



SpreadJS の ASP.NET MVC での使い方

グレープシティ株式会社

2016年8月改訂

GrapeCity[®]

目次

はじめに.....	2
ASP.NET MVC プロジェクトの新規作成.....	3
JavaScript の参照と SpreadJS の配置.....	5
データベースとモデルの作成.....	8
データベースの作成	8
モデルの作成	10
コントローラーの作成	15
SpreadJS を使用した CRUD の実装.....	17
データ一覧の表示	17
[作成] ボタン押下時の処理	18
[登録/更新] ボタン押下時の処理	19
[削除] ボタン押下時の処理	21
まとめ.....	22

はじめに

SpreadJS は Web アプリケーションにおけるフロントエンドに特化した HTML5 スプレッドシートコンポーネントです。html 上に配置して使用できるだけでなく、ASP.NET MVC と組み合わせて使用することでサーバーサイド処理を含んだ Web アプリケーションを作成することが可能です。本チュートリアルでは、フロントエンドに SpreadJS、サーバーサイドは ASP.NET MVC (C#) を用い Visual Studio での操作方法を含む Web アプリケーションの開発手順を解説していきます。なお、本チュートリアルでは 2016 年 7 月に公開した SpreadJS SP1 (9.20161.0) を使用しています。

環境

- Visual Studio 2015
- .NET Framework 4.6

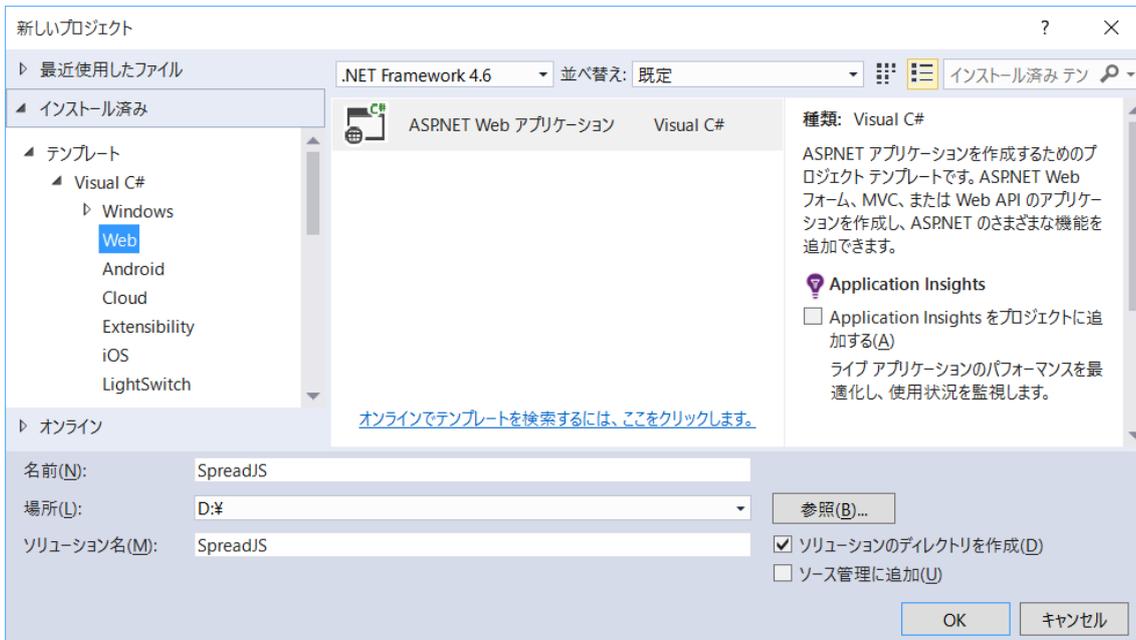
アプリケーションの完成イメージ

	名前	カナ	生年月日	性別	血液型	部署	入社日	メールアドレス
1	亀甲 滋万	トリコ シヅマ	1950-02-04T00:00:00	男	A	人事部	1972-04-01T00:00:00	sigema_kikou@abc.co.jp
2	寒田 希世	カンダ キヨ	1959-06-28T00:00:00	女	B	人事部	1981-04-01T00:00:00	kiyo_kanda@bbb.or.jp
3	小和精 澄	オセキヨ	1969-03-06T00:00:00	男	B	人事部	1991-04-01T00:00:00	kiyo_owase@aaa.co.jp
4	宇夫 早余子ウブサヨコ		1976-07-28T00:00:00	女	O	人事部	1998-04-01T00:00:00	sayoko_ubu@bbb.or.jp
5	宇田津 聖智ウケツキヨモ		1965-09-04T00:00:00	男	A	営業部	1987-04-01T00:00:00	kiyotomo_udatu@abc.co.jp
6	茨城 昭見	イハラク ショウジ	1963-04-28T00:00:00	男	O	営業部	1985-04-01T00:00:00	shouzi_ibaraki@xyz.ne.jp
7	石ヶ休 権茄	イシガキウシイナ	1953-02-21T00:00:00	男	O	営業部	1975-04-01T00:00:00	siina_isigagyu@abc.co.jp
8	赤司 恵治郎	アカサカケイジロウ	1968-08-02T00:00:00	男	O	経理部	1990-04-01T00:00:00	keizirou_akatakasa@abc.co.jp
9	小橋 仰一	オハシキョウイチ	1972-03-02T00:00:00	男	B	経理部	1994-04-01T00:00:00	gyouiti_ohasi@abc.co.jp
10	一重 公大	イチジュウコウダイ	1964-04-19T00:00:00	男	B	経理部	1986-04-01T00:00:00	koudai_itizyuu@xyz.ne.jp
11	稲並 勝五郎	イナミシヨウゴロウ	1962-02-18T00:00:00	男	A	営業部	1984-04-01T00:00:00	shougorou_inami@bbb.or.jp
12	頼原 紀代一	エイハラキヨカズ	1965-02-13T00:00:00	男	O	営業部	1987-04-01T00:00:00	kiyokazu_eihara@bbb.or.jp

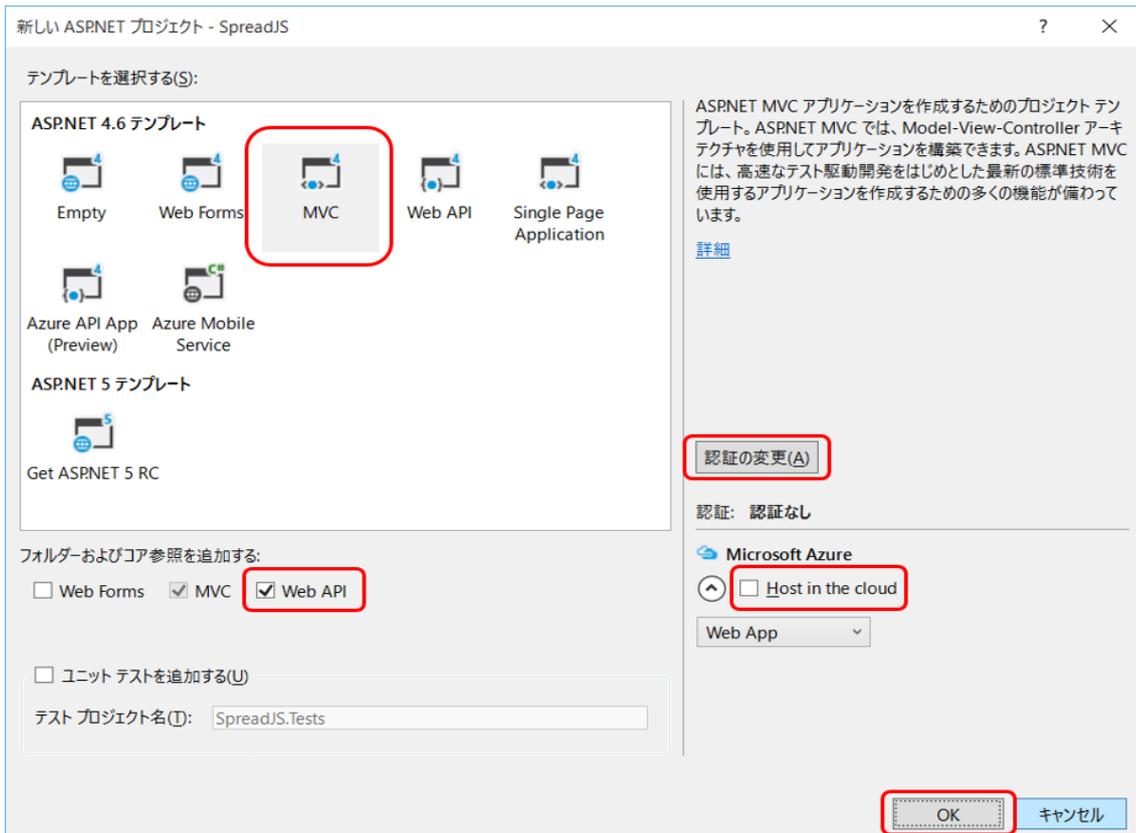
ASP.NET MVC プロジェクトの新規作成

以下の手順で ASP.NET MVC プロジェクトを新規に作成します。

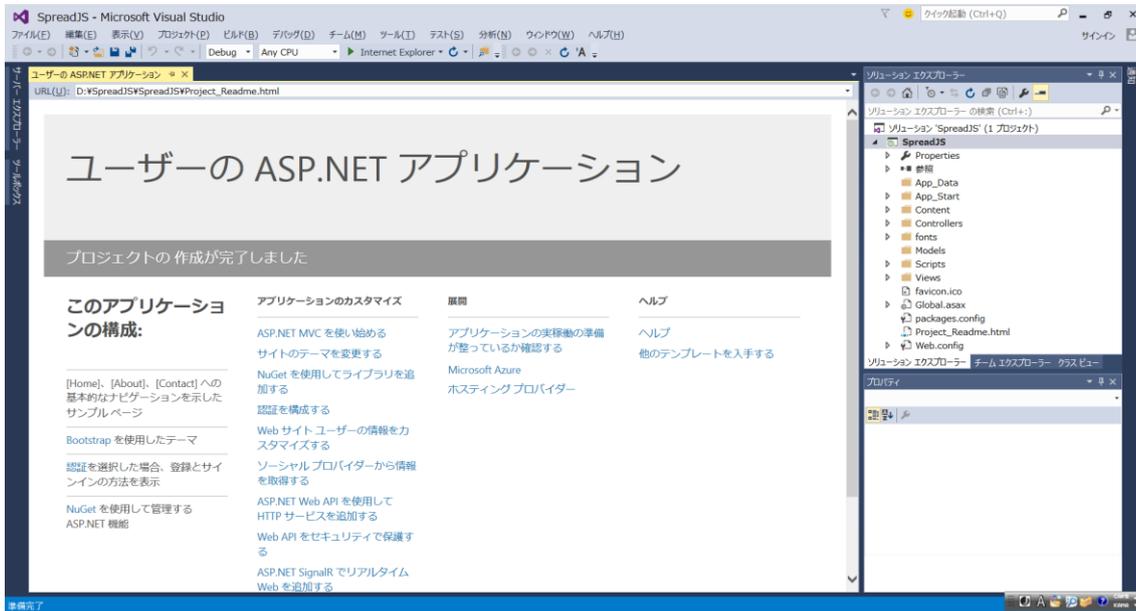
1. Visual Studio を起動し、[ファイル]メニューから[新規作成] - [プロジェクト]を選択します。
2. 左ペインのツリーメニューから [テンプレート] - [Visual C#] - [Web] を選択し、一覧から [ASP.NET Web アプリケーション] を選択します。



3. 「名前」や「場所」など必要事項を入力し、〈OK〉ボタンを押下します。
4. 表示されるダイアログの [テンプレートを選択する] から [ASP.NET 4.6 テンプレート] - [MVC] を選択し [フォルダおよびコア参照を追加する]の項目で [Web API] をチェックします (サーバー側の機能の呼び出しは HTTP メソッドを使用して行うため)。
また、サンプルプロジェクトでは認証機能は使用しませんので、[認証の変更] ボタンから [認証なし] を選択します。 [Host in the cloud] にチェックがついている場合、チェックを外します。

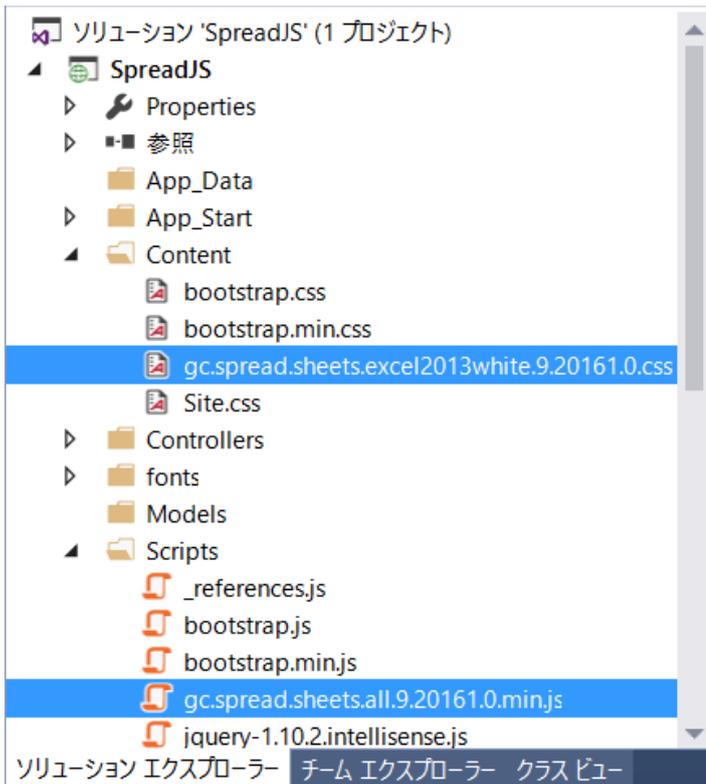


5. [OK] ボタンをクリックするとプロジェクトが作成されます。

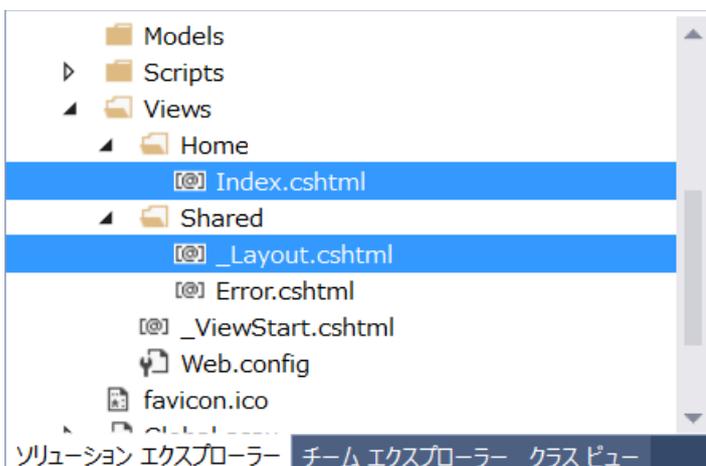


JavaScript の参照と SpreadJS の配置

SpreadJS のスクリプトファイルは [spreadsheets.zip] という zip ファイルに圧縮されていますが、zip ファイルを展開し [spreadsheets¥scripts] フォルダにある [gc.spread.sheets.all.9.20161.0.min.js] をプロジェクトの [Scripts] フォルダに、[spreadsheets¥css] フォルダにある [gc.spread.sheets.excel2013white.9.20161.0.css] を [Content] フォルダに追加します。



ASP.NET MVC では [Views] フォルダ以下にある [_Layout.cshtml] がページのテンプレートの役割を果たし、[Index.cshtml] がアプリケーションのデフォルトページになります。



[_Layout.cshtml] の<head></head>タグ内に先ほど追加したスクリプトファイルおよび CSS ファイルの参照を記述します。

_Layout.cshtml

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>@ViewBag.Title - マイ ASP.NET アプリケーション</title>
  @Styles.Render("~/Content/css")
  @Scripts.Render("~/bundles/modernizr")
  @Scripts.Render("~/bundles/jquery")
  <!--SpreadJS JavaScript-->
  <script src="~/Scripts/gc.spread.sheets.all.9.20161.0.min.js"></script>
  <!--SpreadJS CSS-->
  <link href="~/Content/gc.spread.sheets.excel2013white.9.20161.0.css" rel="stylesheet" />
</head>
```

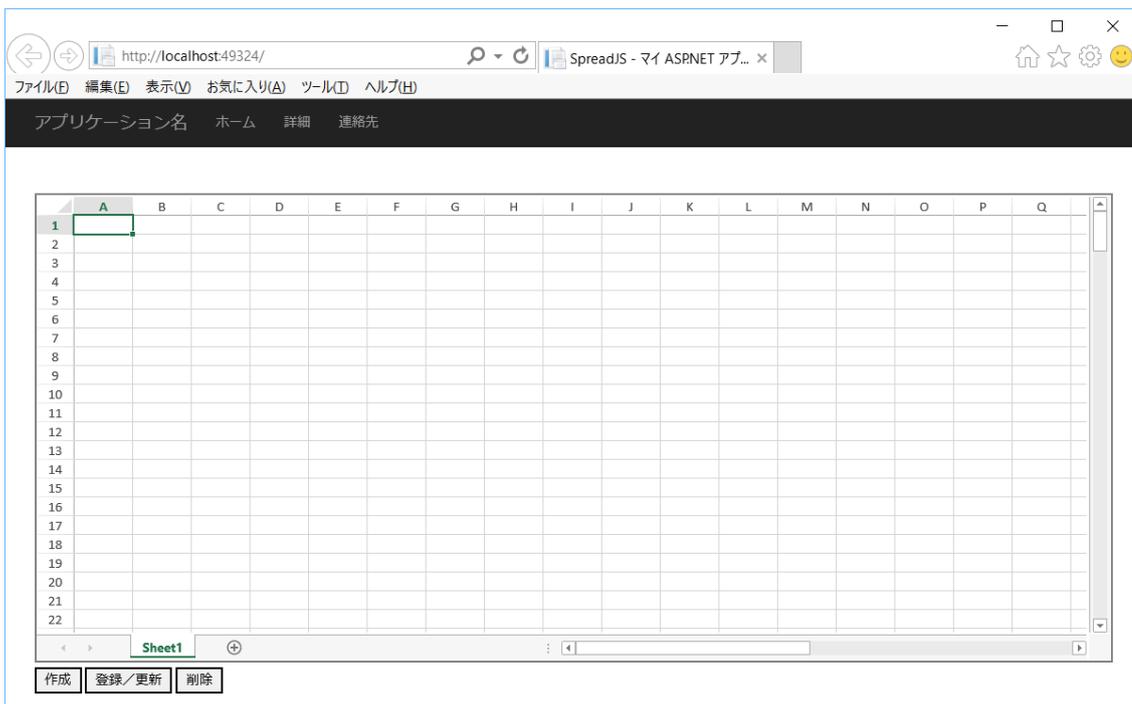
次に [Index.cshtml] を書き換えます。[Index.cshtml] では、SpreadJS の初期化ロジック、SpreadJS 用の div タグおよびデータベース操作のための各ボタンを配置します。

Index.cshtml

```
@{
    ViewBag.Title = "SpreadJS";
}
<script type="text/javascript">
    window.onload = function () {
        // 初期化ロジック
        var DOM = document.getElementById('ss');
        var spread = new GC.Spread.Sheets.Workbook(DOM, {
            sheetCount: 1
        });
        var sheet = spread.getActiveSheet();
    };
</script>

<div class="main demo">
    <div id="ss" style="width:100%;height:500px;border: 2px solid gray; margin-top: 50px;margin-bottom: 5px;margin-right:10px;float:left"></div>
    <div id="operation">
        <input type="button" value="作成" id="create" />
        <input type="button" value="登録/更新" id="update" />
        <input type="button" value="削除" id="delete" />
    </div>
</div>
```

アプリケーションを実行するとデフォルトページに SpreadJS が表示されます。

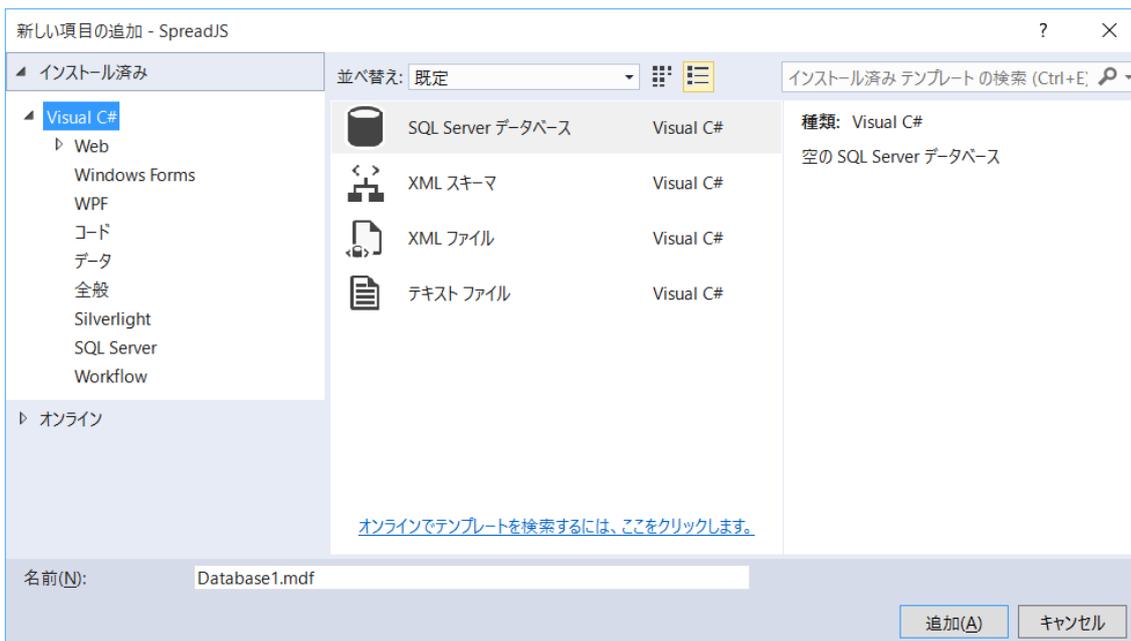


データベースとモデルの作成

データベースの作成

以下の手順でデータベースを作成します。

- ソリューションエクスプローラーの [App_Data] フォルダを右クリックし、[追加] - [新しい項目...] から新しいデータベースを作成します。
- [新しい項目の追加] ダイアログから [SQL Server データベース] を選択し、[追加] ボタンを押します。



- データベースファイルには以下のようなテーブル定義を持つテーブルを作成します。

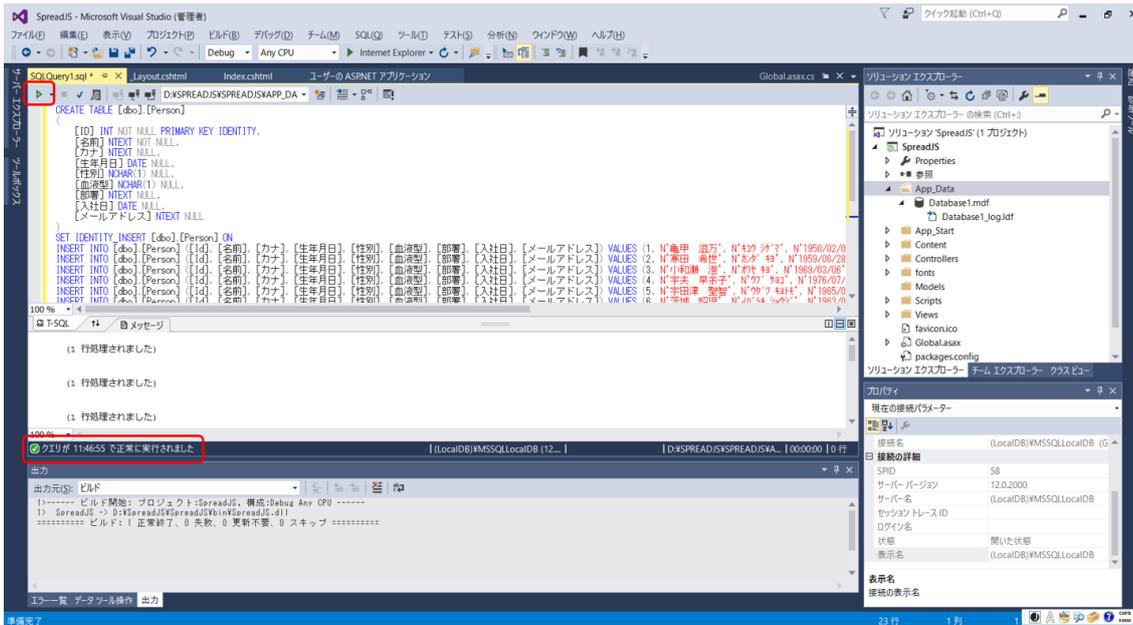
名前	データ型	Null を許容	主キー
ID	int	No	Yes
名前	ntext	No	No
カナ	ntext	Yes	No
生年月日	date	Yes	No
性別	nchar(1)	Yes	No
血液型	nchar(1)	Yes	No
部署	ntext	Yes	No
入社日	date	Yes	No
メールアドレス	ntext	Yes	No

- 次のクエリーを実行することでテーブルの作成およびデータの挿入が完了します。[サーバーエクスプローラー] の [データ接続] から先ほど作成したデータベース「Database1.mdf」のツリーを開き、[テーブル] を右クリックして [新しいクエリー] を選択します。エディタに以下のクエリーを貼り付け、[実行] ボタンを選択します。

```

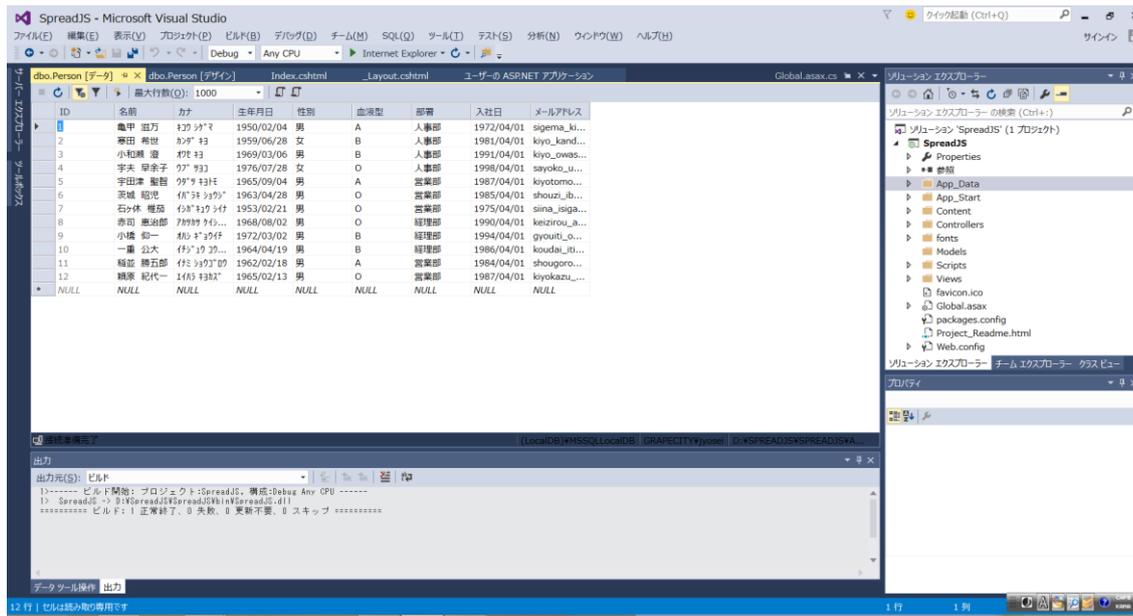
CREATE TABLE [dbo].[Person]
(
    [ID] INT NOT NULL PRIMARY KEY IDENTITY,
    [名前] NTEXT NOT NULL,
    [カナ] NTEXT NULL,
    [生年月日] DATE NULL,
    [性別] NCHAR(1) NULL,
    [血液型] NCHAR(1) NULL,
    [部署] NTEXT NULL,
    [入社日] DATE NULL,
    [メールアドレス] NTEXT NULL
)
SET IDENTITY_INSERT [dbo].[Person] ON
INSERT INTO [dbo].[Person] ([Id], [名前], [カナ], [生年月日], [性別], [血液型], [部署], [入社日], [メールアドレス]) VALUES (1,
N'亀甲 滋万', N'カメガキ シゲマ', N'1950/02/04', N'男', N'A', N'人事部', N'1972/04/01', N'sigema_kikou@abc.co.jp')
INSERT INTO [dbo].[Person] ([Id], [名前], [カナ], [生年月日], [性別], [血液型], [部署], [入社日], [メールアドレス]) VALUES (2,
N'寒田 希世', N'カマタ キセ', N'1959/06/28', N'女', N'B', N'人事部', N'1981/04/01', N'kiyo_kanda@bbb.or.jp')
INSERT INTO [dbo].[Person] ([Id], [名前], [カナ], [生年月日], [性別], [血液型], [部署], [入社日], [メールアドレス]) VALUES (3,
N'小和瀬 澄', N'コワセ アキラ', N'1969/03/06', N'男', N'B', N'人事部', N'1991/04/01', N'kiyo_owase@aaa.co.jp')
INSERT INTO [dbo].[Person] ([Id], [名前], [カナ], [生年月日], [性別], [血液型], [部署], [入社日], [メールアドレス]) VALUES (4,
N'宇夫 早余子', N'ウツノ ヒココ', N'1976/07/28', N'女', N'O', N'人事部', N'1998/04/01', N'sayoko_ubu@bbb.or.jp')
INSERT INTO [dbo].[Person] ([Id], [名前], [カナ], [生年月日], [性別], [血液型], [部署], [入社日], [メールアドレス]) VALUES (5,
N'宇田津 聖智', N'ウダツ ヒロトシ', N'1965/09/04', N'男', N'A', N'営業部', N'1987/04/01', N'kiyotomo_udatu@abc.co.jp')
INSERT INTO [dbo].[Person] ([Id], [名前], [カナ], [生年月日], [性別], [血液型], [部署], [入社日], [メールアドレス]) VALUES (6,
N'茨城 昭児', N'イバラキ ショウジ', N'1963/04/28', N'男', N'O', N'営業部', N'1985/04/01', N'shouzi_ibaraki@xyz.ne.jp')
INSERT INTO [dbo].[Person] ([Id], [名前], [カナ], [生年月日], [性別], [血液型], [部署], [入社日], [メールアドレス]) VALUES (7,
N'石ヶ休 椎茄', N'イシガキ ヒロシ', N'1953/02/21', N'男', N'O', N'営業部', N'1975/04/01', N'siina_isigagyuu@abc.co.jp')
INSERT INTO [dbo].[Person] ([Id], [名前], [カナ], [生年月日], [性別], [血液型], [部署], [入社日], [メールアドレス]) VALUES (8,
N'赤司 恵治郎', N'アカシ ケイジロウ', N'1968/08/02', N'男', N'O', N'経理部', N'1990/04/01', N'keizirou_akatukasa@abc.co.jp')
INSERT INTO [dbo].[Person] ([Id], [名前], [カナ], [生年月日], [性別], [血液型], [部署], [入社日], [メールアドレス]) VALUES (9,
N'小橋 仰一', N'コハシ キョウイチ', N'1972/03/02', N'男', N'B', N'経理部', N'1994/04/01', N'gyouiti_ohasi@abc.co.jp')
INSERT INTO [dbo].[Person] ([Id], [名前], [カナ], [生年月日], [性別], [血液型], [部署], [入社日], [メールアドレス]) VALUES
(10, N'一重 公大', N'イツムラ ヒロシ', N'1964/04/19', N'男', N'B', N'経理部', N'1986/04/01', N'koudai_itizyuu@xyz.ne.jp')
INSERT INTO [dbo].[Person] ([Id], [名前], [カナ], [生年月日], [性別], [血液型], [部署], [入社日], [メールアドレス]) VALUES
(11, N'稲並 勝五郎', N'イナナミ カチウロウ', N'1962/02/18', N'男', N'A', N'営業部', N'1984/04/01', N'shougorou_inami@bbb.or.jp')
INSERT INTO [dbo].[Person] ([Id], [名前], [カナ], [生年月日], [性別], [血液型], [部署], [入社日], [メールアドレス]) VALUES
(12, N'穎原 紀代一', N'エハラ キヨイチ', N'1965/02/13', N'男', N'O', N'営業部', N'1987/04/01', N'kiyokazu_eihara@bbb.or.jp')
SET IDENTITY_INSERT [dbo].[Person] OFF

```



なお、クエリは「SQLQuery1.sql」という名前前で保存します。

5. [サーバーエクスプローラー] 内のツリーで先ほど作成したテーブル [person] を右クリックし、[テーブルデータの表示] を選択して Visual Studio 上に一覧を表示します。



モデルの作成

ここでは ADO.NET Entity Data Model を使用します。

【メモ】

- ADO.NET Entity Framework が登場した背景

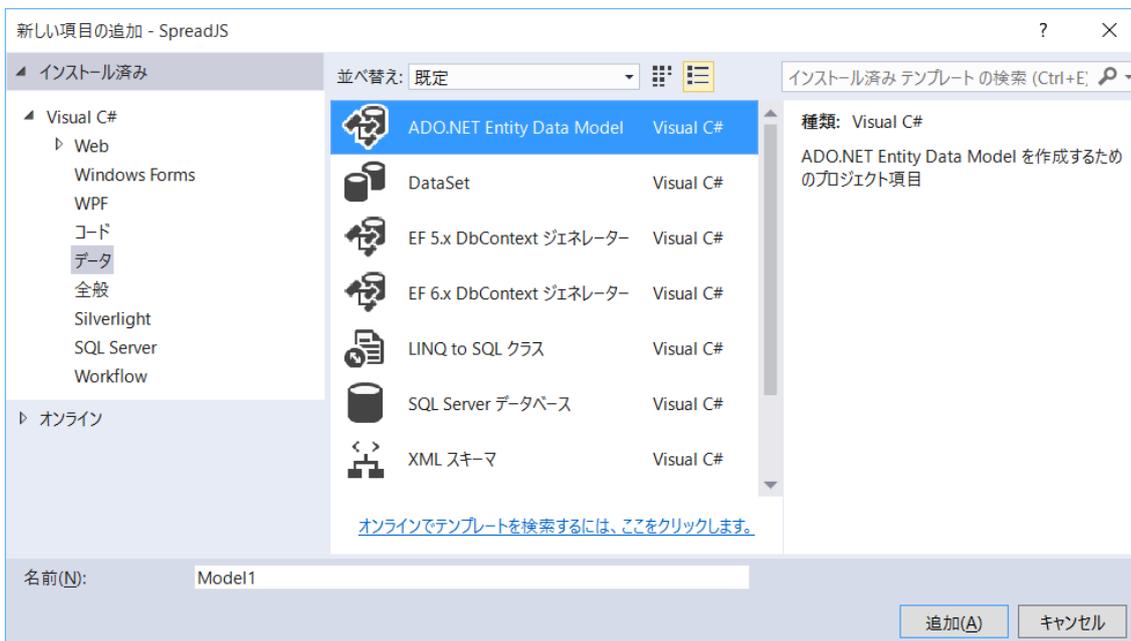
<http://blogs.msdn.com/b/daisukei/archive/2014/08/29/net-framework-4-5-3-ado-net-entity-framework.aspx>

■ 第 4 回 データベースからの Entity Data Model 生成

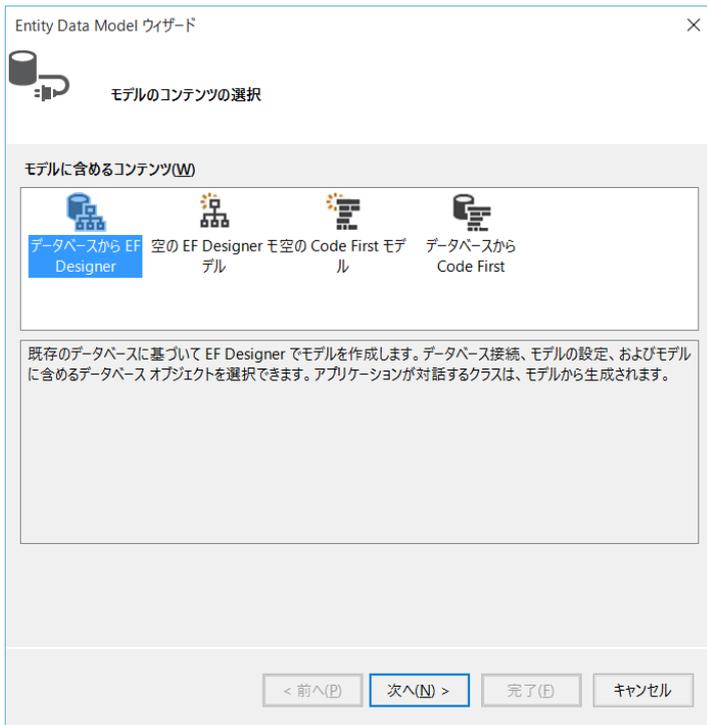
http://www.atmarkit.co.jp/fdotnet/ef4basic/ef4basic04/ef4basic04_01.html

以下の手順でモデルを作成します。

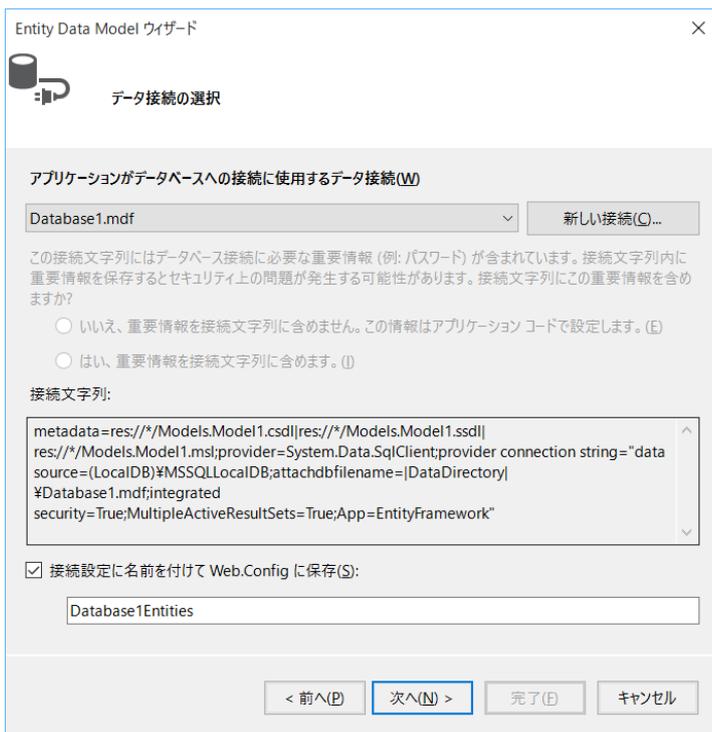
1. ソリューションエクスプローラーの [Models] フォルダを右クリックし、[追加] - [新しい項目...] を選択します。
2. 表示されるダイアログで [ADO.NET Entity Data Model] を選択し、[追加] ボタンを選択します。



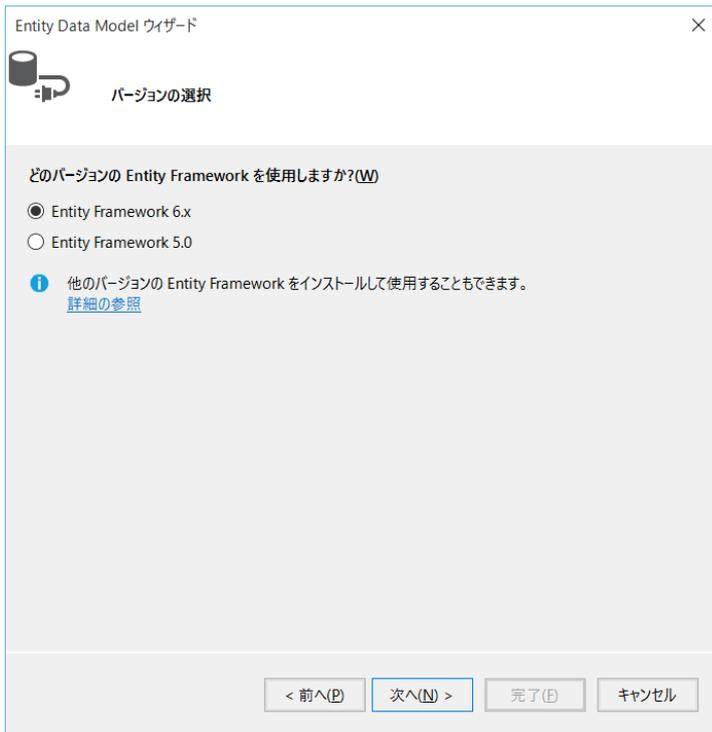
3. モデルは取り込んだデータベースから作成しますので、[データベースから EF Designer] を選択します。



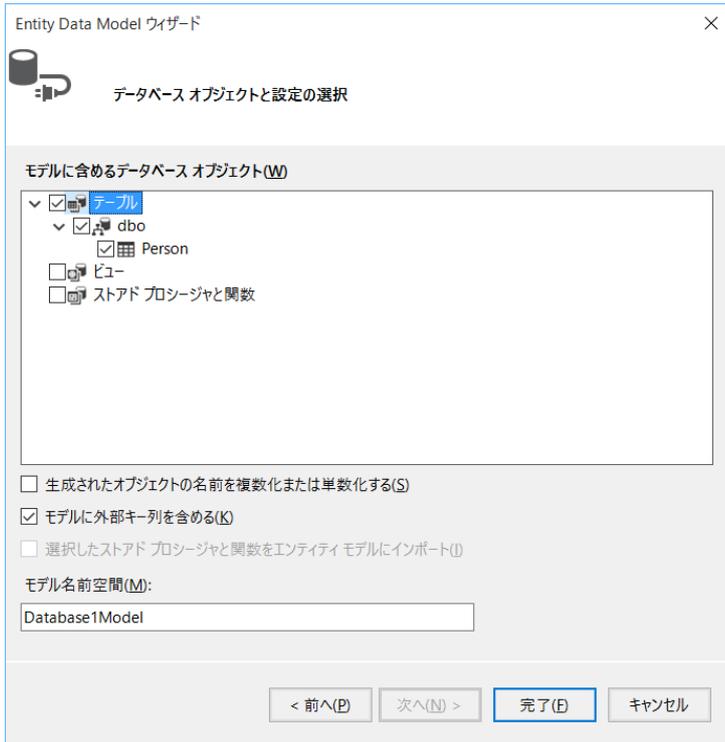
4. ウィザードの [データ接続の選択] ページが表示されますが、そのまま [次へ] を選択します。



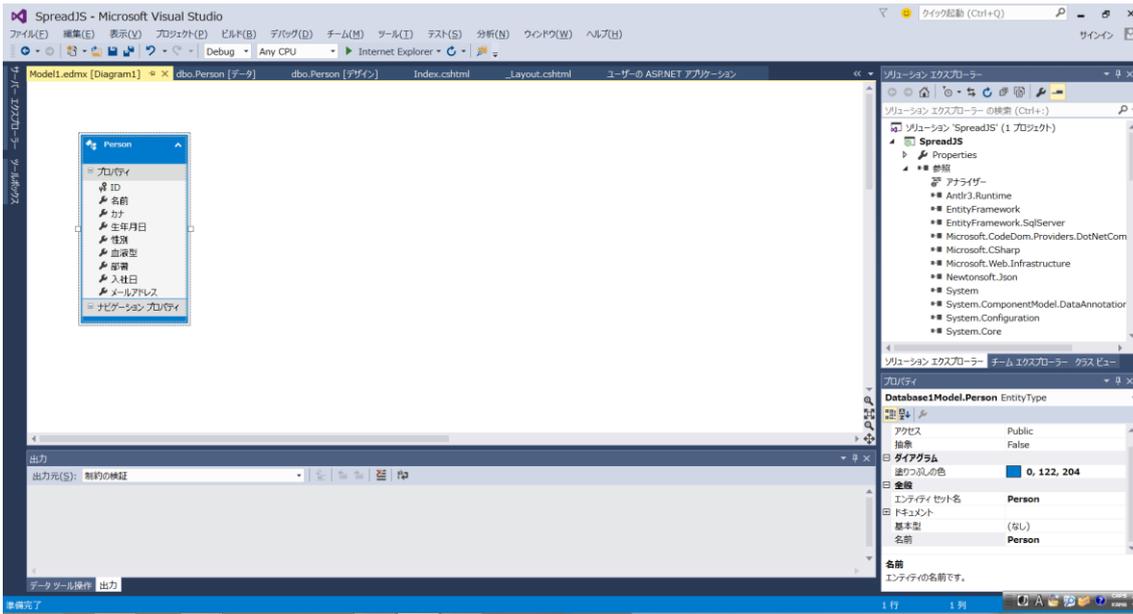
5. [Entity Framework 6.x] を使用します。



6. Person テーブルを選択し [完了] ボタンを押下します。なお、警告ダイアログが表示される場合がありますが、問題はありませんのでそのまま [OK] ボタンを押下します。



7. [Person] テーブルのみのモデルが作成されます。

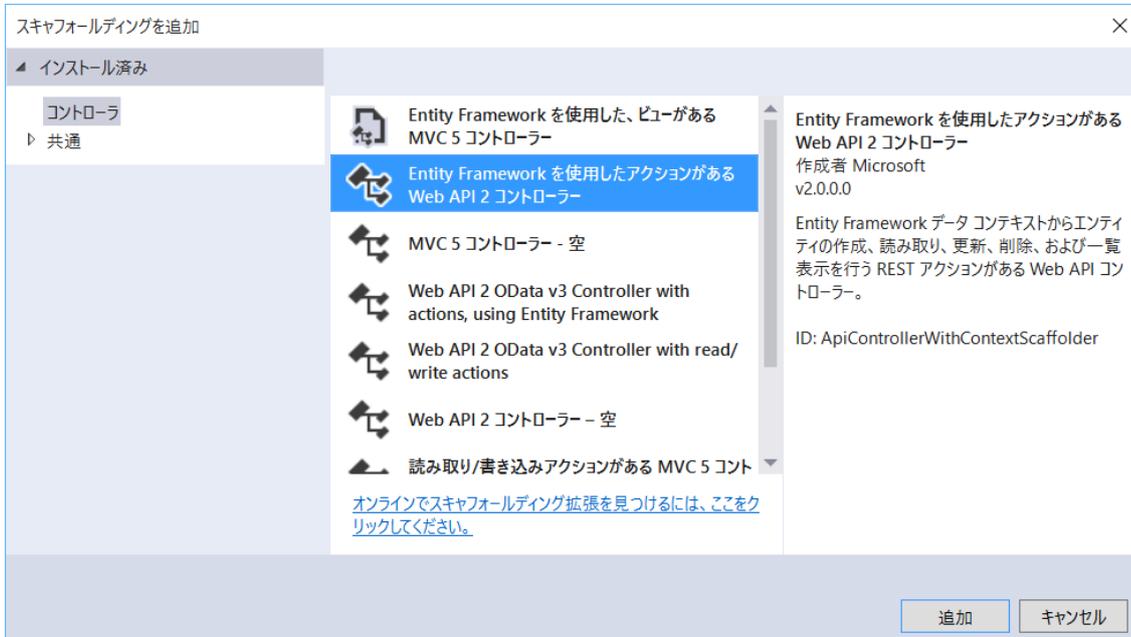


8. ここでプロジェクトを一度ビルドします。

コントローラーの作成

以下の手順で Web API 2 コントローラーを使用して新しいコントローラーを作成します。

1. ソリューションエクスプローラーの [Controllers] フォルダを右クリックし、[追加] - [コントローラー] を選択します。



Web API 2 コントローラーを使用してコントローラーを作成すると、各データ操作はそれぞれ次の HTTP メソッドを経由して行うことになります。

データ操作	HTTP メソッド
作成	POST
取得	GET
更新	PUT
削除	DELETE

2. モデルクラスに [Person]、データコンテキストクラスに [Database1Entities]、コントローラー名に [PersonController] を設定し [追加] ボタンを押下します。

コントローラーの追加 ×

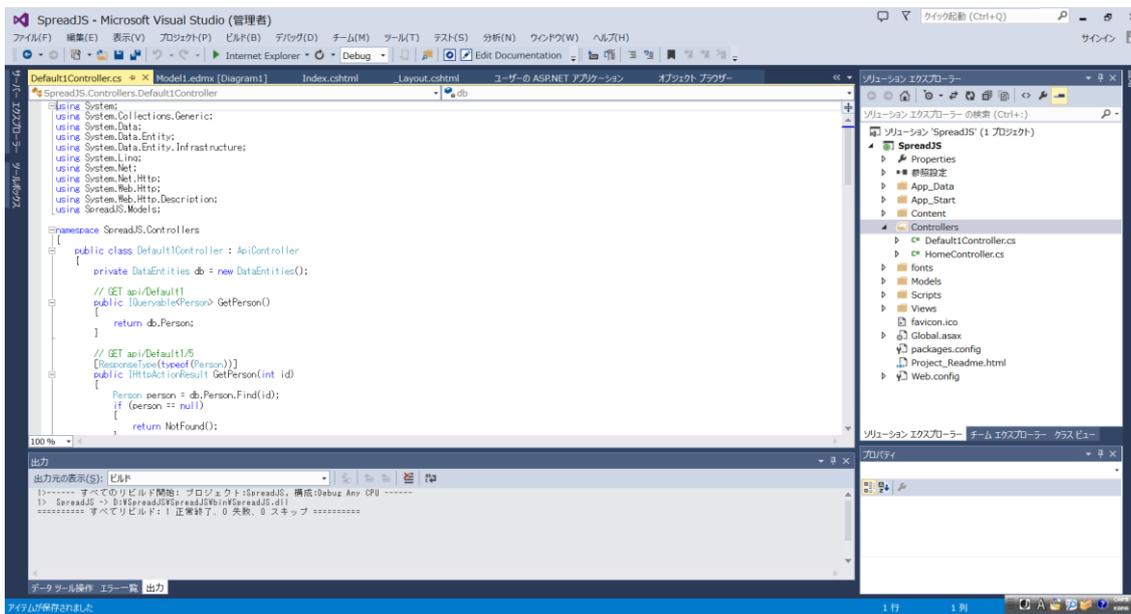
モデル クラス(M):

データ コンテキスト クラス(D): +

非同期コントローラー アクションの使用(A)

コントローラー名(O):

3. [PersonController.cs] が生成されます。



4. [PersonController.cs] には 5 つの public メソッドが含まれますが、該当のメソッドはそれぞれ以下のような HTTP メソッドおよび URL を指定することで操作可能です。

Public メソッド	HTTP メソッド	URL
GetPerson()	GET	api/Person
GetPerson(int id)	GET	api/Person/id
PutPerson(int id, Person person)	PUT	api/Person/id
PostPerson(Person person)	POST	api/Person
DeletePerson(int id)	DELETE	api/Person/id

SpreadJS を使用した CRUD の実装

これまでの手順によりサーバー側の実装はすでに完了しています。ここでは SpreadJS を使用し、クライアント側からサーバーのメソッドを呼び出すことで CRUD を実装していきます。

データ一覧の表示

最初に一覧表示のロジックを入れます。挿入箇所は、index.cshtml にある SpreadJS 初期化ロジックの後になります。具体的な処理内容としては、HTTP メソッドの GET により一覧を取得したあと、シートのスタイルおよび列幅を調整しています。

index.cshtml

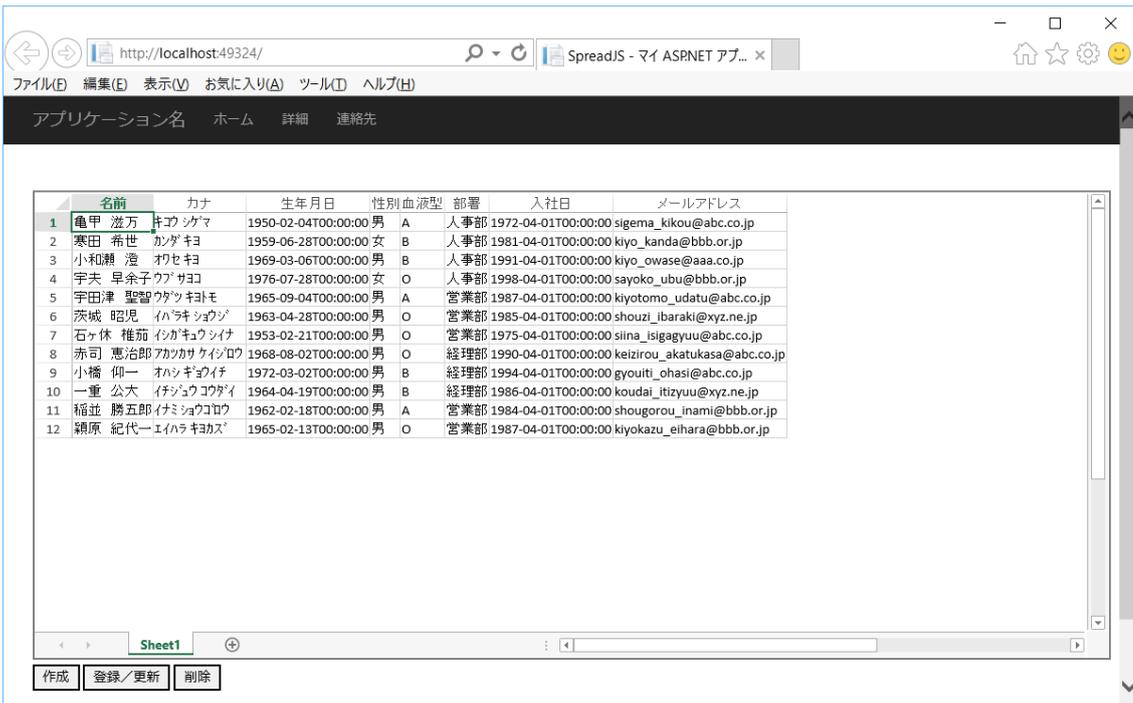
```
$.getJSON("/api/Person", function (data) {
    sheet.suspendPaint();

    // データ連結
    sheet.setDataSource(data);
    var columnInfos = [
        { name: "名前" },
        { name: "カナ" },
        { name: "生年月日" },
        { name: "性別" },
        { name: "血液型" },
        { name: "部署" },
        { name: "入社日" },
        { name: "メールアドレス" }
    ];
    sheet.bindColumns(columnInfos);

    // 行モード／単一選択
    sheet.selectionUnit(GC.Spread.Sheets.SelectionUnit.row);
    sheet.selectionPolicy(GC.Spread.Sheets.SelectionPolicy.single);

    // 各列の列幅調整
    spread.options.autoFitType = GC.Spread.Sheets.AutoFitType.cellWithHeader;
    for (var i = 0; i < sheet.getColumnCount(); i++) {
        sheet.autoFitColumn(i);
    }

    sheet.resumePaint();
});
```



[作成] ボタン押下時の処理

[作成] ボタンの処理としては、最終行の新しい空の行を追加し、該当列に画面をスクロールしています。この時点ではサーバーへのアクセスはありません。

index.cshhtml

```
// 作成
$("#create").click(function () {
    var insertRows = sheet.getInsertRows();
    // 新規行がない場合
    if (insertRows.length == 0) {
        // 新規行を作成
        var rowCount = sheet.getRowCount();
        sheet.addRows(rowCount, 1);
        // アクティブセルの移動/スクロール
        sheet.setActiveCell(rowCount, 0);
        sheet.showCell(rowCount, 0, GC.Spread.Sheets.VerticalPosition.center,
GC.Spread.Sheets.HorizontalPosition.center);
    }
});
```

	名前	カナ	生年月日	性別	血液型	部署	入社日	メールアドレス
3	小和精 澄	オワセキヨ	1969-03-06T00:00:00	男	B	人事部	1991-04-01T00:00:00	kiyo_owase@aaa.co.jp
4	宇夫 早余子ウツ サヨコ	ウツサヨコ	1976-07-28T00:00:00	女	O	人事部	1998-04-01T00:00:00	sayoko_ubu@bbb.or.jp
5	宇田津 聖智ウタツキキトモ	ウタツキキトモ	1965-09-04T00:00:00	男	A	営業部	1987-04-01T00:00:00	kiyotomo_udatu@abc.co.jp
6	茨城 昭児 伊ハキショウジ	イハキショウジ	1963-04-28T00:00:00	男	O	営業部	1985-04-01T00:00:00	shouji_ibaraki@xyz.jp
7	石ヶ体 稚萌 イシガキユウシナ	イシガキユウシナ	1953-02-21T00:00:00	男	O	営業部	1975-04-01T00:00:00	siina_isigagiyuu@abc.co.jp
8	赤司 恵治郎アカサケイジロウ	アカサケイジロウ	1968-08-02T00:00:00	男	O	経理部	1990-04-01T00:00:00	keizirou_akatukasa@abc.co.jp
9	小橋 仰一 オハシキョウイチ	オハシキョウイチ	1972-03-02T00:00:00	男	B	経理部	1994-04-01T00:00:00	gyouti_ohasi@abc.co.jp
10	一重 公大 イチジュウコウダイ	イチジュウコウダイ	1964-04-19T00:00:00	男	B	経理部	1986-04-01T00:00:00	koudai_itizyuu@xyz.ne.jp
11	稲並 勝五郎イナミショウゴロウ	イナミショウゴロウ	1962-02-18T00:00:00	男	A	営業部	1984-04-01T00:00:00	shougorou_inami@bbb.or.jp
12	絹原 紀代エハラキヨカズ	エハラキヨカズ	1965-02-13T00:00:00	男	O	営業部	1987-04-01T00:00:00	kiyokazu_eihara@bbb.or.jp
13								

[登録/更新] ボタン押下時の処理

[登録/更新] ボタンでは、登録および更新処理を実装しています。新規行が存在する場合には登録処理を行い、データ行が存在する場合には更新処理を行います。

index.cshhtml

```
// 登録/更新
$("#update").click(function () {
    // 新規行がある場合
    var insertRows = sheet.getInsertRows();
    if (insertRows.length > 0) {
        insertRows[0].item["ID"] = -1;
        $.ajax({
            url: "/api/Person",
            type: 'POST',
            data: JSON.stringify(insertRows[0].item),
            contentType: "application/json",
            success: function (data) {
                alert("登録完了");
            },
            error: function (ex) {
                alert('登録失敗');
            }
        });
    }
    // データ行がある場合
```

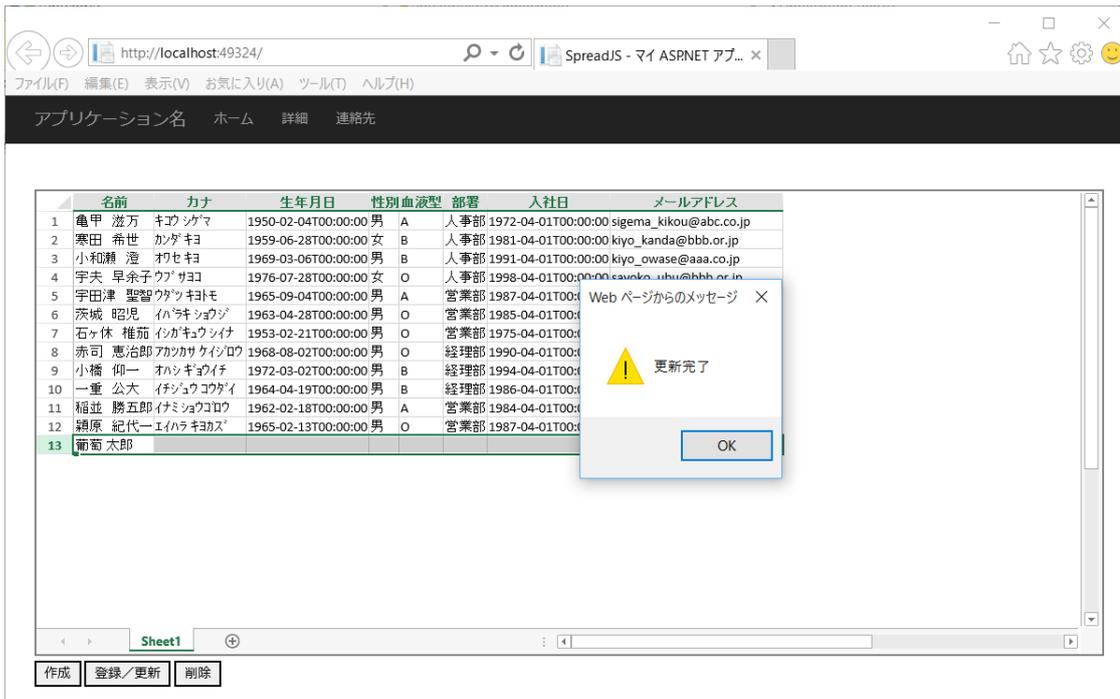
```
var dirtyRows = sheet.getDirtyRows();
if (dirtyRows.length > 0) {
    var deferreds = updateRecords();

    $.when.apply($, deferreds).done(function () {
        alert('更新完了');
    }).fail(function () {
        alert('更新失敗');
    });
}

// ダーティ状態の解除
sheet.clearPendingChanges();

// レコードの更新処理
function updateRecords() {
    var deferreds = [];
    for (var i = 0; i < dirtyRows.length ; i++) {
        var id = dirtyRows[i].item["ID"];

        deferreds.push(
            $.ajax({
                url: "/api/Person/" + id,
                type: 'PUT',
                data: JSON.stringify(dirtyRows[i].item),
                contentType: "application/json"
            })
        );
    }
    return deferreds;
}
});
```



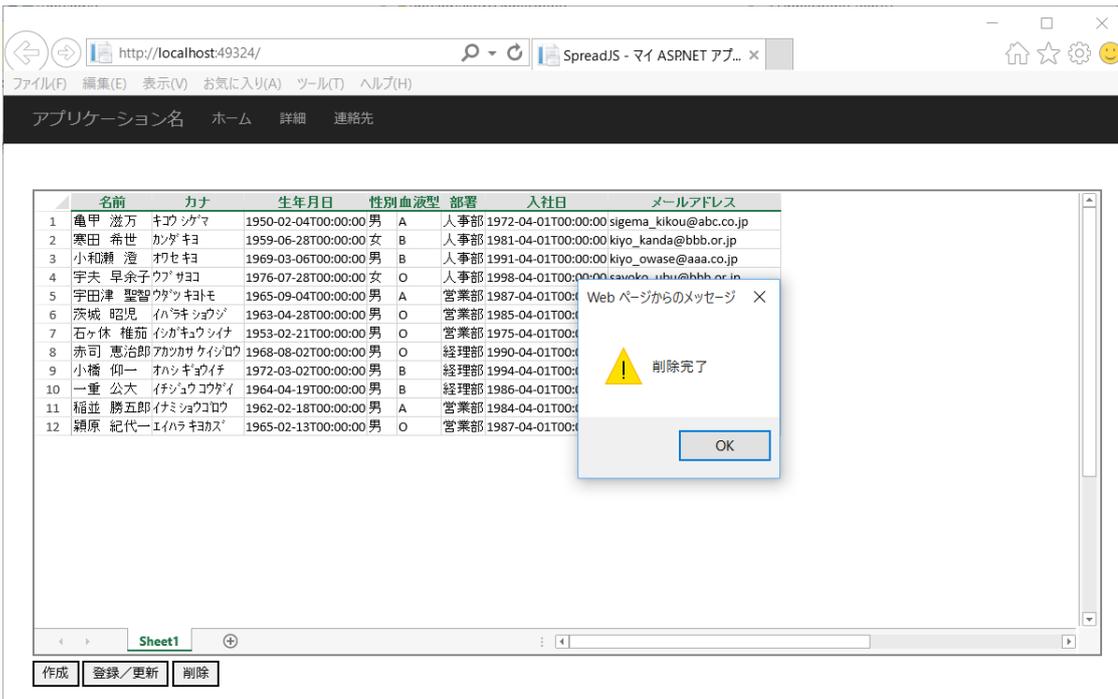
[削除] ボタン押下時の処理

最後に削除のロジックを入れます。アクティブ行の ID を取得し、HTTP メソッドの DELETE によりコールします。

index.cshtml

```
// 削除
$("#delete").click(function () {
    var deletingRowIndex = sheet.getActiveRowIndex();
    var id = sheet.getDataItem(deletingRowIndex)["ID"];

    $.ajax({
        url: "/api/Person/" + id,
        type: 'DELETE',
        success: function (data) {
            // シート上の行の削除
            sheet.deleteRows(deletingRowIndex, 1);
            alert("削除完了");
        },
        error: function (ex) {
            alert("削除失敗");
        }
    });
});
```



まとめ

本チュートリアルでは、ASP.NET MVC のプロジェクトテンプレートを用い、SpreadJS をクライアントサイドのコントロールとして使用する場合に、どのようにアプリケーションを構築していくかを具体的な手順を用いて紹介しました。実務で開発するシステムでは、より多くのビジネスロジックが要求されることと思いますが、今回作成したアプリケーションは、基本的なデータベース操作機能を備えています。作成されたアプリケーションをベースに、または考え方を参考にさせていただき、SpreadJS のアプリケーション開発にご活用ください。

あわせて製品の「オンラインヘルプ」および「チュートリアルデモ」もご参考にいただければと思います。

- オンラインヘルプ

<http://docs.grapacity.com/help/spread-js-9/>

- チュートリアルデモ

<http://demo.grapacity.com/spread/spreadjs/tutorialsample/>