# How to serialize and deserialize enumerations with Json.NET in ASP.NET Core

## Introduction

In default case, when serialize an object in json format, the enumerations always convert to an int type value, it is bad for readability.

This sample demonstrate how to serialize and deserialize enumerations with Json.NET.

## Sample prerequisites

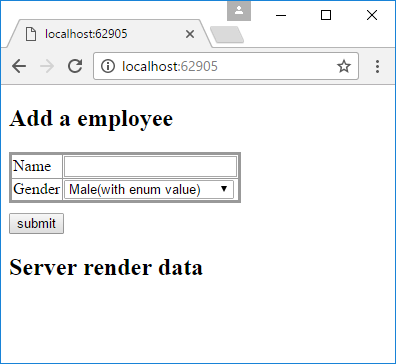
* Visual Studio 2017 or above. [[Visual Studio Home Page](https://www.visualstudio.com/)]
* Visual Studio enabled ASP.NET Core develop component.

## Building the sample

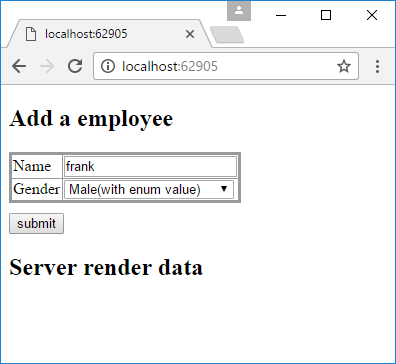
Use Visual Studio to open the sample solution, then press **F6 Key** to build the sample project.

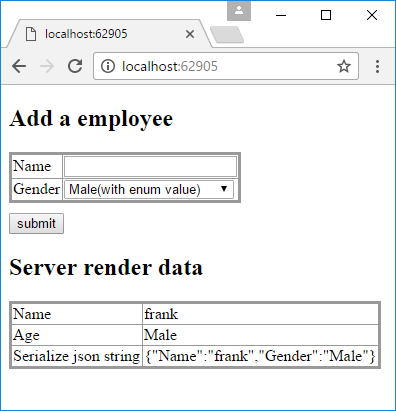
## Running the sample

* Use Visual Studio to open the sample solution, then press F5 Key or select **Debug -> Start Debugging** from the menu.
* When the project is running, you will see below page in browser.



* Type a name, and select a gender, then click **submit** button.





## Using the code

Enum

[JsonConverter(typeof(StringEnumConverter))]

public enum Gender

{

Male = 0,

Female = 1

}

Model

public class Employee

{

public string Name { get; set; }

public Gender Gender { get; set; }

}

Controller

public class HomeController : Controller

{

private static List<Employee> \_employees = new List<Employee>();

[HttpGet]

public IActionResult Index()

{

return View(\_employees);

}

[HttpPost]

public IActionResult Index(Employee employee)

{

if (employee != null)

{

\_employees.Add(employee);

}

return View(\_employees);

}

}

Razor

<form method="post">

<table>

<tr>

<td>Name</td>

<td><input type="text" name="Name" /></td>

</tr>

<tr>

<td>Gender</td>

<td>

<select name="Gender">

<option value="0">Male(with enum value)</option>

<option value="1">Female(with enum value)</option>

<option value="Male">Male(with enum text)</option>

<option value="Female">Female(with enum text)</option>

</select>

</td>

</tr>

</table>

<button id="submit">submit</button>

</form>

<h2>Server render data</h2>

@foreach (var item in Model)

{

<table>

<tr>

<td>Name</td>

<td>@item.Name</td>

</tr>

<tr>

<td>Age</td>

<td>@item.Gender</td>

</tr>

<tr>

<td>Serialize json string</td>

<td>@Newtonsoft.Json.JsonConvert.SerializeObject(item)</td>

</tr>

</table>

}

## More information

Serialization Attributes

<http://www.newtonsoft.com/json/help/html/SerializationAttributes.htm>