CST8259 Web Programming Language II

Lab 2

Objective

- Understand and Create Schema for an XML document,
- Validate XML documents in C#.

Due Date

See Brightspace for the due date of this lab. To earn 5 points, you are required:

1. Complete the lab as required.
2. Submit your lab work to the Brightspace before the due date.
3. Demo your lab work during the lab session in the week after the due date.

Requirements

Part 1 – XML Schema

Create an XML Schema for your ***Restaurant_Reviews.xml*** document you created in Lab 1.

1. The schema should target the following namespace:

   `http://www.algonquincollege.com/cst8259/labs`

2. The schema should define the following elements and types:

   a) A global element to use as the root element of your XML document.

   b) It defines a global element for use inside both **appetizers** and **entrees** elements to define menu items.

   c) A global element "**address**" to be referenced in the schema for the address element of a restaurant.

   d) A simple type of string with the restriction of a regular express pattern for postal code element in the global address element defined in **c)**.

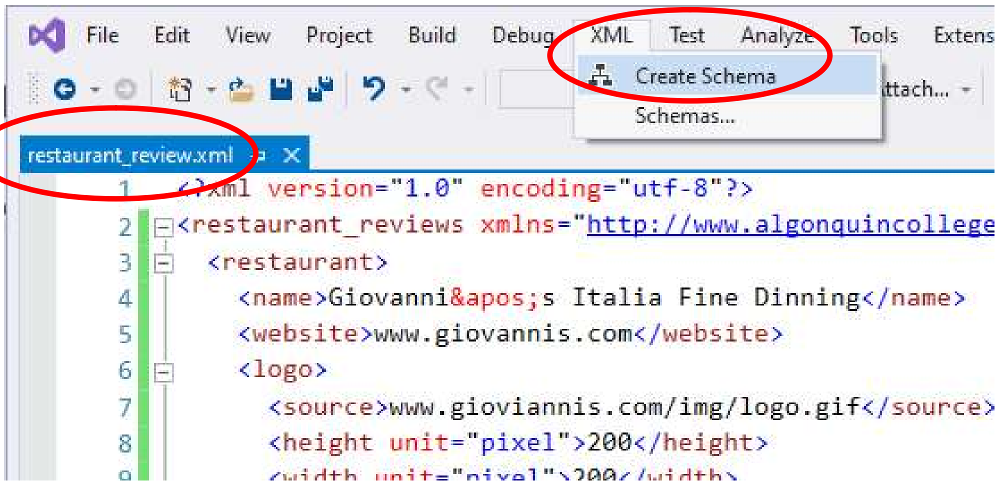
e) It defines a simple type of string with the restriction of the following international standard code for Canadian provinces:

AB,
BC,
MB,
NB,
NL,
NS,
ON,
PE,
QC,
SK,
NT,
NU,
YT

The type is referenced by the **province** element in the global **address** element defined in **c)**.

Hint:

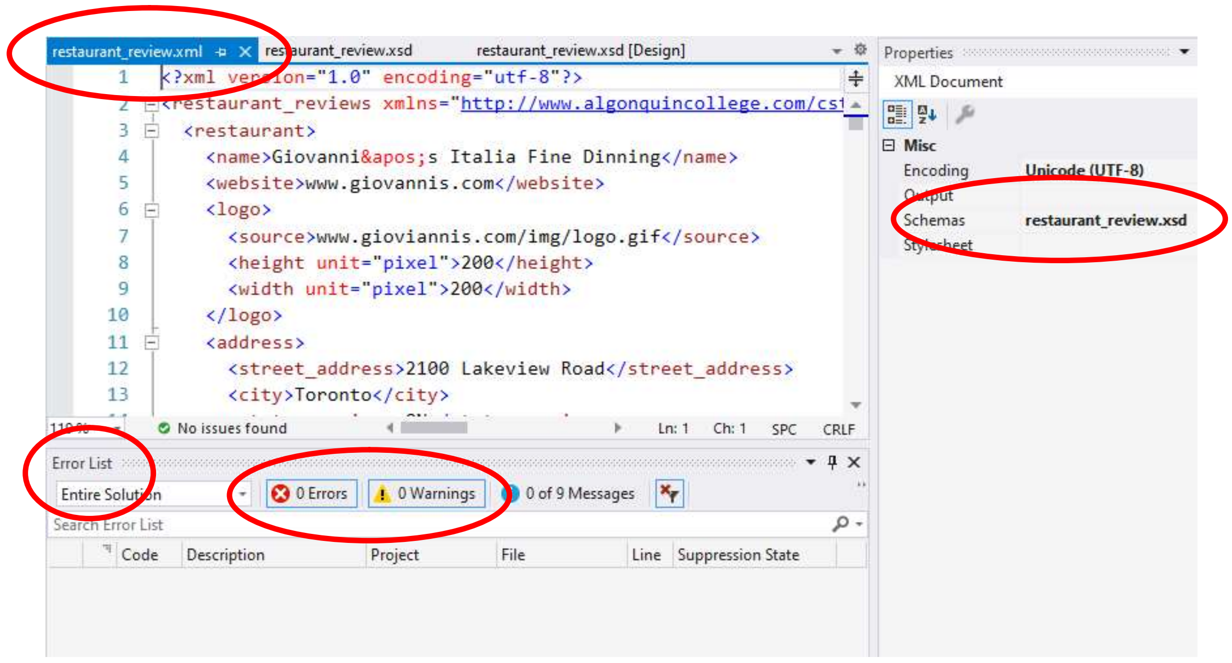
After completing your XML document, you can generate its schema in Visual Studio by select menu item **XML > Create Schema**



```
<?xml version="1.0" encoding="utf-8"?>
<restaurant_reviews xmlns="http://www.algonquincollege
  <restaurant>
    <name>Giovanni&apos;s Italia Fine Dinning</name>
    <website>www.giovannis.com</website>
    <logo>
      <source>www.gioviannis.com/img/logo.gif</source>
      <height unit="pixel">200</height>
      <width unit="pixel">200</width>
```

You can then modify the generated schema to factor out the definitions of local elements for address and menu items to create the global elements as required above.

3. Modify **your restaurant_reviews.xml** file to ensure that there is not error nor warning message in Visual Studio's Error List window when the schema is used by Visual Studio to validate your restaurant_review.xml file.



```
<?xml version="1.0" encoding="utf-8"?>
<restaurant_reviews xmlns="http://www.algonquincollege.com/cst
  <restaurant>
    <name>Giovanni&apos;s Italia Fine Dinning</name>
    <website>www.giovannis.com</website>
    <logo>
      <source>www.giovannis.com/img/logo.gif</source>
      <height unit="pixel">200</height>
      <width unit="pixel">200</width>
    </logo>
    <address>
      <street_address>2100 Lakeview Road</street_address>
      <city>Toronto</city>
```

No issues found   Ln: 1   Ch: 1   SPC   CRLF

Error List

Entire Solution   0 Errors   0 Warnings   0 of 9 Messages

Search Error List

	Code	Description	Project	File	Line	Suppression State

Encoding	Unicode (UTF-8)
Output	
Schemas	restaurant_review.xsd
Stylesheet	

Hints:

a) To minimumize the changes, set the target namespace defined in the schema as default namespace of your restaurant_review XML document.

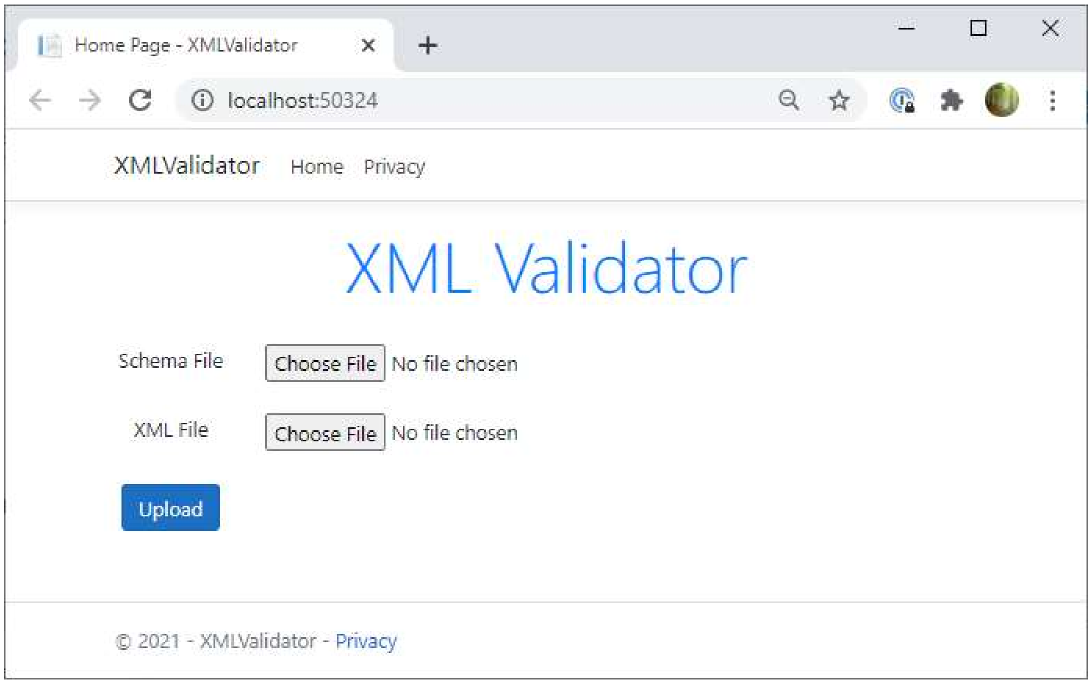
b) If you don't see the Error List window in Visual Studio, select menu item View > Error List.

Part 2 – Validate XML in ASP.NET webform application

Create an ASP.NET MVC Core web application for validating XML files against their Schema.

As a professional web developer, you should properly style your web pages. At minimum, all elements on the page should aligned, evenly size and spaced.

1. On start, the application presents the user a view to upload a schema file and a XML file to the server for validate the XML file against the schema:



Hints

- Use the following HTML elements for file uploading in the view:

```
<input asp-for="SchemaFile" type="file" accept=".xsd" required>

<input asp-for="XmlFile" type="file" accept=".xml" required>
```

- Define a model class for binding the form data (two files in this case) in the view to the parameter of the action the form is posted to

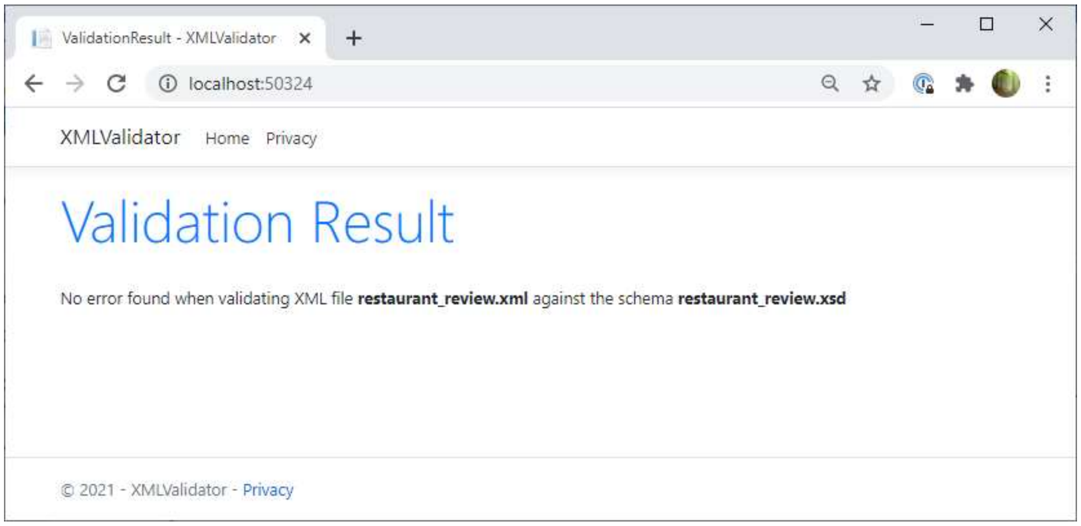
```
public class XMLandSchemaFileUpload
{
        [Display(Name = "XML File")]
        public IFormFile XmlFile { get; set; }

        [Display(Name = "Schema File")]
        public IFormFile SchemaFile { get; set; }
 }
```

- Google the .Net built-in interface IFormFile to find out its methods and properties you can use for processing the uploaded files.

- For simplicity, use the default Layout for the application's views.

- For simplicity, sever side file type validations are not required (not safe). You can rely on HTML input element's **accept** and **required** attributes to safeguard that the correct files are specified by the user.

2. After the user uploaded the schema file and the XML file, the application validates the XML file against the schema file. If no error is found, the application informs the user with a view:



Validation Result

No error found when validating XML file **restaurant_review.xml** against the schema **restaurant_review.xsd**

© 2021 - XMLValidator - Privacy

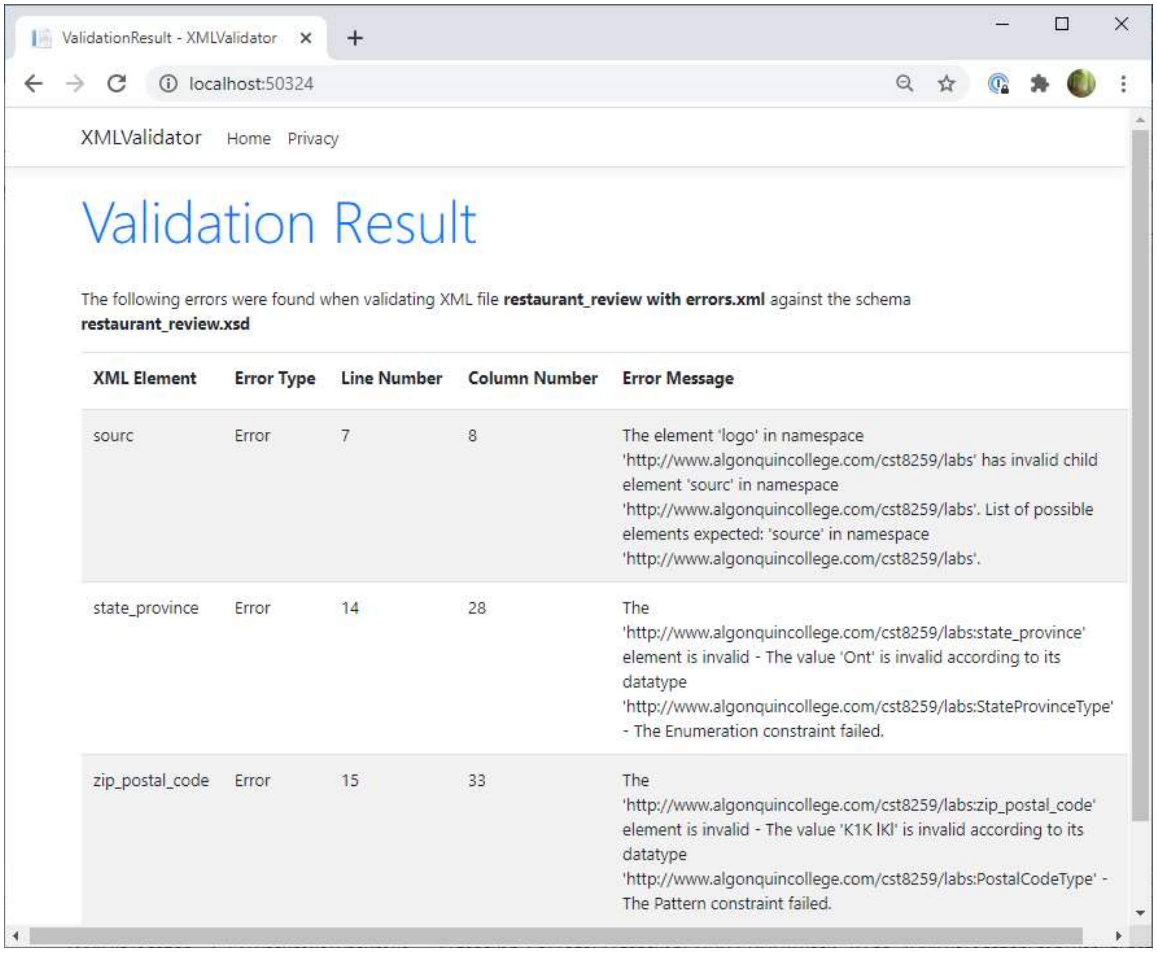
3. If the validation found errors in the XML file, it presents the error message to the user, as shown on the following page, with the following details for each error:

- XML element containing the error
- Type of the error
- Line and column number at which error is located
- The details description of the error

Hint

- Define a model class to hold the details for an error found in the XML file:

```
public class XmlValidationError
{
        [Display(Name = "XML Element")]
        public string Element { get; set; }

        [Display(Name = "Error Type")]
        public string ErrorType { get; set; }

        [Display(Name = "Line Number")]
        public int Line { get; set; }

        [Display(Name = "Column Number")]
        public int Column { get; set; }

        [Display(Name = "Error Message")]
        public string Message { get; set; }
 }
```



ValidationResult - XMLValidator

localhost:50324

XMLValidator Home Privacy

Validation Result

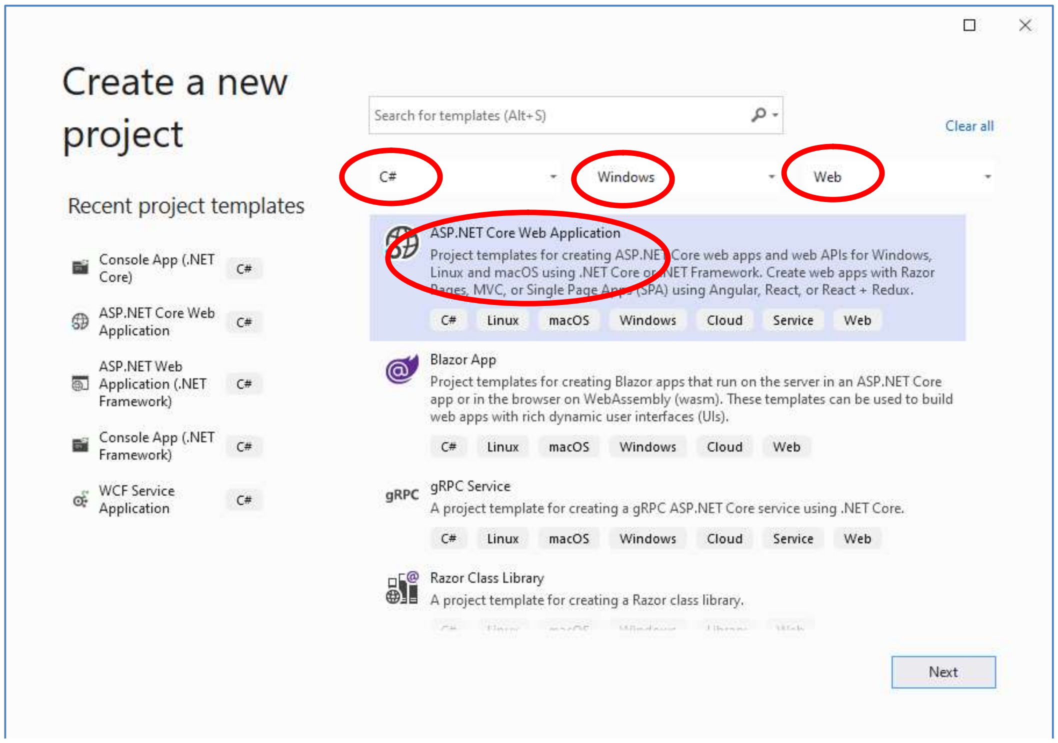
The following errors were found when validating XML file **restaurant_review with errors.xml** against the schema **restaurant_review.xsd**

XML Element	Error Type	Line Number	Column Number	Error Message
sourc	Error	7	8	The element 'logo' in namespace 'http://www.algonquincollege.com/cst8259/labs' has invalid child element 'sourc' in namespace 'http://www.algonquincollege.com/cst8259/labs'. List of possible elements expected: 'source' in namespace 'http://www.algonquincollege.com/cst8259/labs'.
state_province	Error	14	28	The 'http://www.algonquincollege.com/cst8259/labs:state_province' element is invalid - The value 'Ont' is invalid according to its datatype 'http://www.algonquincollege.com/cst8259/labs:StateProvinceType' - The Enumeration constraint failed.
zip_postal_code	Error	15	33	The 'http://www.algonquincollege.com/cst8259/labs:zip_postal_code' element is invalid - The value 'K1K lKl' is invalid according to its datatype 'http://www.algonquincollege.com/cst8259/labs:PostalCodeType' - The Pattern constraint failed.

Implementation Notes:

If you forget how to develop an ASP.NET MVC Core web application, you should review what you learnt in **CST8256 Web Programming Language I**. The followings are some highlights:

1. Following the steps below to create an ASP.NET MVC Core web application



Create a new project

Recent project templates

Console App (.NET Core)	C#
ASP.NET Core Web Application	C#
ASP.NET Web Application (.NET Framework)	C#
Console App (.NET Framework)	C#
WCF Service Application	C#

Search for templates (Alt+S)

Clear all

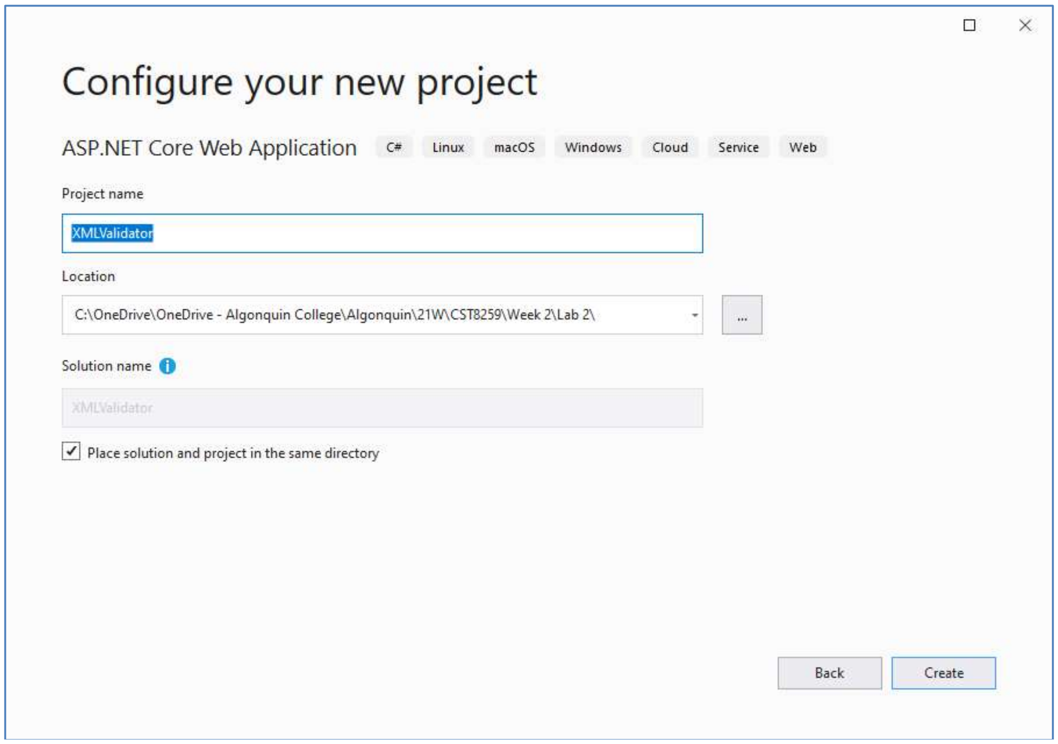
C# | Windows | Web

ASP.NET Core Web Application
Project templates for creating ASP.NET Core web apps and web APIs for Windows, Linux and macOS using .NET Core or .NET Framework. Create web apps with Razor Pages, MVC, or Single Page Apps (SPA) using Angular, React, or React + Redux.
C# Linux macOS Windows Cloud Service Web

Blazor App
Project templates for creating Blazor apps that run on the server in an ASP.NET Core app or in the browser on WebAssembly (wasm). These templates can be used to build web apps with rich dynamic user interfaces (UIs).
C# Linux macOS Windows Cloud Web

gRPC Service
A project template for creating a gRPC ASP.NET Core service using .NET Core.
C# Linux macOS Windows Cloud Service Web

Razor Class Library
A project template for creating a Razor class library.



Configure your new project

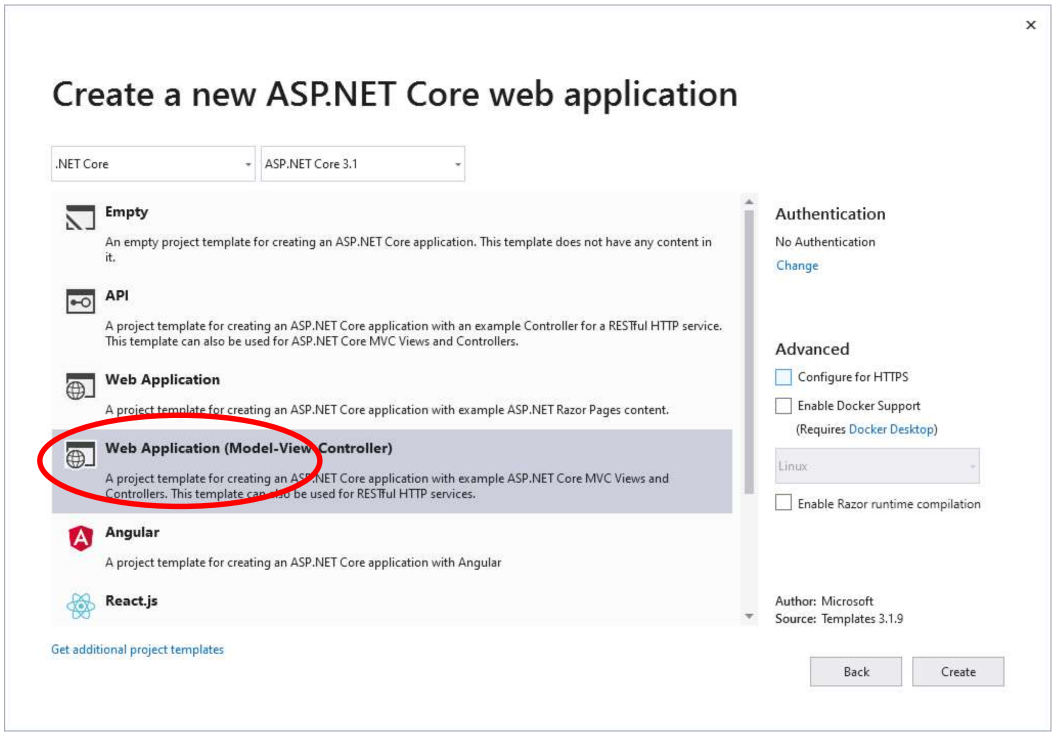
ASP.NET Core Web Application — C# Linux macOS Windows Cloud Service Web

Project name: XMLValidator

Location: C:\OneDrive\OneDrive - Algonquin College\Algonquin\21W\CST8259\Week 2\Lab 2\

Solution name: XMLValidator

[✓] Place solution and project in the same directory


Create a new ASP.NET Core web application

.NET Core | ASP.NET Core 3.1

Empty
An empty project template for creating an ASP.NET Core application. This template does not have any content in it.

API
A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core MVC Views and Controllers.

Web Application
A project template for creating an ASP.NET Core application with example ASP.NET Razor Pages content.

Web Application (Model-View-Controller)
A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.

Angular
A project template for creating an ASP.NET Core application with Angular

React.js

Authentication
No Authentication
Change

Advanced
Configure for HTTPS
Enable Docker Support
(Requires Docker Desktop)
Linux
Enable Razor runtime compilation

Author: Microsoft
Source: Templates 3.1.9

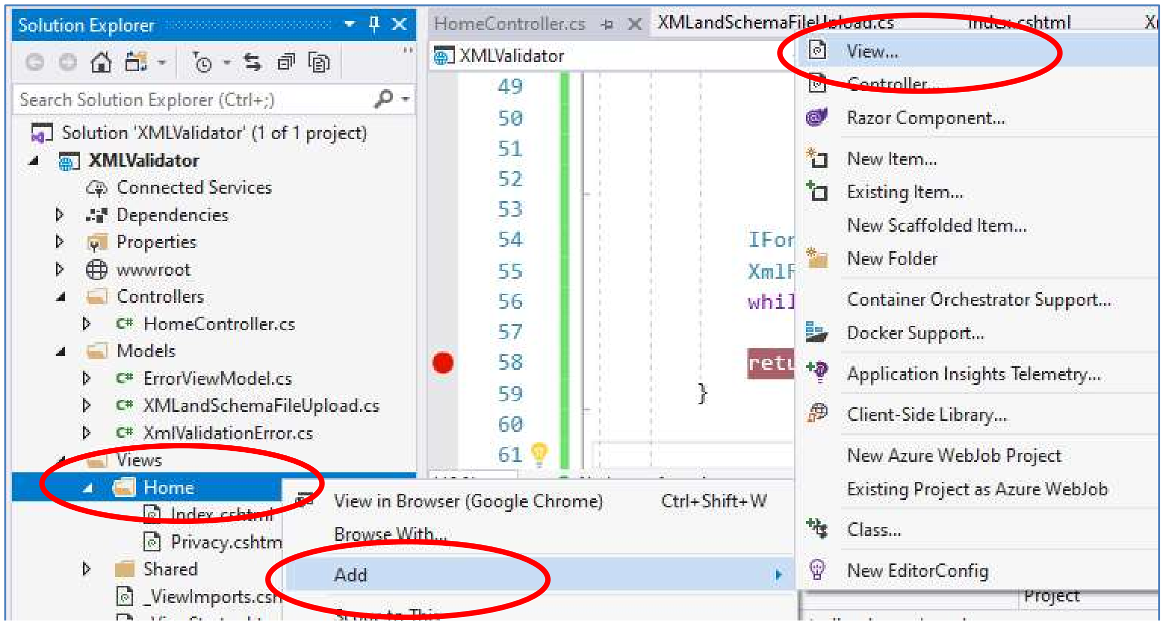
Get additional project templates

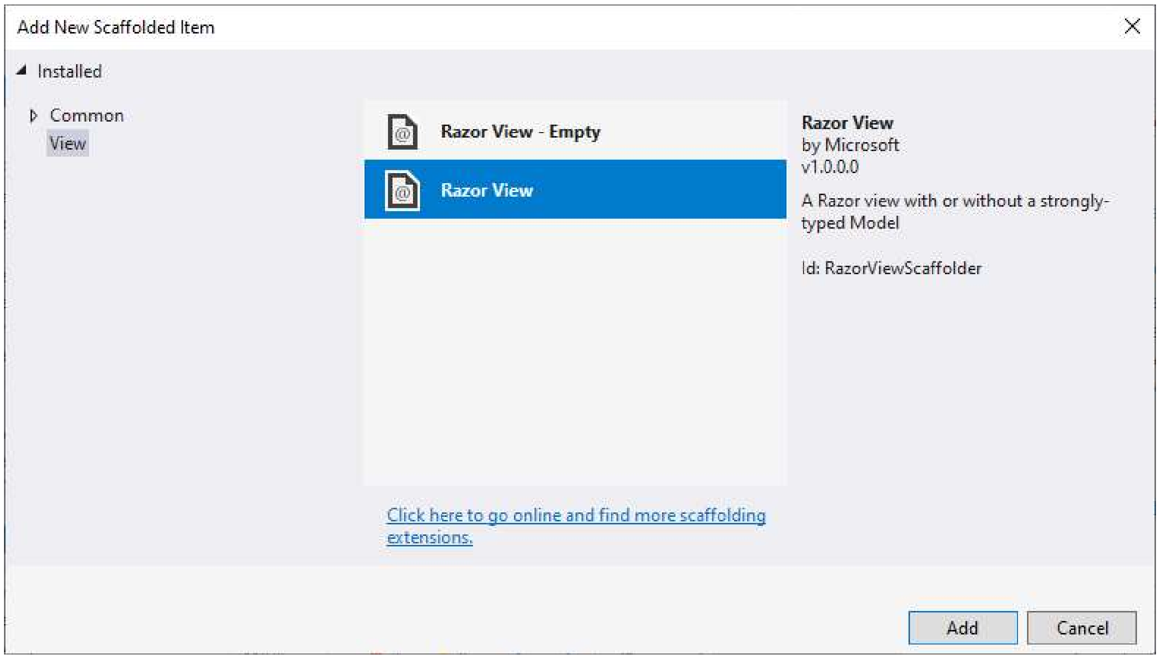
Back | Create

2. Following the steps below to create a view to display a list of error messages, you can show this view from your action method as:

```
return View("ValidationResult", validationResults);
```

Here **validationResults** is a list of `XmlValidationError` objects





Add New Scaffolded Item

Installed

Common

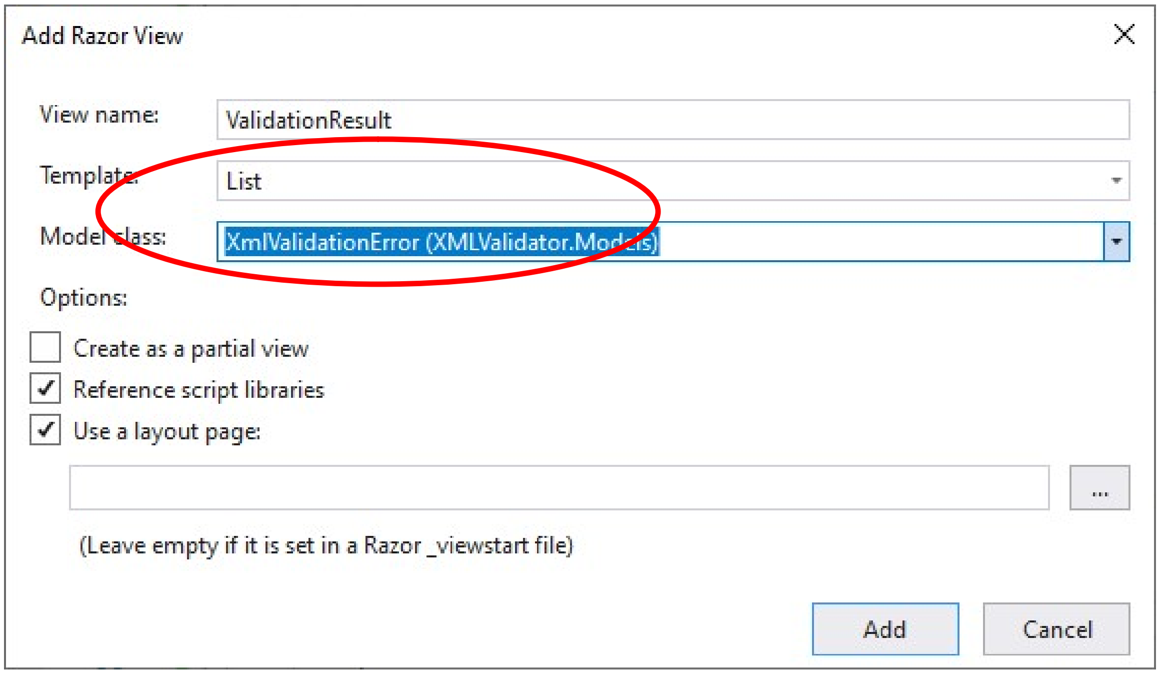
View

Razor View
by Microsoft
v1.0.0.0

A Razor view with or without a strongly-typed Model

Id: RazorViewScaffolder

Click here to go online and find more scaffolding extensions.



Add Razor View

View name: ValidationResult

Template: List

Model class: XmlValidationError (XMLValidator.Models)

Options:

- [ ] Create as a partial view
- [✓] Reference script libraries
- [✓] Use a layout page:

(Leave empty if it is set in a Razor _viewstart file)