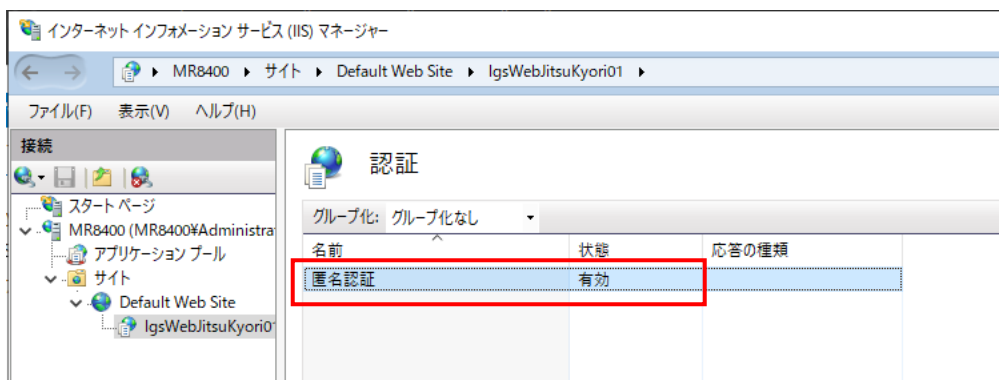
4.1.5 アプリケーション lgsWebJitsuKyori01 の認証を選択する



4.1.6 匿名認証資格情報の編集で特定のユーザーを IUSR と設定して OK



4.1.7 認証で匿名認証が有効を確認

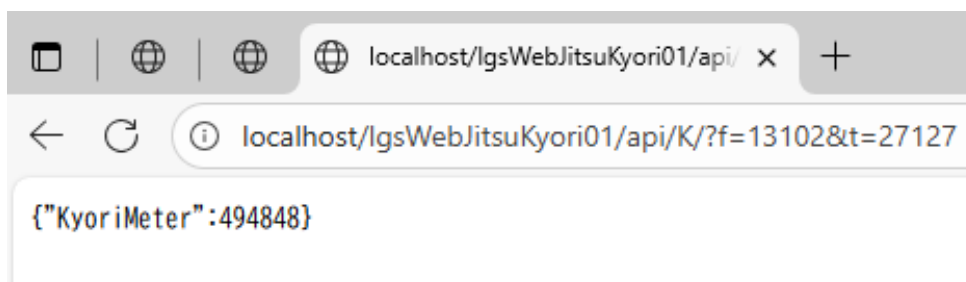


4.2 IIS WebAPI lgsWebJitsuKyori01 の動作確認

- 4.2.1 距離計算サーバを起動します。
下記のコマンドプロンプトを実行します。
CD ¥lgs¥lgsPAk015S
lgsPAk01M015 Start

4.2.2 ブラウザで確認

<http://localhost/lgsWebJitsuKyori01/api/K/?f=13102&t=27127>
{ "KyoriMeter":494848 }

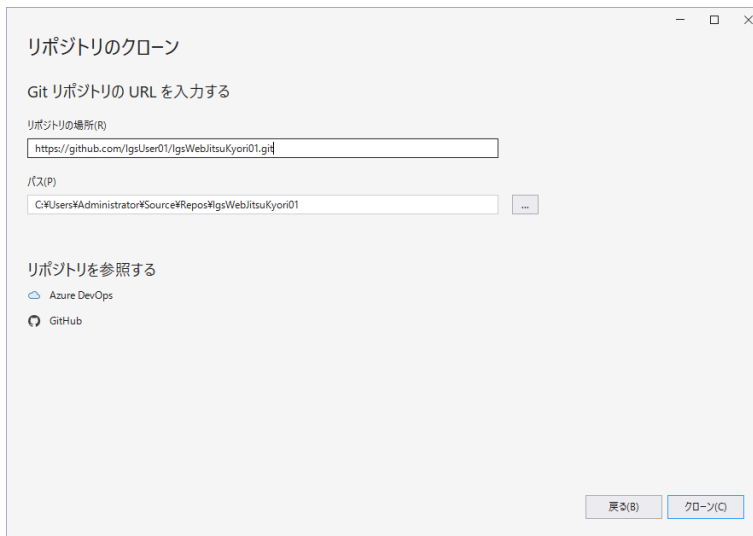


5.WebApi 呼び出し C#サンプル

5.1 VS2022 リポジトリのクローン

5.1.1 リポジトリの場所

<https://github.com/lgsUser01/lgsWebJitsuKyori01.git>



5.1.2 lgsWebJitsuKyori01.sl を開きます。

5.1.3 lgsCallMyWebApiHttpClient01 プロジェクトの Form1.cs で 2 か所変更します。

```
35 | | | | this.textBoxSQL接続先.Text = @"MR8400¥lgstrksvr"; // 【1】 距離計算DB SQLServer インスタンス  
36 | | | | this.textBoxWebApiUrl.Text = "https://localhost:5000/api/K/"; // 【2】 WebAPIのURL  
37 |
```

5.1.4 距離計算サーバを起動します。

コマンドプロンプトから下記コマンドを実行します。

```
CD ¥lgs¥lgsPAk015S  
lgsPAk01M015 Start
```

5.1.5 デバッグ開始

