**NISweb  
ファイル構成**

**目次**

[1. 目的 4](#_Toc507069274)

[2. 構成 5](#_Toc507069275)

[3. View層 7](#_Toc507069276)

[3.1. URLと画面構成ファイルとの関連 7](#_Toc507069277)

[4. Controller層 8](#_Toc507069278)

[4.1. メソッド一覧 8](#_Toc507069279)

[5. Model層 9](#_Toc507069280)

[5.1. クラス一覧 9](#_Toc507069281)

[6. 画面・機能の追加方法 10](#_Toc507069282)

[6.1. URLの組み立て方(ルーティング) 11](#_Toc507069283)

[6.2. ControllerとViewの対応付け 12](#_Toc507069284)

[6.3. 画面追加・変更の手順 13](#_Toc507069285)

[6.4. Modelの追加 14](#_Toc507069286)

[6.5. css/jsの追加・変更方法 15](#_Toc507069287)

[6.6. 補足 ControllerからViewへのデータの渡し方 16](#_Toc507069288)

# 目的

NISwebのファイル構成を示すとともに、画面・機能の追加・変更を行う際の手順を記載する

# 構成

MVCモデルに則りクラスの階層化を図っている。

NIS web ソリューション

View

main.cshtml

\_MainContentView.cshtml

imagelist.cshtml

Controller

retrievedImage.cshtml

\_RetrievedImageContentView.cshtml

apilog.cshtml

\_APILogContentView.cshtml

Model

NISController.cs

FileUtil

FileOperator.cs

GetNewestImage.cs

LogHelper.cs

HttpUtil

HttpConversation.cs

HttpOperator.cs

InitialSetting.cs

階層毎の主要なクラスは以下のとおり

View

|  |  |
| --- | --- |
| 名称 | 内容 |
| main.cshtml | Main画面Vew |
| \_MainContentView.cshtml | Main画面用PartialView |
| imagelist.cshtml | ImageList画面View |
| retrievedImage.cshtml | RetrievedImage画面View |
| \_RetrievedImageContentView.cshtml | RetrievedImage画面用PartialView |
| apilog.cshtml | API Log画面View |
| \_APILogContentView.cshtml | API Log画面用PartialView |

Controller

|  |  |
| --- | --- |
| 名称 | 内容 |
| NISController.cs | View・Modelを紐づけるコントローラー |

Model

|  |  |
| --- | --- |
| 名称 | 内容 |
| FileOperator.cs | ファイル操作全般を扱うクラス |
| GetLatestImage.cs | 画像ファイル取得クラス |
| LogHelper.cs | Logファイル取得クラス |
| HttpOperator.cs | HTTPリクエスト/レスポンスの送受信クラス |
| HttpConversation.cs | HTTPリクエスト送信実行クラス |
| InitialSetting.cs | 起動時処理実行クラス |

その他

|  |  |
| --- | --- |
| 名称 | 内容 |
| Nis.css | 全画面に適用するカスケードスタイル |
| Nis.js | 全画面に適用するjavascript/jquiery |

# View層

## URLと画面構成ファイルとの関連

**Main画面**

URL ： <HTTP://IP>アドレス/NIS/main

　 構成ファイル ： main.cshtml

　　\_MainContentView.cshtml

**ImageList画面**

URL ： <HTTP://IP>アドレス/NIS/imagelist

　 構成ファイル ： imagelist.cshtml

**Retrieved Image画面**

URL ： <HTTP://IP>アドレス/NIS/ retrievedimage

構成ファイル ： retrievedImage.cshtml

\_RetrievedImageContentView.cshtml

**API Log画面**

URL ： <HTTP://IP>アドレス/NIS/ apilog

構成ファイル ： apilog.cshtml

\_APILogContentView.cshtml

なお、PartialViewの作成は、ajaxによる部分更新に対応することを目的としている

# Controller層

ControllerはViewとModelの橋渡しを担い、

全てのメソッド名はViewファイルと同名とし、戻り値はActionResultである

## メソッド一覧

|  |  |
| --- | --- |
| 名称 | 内容 |
| Main | Main画面の表示 |
| MainContent | ・Main画面のPartialViewの表示  ・画面表示時にHTTPサーバへの接続確認  ・Connectボタン押下時にHTTPサーバへ接続確認を行う  ・IsInExperimentボタン押下時にND\_Acquisition動作確認リクエストを送信  ・StageLocationボタン押下時にXYZステージ座標取得リクエストを送信 |
| ApiLog | ApiLog画面の表示 |
| ApiLogContent | ・ApiLog画面のPartialViewの表示  ・HTTPサーバが作成したAPI Logの取得及び内容表示 |
| ImageList | ・ImageList画面の表示  ・HTTPサーバが作成したtifファイル一覧の取得及び内容表示  ・DLボタン押下時に対象ファイルのダウンロード処理 |
| RetrievedImage | RetrievedImage画面の表示 |
| RetrievedImageContent | ・RetrievedImage画面のPartialViewの表示  ・HTTPサーバが作成した最新tifファイルの取得及びプレビュー表示 |
| Exception | Exception画面の表示 |

# Model層

画面状態に依存しない階層

初期設定、ファイル操作、HTTPリクエスト操作を担う

## クラス一覧

|  |  |
| --- | --- |
| 名称 | 内容 |
| FileOperator.cs | ファイル操作全般を扱うクラス  ディレクトリパス・ファイルパスを作成/取得する |
| GetLatestImage.cs | 最新tifファイル取得クラス |
| LogHelper.cs | ローカルからサーバへのLogファイルコピー  Logファイルの内容取得処理を行う |
| HttpOperator.cs | HTTPリクエスト用のURLの作成  HTTPリクエスト送信時/レスポンス受信時の処理を行う |
| HttpConversation.cs | HTTPリクエスト送信実行を行う |
| InitialSetting.cs | 起動時の初期設定、フォルダの初期化を行う |

# 画面の追加方法

ルーティングによりControllerとViewの紐づけがなされ、

URLを指定することにより紐づけされた状態を呼び出すことができる

以降にその紐づけの方法から変更方法を記載する

また、各画面要素に適用するcss/jsの適用方法についても触れていく

## URLの組み立て方(ルーティング)

ViewとController内のActionメソッドが1対1で紐づいているので

呼び出したい画面用のActionメソッドをURLとして指定すれば画面が表示される

例　Main.cshtmlを表示したい場合

NISController->ActionResult Main()を呼び出す

↓

<http://IP>アドレス/NIS/Main

本来、URLにてファイル名を指定すべきところ、

Actionメソッド名の指定のみで画面の呼び出しを実現させているのは

App\_Start->RouteConfig.csである

RouteConfig.cs内一部抜粋

routes.MapRoute(

name: "Default",

url: "{controller}/{action}/{id}",

defaults: new

{

　　controller = "NIS",

　　action = "Main",

　　id = UrlParameter.Optional

}

);

controller = "NIS"の部分をController名と合わせることにより対象のControllerを操作することができ、

URLに対象のActionメソッド名を入力するとRouteConfig.csが解決してくれるという流れとなる

## ControllerとViewの対応付け

ControllerとViewsフォルダ配下にController名と同名のフォルダを作成することにより

ControllerはどのViewが処理対象のファイル群かということを知ることができる

例　NISフォルダ内のViewを対象として処理したい場合

NISControllerを作成した後、ViewsにNISフォルダを作成する

↓

<http://IP>アドレス/NIS/Main

これで、作成したControllerとViewファイル群との対応付けが完了する

## 画面追加・変更の手順

「3.1」, 「3.2」のことから、任意のControllerとViewを作成する流れは以下のようになる

■画面を追加する場合

※1 Controller名称をNISControllerとする

※2 Viewファイル名称をEX.cshtmlとする

1. Views->NISフォルダ内に、EX.cshtmlファイルを作成する

2. NISControllerに、1.で作成したViewファイル名称と同名のActionメソッドを作成する

Actionメソッド例

Public ActionResult EX()

　　　　　{

return View();

}

■Controllerを入れ替える場合

※1 Controller名称をEXControllerとする

※2 Viewファイル名称をchildEX.cshtmlとする

1. ControllersにEXControllerを作成する

2. Viewsフォルダ内にEXフォルダを作成する

3. Views->EXフォルダ内にchildEX.cshtmlファイルを作成する

4. EXControllerに、3.で作成したViewファイル名称と同名のActionメソッドを作成する

Actionメソッド例

Public ActionResult childEX()

　　　　　{

return View();

　　　　　}

# Modelの追加方法

Model層は通常のC#/VB.netの使い方と変わらず、

Modelsフォルダ内であればユーザーの任意の形で配置することができる

本システム及びAsp.net MVCに関する特有の事項は無い

# css/jsの追加・変更方法

App\_Startフォルダ内のBundleConfig.csは、

cssやjquery/javascriptを一つに纏めておくことができる定義ファイルである

このため、新規でcss等を作成した場合はBundleConfig.cs内に追記することにより

Viewファイルへ反映させることができる

BundleConfig.cs 内一部抜粋

bundles.Add(new StyleBundle("~/Content/css").Include(

"~/Content/bootstrap.css",

"~/Content/site.css",

//追加

"~/Content/nis.css",

　 ・

　 ・

# 補足 画面へのデータ出力方法

## ControllerからViewへのデータの渡し方

Viewは、Controllerから直接C#/VB.netの変数でデータを受け取ることは出来ない

そののため、Controller側でViewData若しくはViewBagにデータを格納し、View側に渡す必要がある

※ViewBagはMVC3から使用可能となったコンテナで、

Microsoftが使用を推奨していることからViewBagの使用方法を説明する

■ViewBagの使用方法

* 「ViewBag . key名」でデータの格納・取得が可能
* 初期化の必要は無い
* ・ Dynamic型を返すため、View側では明示的に型指定をする必要がある

使用例

[Controller]

ViewBag. targetString = “Data”;

ViewBag. targetInt = 100;

[View]

　 var stringData = @ViewBag. targetString as string

var intData = @ViewBag. targetInt as int

**参考サイト**

https://msdn.microsoft.com/en-us/library/system.web.mvc.controllerbase.viewbag(v=vs.118).aspx