Xử lý form control trong Razor Pages

Khi làm việc với form bạn sẽ thường xuyên phải xử lý các điều khiển như checkbox, radio button, file upload, danh sách select. Các điều khiển này đòi hỏi những kỹ thuật riêng để xử lý.

Bài học sẽ hướng dẫn cách làm việc cơ bản với các điều khiển "cao cấp" này. Một số kỹ thuật khác như sử dụng tag helper hay html helper sẽ được xem xét trong những bài học riêng.

NỘI DUNG CỦA BÀI [Ẩn]

File upload

Tải file từ trình duyệt lên server là một công việc phức tạp hơn khá nhiều so với gửi một chuỗi văn bản thông thường qua text input. Tuy nhiên, Razor Pages đã hỗ trợ để quá trình tải file đơn giản hơn.

Hãy cùng thực hiện ví dụ.

Chuẩn bị: Tạo project mới theo mẫu Web Application.

Bước 1. Tạo thư mục `upload` trực thuộc project.

Bước 2. Tạo page Upload sử dụng model class.

Bước 3. Viết code như sau cho Upload.cshtml.cs:

```
using System;
using System.IO;

using Microsoft.AspNetCore.Hosting;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc.RazorPages;

namespace WebApplication1.Pages {
    public class UploadModel : PageModel {
        private readonly IWebHostEnvironment _environment;
        public UploadModel(IWebHostEnvironment environment) => _environment = environ

        public bool Success { get; private set; } = true;

        public void OnPost(IFormFile file) {
            try {
                var f = Path.Combine(_environment.ContentRootPath, "upload", file.Fil
                using var fs = new FileStream(f, FileMode.Create);
                file.CopyTo(fs);
                ViewData["file"] = file.FileName;
            }
            catch (Exception) {
                Success = false;
            }
        }
    }
}
```

Việc inject biến _environment chỉ để sử dụng được thư mục upload thuộc project. Bạn hoàn toàn có thể sử dụng một đường dẫn tuyệt đối tới một thư mục có sẵn.

> Mấu chốt của vấn đề khi upload file là tham số của OnPost tương ứng với file được tải lên phải có kiểu là `IFormFile`.

Bước 4. Viết mã razor cho Upload.cshtml như sau:

```
@page
@model WebApplication1.Pages.UploadModel
@{
    ViewData["Title"] = "Upload";
}

<h1>File Upload</h1>

<form method="post" enctype="multipart/form-data">
    <input type="file" name="file" />
    <input type="submit" value="Submit now" class="btn btn-info" />
</form>

@if (Request.Method == "POST") {
    @if (Model.Success) {
        <h3 class="text-success">File '@ViewData["file"]' uploaded successfully!</h3>
    }
    else {
        <h2 class="text-danger">Failed to upload file '@ViewData["file"]'!</h2>
    }
}
```

> Lưu ý form phải thiết lập enctype=”multipart/form-data” mới có thể upload file.

Làm việc với checkbox

Checkbox (hộp chọn) cho phép người dùng click chọn đồng thời nhiều mục. HTML sử dụng thẻ input để tạo checkbox cơ bản như sau:

```
<input type="checkbox" id="idEnglish" name="languages" value="English">
<label for="idEnglish">English</label>
```

Thẻ trên sẽ tạo ra checkbox như sau:

> Lưu ý:
> (1) Nhiều input cùng thuộc một nhóm phải có cùng giá trị name;
> (2) Giá trị của thuộc tính value sẽ được gửi về server nếu checkbox được chọn chứ không hiển thị cạnh checkbox;
> (3) Thẻ <label> hiển thị văn bản bên cạnh checkbox (trong khi value sẽ không hiển thị);
> (4) Checkbox sẽ được chọn mặc định nếu có mặt thuộc tính checked (không quan tâm đến giá trị).

Hãy cùng thực hiện ví dụ sau:

Bước 1. Tạo trang Checkbox.cshtml.

Bước 2. Viết code cho Checkbox.cshtml.cs (model class) như sau:

```
using System.Collections.Generic;
using Microsoft.AspNetCore.Mvc.RazorPages;

namespace WebApplication1.Pages {
    public class CheckboxModel : PageModel {
        public List<string> Languages { get; private set; }

        public void OnPost(List<string> languages) => Languages = languages;
    }
}
```

Để ý rằng chúng ta sử dụng `List<string>` để chứa những gì người dùng gửi lại thông qua tích đánh dấu các checkbox.

Bước 3. Viết code cho Checkbox.cshtml như sau:

```
@page
@model WebApplication1.Pages.CheckboxModel
@{
    ViewData["Title"] = "Checkbox";
}

@if (Request.Method.ToUpper() == "GET") {
<h3>Which languages can you speak?</h3>
<form method="post">
    <!-- Default unchecked -->
    <div class="custom-control custom-checkbox">
        <input type="checkbox" class="custom-control-input" id="Arabic" name="languag
        <label class="custom-control-label" for="Arabic">Arabic</label>
    </div>
    <div class="custom-control custom-checkbox">
        <input type="checkbox" class="custom-control-input" id="Chinese" name="langua
        <label class="custom-control-label" for="Chinese">Chinese</label>
    </div>
    <!-- Default checked -->
    <div class="custom-control custom-checkbox">
        <input type="checkbox" class="custom-control-input" id="English" name="langua
        <label class="custom-control-label" for="English">English</label>
    </div>
    <div class="custom-control custom-checkbox">
        <input type="checkbox" class="custom-control-input" id="French" name="languag
        <label class="custom-control-label" for="French">French</label>
    </div>
    <div class="custom-control custom-checkbox">
        <input type="checkbox" class="custom-control-input" id="Russian" name="langua
        <label class="custom-control-label" for="Russian">Russian</label>
    </div>
    <div class="custom-control custom-checkbox">
        <input type="checkbox" class="custom-control-input" id="Spanish" name="langua
        <label class="custom-control-label" for="Spanish">Spanish</label>
    </div>
    <div>
        <button class="btn btn-success" type="submit">Submit</button>
        <button class="btn btn-secondary" type="reset">Clear</button>
    </div>
</form>
}
else if(Request.Method.ToUpper() == "POST") {
    <h3 class="alert-success">Great! You can speak: @foreach (var l in Model.Language
}
```

Để ý rằng tất cả thẻ input trên đều có cùng `name="languages"` cũng trùng tên với tham số của `OnPost(List<string> languages)`.

Chạy chương trình và bạn thu được kết quả như sau:

Để ý rằng code ở trên có nhiều chỗ lặp lại mà bạn có thể sử dụng Razor code thay thế.

Do checkbox thường là dạng danh sách, bạn có thể sử dụng vòng lặp để xuất ra HTML. Bạn có thể cải tiến ví dụ trên sử dụng cú pháp Razor cho ngắn gọn như sau:

Checkbox.cshtml.cs | Checkbox.cshtml

```
using System.Collections.Generic;

using Microsoft.AspNetCore.Mvc.RazorPages;

namespace WebApplication1.Pages {
    public class CheckboxModel : PageModel {
        public List<string> Languages { get; private set; }
        public string[] OfficialLanguages => new[] {
            "Arabic",
            "Chinese",
            "English",
            "French",
            "Russian",
            "Spanish"
        };

        public void OnPost(List<string> languages) => Languages = languages;
    }
}
```

Lưu ý: Razor Pages đọc các giá trị gửi về có cùng tên languages và ghép chúng vào object thuộc kiểu IEnumerable. Vì vậy, bạn cũng có thể truy xuất giá trị của nhóm checkbox trên như sau:

```
IEnumerable<string> languages = Request.Form["languages"];
```

Từ đây bạn có thể chuyển đổi về kiểu dữ liệu danh sách: `Languages = languages.ToList();`

Sử dụng radio button

Radio button, còn gọi là nút đài, có hình thức gần tương tự như checkbox, nhưng chỉ cho phép chọn một phương án duy nhất.

Trong Razor Pages, nút đài có cách xuất (ra html) tương tự như checkbox nhưng có cách đọc dữ liệu từ form giống như textbox hay textarea bạn đã biết trong bài học trước.

Radio button được tạo ra trên form với thẻ input, tương tự như checkbox, nhưng thuộc tính `type="radio"` :

```
<input type="radio" />
```

Hãy cùng thực hiện một ví dụ:

Bước 1. Thêm trang RadioButton vào dự án.

Bước 2. Viết code cho model class trong file RadioButton.cshtml.cs như sau:

```
using Microsoft.AspNetCore.Mvc.RazorPages;

namespace WebApplication1.Pages {
    public class RadioButtonModel : PageModel {
        public string Language { get; private set; }
        public string[] OfficialLanguages => new[] {
            "Arabic",
            "Chinese",
            "English",
            "French",
            "Russian",
            "Spanish"
        };

        public void OnPost(string language) => Language = language;

        //public void OnPost() => Language = Request.Form["language"];
    }
}
```

Bạn có thể thấy cách truy xuất giá trị của nút đài từ form thực hiện thông qua tên, tương tự như các giá trị đơn của textbox hay textarea.

Bước 3. Viết code razor cho RadioButton.cshtml như sau:

```
@page
@model WebApplication1.Pages.RadioButtonModel
@{
    ViewData["Title"] = "Radio Button";
}

@if (Request.Method.ToUpper() == "GET") {
    <h3>What is your native language?</h3>

    <form method="post">
        @foreach (var l in Model.OfficialLanguages) {
            <div class="custom-control custom-radio">
                <input type="radio" class="custom-control-input" id="@l" name="langua
                <label class="custom-control-label" for="@l">@l</label>
            </div>
        }
        <div class="form-group">
            <button class="btn btn-success" type="submit">Submit</button>
            <button class="btn btn-secondary" type="reset">Clear</button>
        </div>
    </form>
}
else if (Request.Method.ToUpper() == "POST") {
    <h3 class="alert-success">Your native language is <b>@Model.Language</b></h3>
}
```

Bạn có thể thấy rằng việc xuất nút đài ra form giống hệt như đối với checkbox. Khác biệt duy nhất là thuộc tính `type="radio"`.

Kết quả thực hiện chương trình như sau:

> Lưu ý khi sử dụng nút đài:
> (1) Các input cùng thuộc một nhóm phải có cùng giá trị name;
> (2) Giá trị của thuộc tính value sẽ được gửi về server nếu nút đài được chọn;
> (3) Thẻ <label> hiển thị văn bản bên cạnh nút (value thì không hiển thị);
> (4) Nút sẽ được chọn mặc định nếu có mặt thuộc tính checked (không quan tâm đến giá trị).

Sử dụng Combo Box

Combo Box cho phép bạn chọn một giá trị từ một danh sách. Trong HTML, combo box được tạo ra bằng thẻ <select></select>. Mỗi phương án chọn được tạo bằng thẻ <option /> nằm bên trong <select></select>.

Ví dụ

```
<select class="browser-default custom-select" name="country">
    <option value="Not specified" selected>Choose a country ...</option>
    <option value="US">United States of America</option>
    <option value="China">People Republic of China</option>
    <option value="Japan">Japan</option>
    <option value="Germany">Federal Republic of Germany</option>
</select>
```

Sẽ tạo ra combo box

Thuộc tính selected quyết định xem option nào sẽ được chọn ngay từ đầu (giá trị mặc định).

Khi bạn chọn một phần tử và submit form, dữ liệu từ thuộc tính value của option tương ứng sẽ trả về server. Trong ví dụ trên, giả sử bạn chọn mục "United States of America" thì giá trị trả về là US.

Giá trị trả về server được truy xuất qua handler hoặc object Request qua tên tương ứng với thuộc tính name của thẻ select. Như trong ví dụ trên, bạn truy xuất giá trị thông qua tham số country của `OnPost(string country)` hoặc qua `Request.Form["country"]`.

Hãy cùng thực hiện một ví dụ nhỏ.

Bước 1. Tạo trang mới ComboBox.

Bước 2. Viết code cho model class trong ComboBox.cshtml.cs như sau:

```
using Microsoft.AspNetCore.Mvc.RazorPages;

namespace WebApplication1.Pages {
    public class ComboBoxModel : PageModel {
        public string Country { get; private set; }
        public void OnPost(string country) => Country = country;
        //public void OnPost() => Country = Request.Form["country"];
    }
}
```

Bước 3. Viết code cho razor page ComboBox.cshtml như sau:

```
@page
@model WebApplication1.Pages.ComboBoxModel
@{
    ViewData["Title"] = "ComboBox";
    var method = Request.Method.ToUpper();
}

@if (method == "GET") {
    <form method="post" class="border border-light p-5">
        <h3>What is your country of birth?</h3>
```

```
            <div>
                <select class="browser-default custom-select" name="country">
                    <option value="Not specified" selected>Choose a country ...</option>
                    <option value="US">United States of America</option>
                    <option value="China">People Republic of China</option>
                    <option value="Japan">Japan</option>
                    <option value="Germany">Federal Republic of Germany</option>
                </select>
            </div>
            <button class="btn btn-success btn-block my-2" type="submit">Submit</button>
        </form>
    }
    else if (method == "POST") {
        <h3>You were born in @Model.Country</h3>
    }
```

Bạn thu được kết quả như sau:

Nếu ưa thích xử lý code thay cho viết thẻ HTML, bạn có thể sử dụng phương án sau:

ComboBox.cshtml.cs | ComboBox.cshtml

```
using Microsoft.AspNetCore.Mvc.RazorPages;

namespace WebApplication1.Pages {
    public class ComboBoxModel : PageModel {
        public string Country { get; private set; }
        public (string value, string display)[] Countries => new[] {
            ("US", "United States of America"),
            ("China", "People Republic of China"),
            ("Japan", "Japan"),
            ("Germany", "Federal Republic of Germany"),
        };
        public void OnPost(string country) => Country = country;
        //public void OnPost() => Country = Request.Form["country"];
    }
}
```

Sử dụng list box

List box là một dạng khác của danh sách chọn, trong đó bạn có thể chọn nhiều mục từ một danh sách.

Trong HTML, list box được tạo ra giống hệt như combo box, khác biệt duy nhất là trong thẻ <select> cần có thêm thuộc tính multiple (không cần giá trị):

```
<select name="countries" multiple>
    <option value="US">United States of America</option>
    <option value="China">People Republic of China</option>
    <option value="Japan">Japan</option>
    <option value="Germany">Federal Republic of Germany</option>
</select>
```

Tuy cách tạo ra listbox tương tự như combobox nhưng cách xử lý listbox tại server lại giống hệt như đối với checkbox (do cùng trả về nhiều giá trị qua cùng một tên).

Hãy cùng thực hiện ví dụ sau:

Bước 1. Tạo trang mới ListBox

Bước 2. Viết code cho model class trong ListBox.cshtml.cs như sau:

```
using System.Collections.Generic;
using System.Linq;
using Microsoft.AspNetCore.Mvc.RazorPages;

namespace WebApplication1.Pages {
    public class ListBoxModel : PageModel {
        public (string value, string display)[] Countries => new[] {
            ("US", "United States of America"),
            ("China", "People Republic of China"),
            ("Japan", "Japan"),
            ("Germany", "Federal Republic of Germany"),
        };
        public string[] VisitedCountries { get; private set; }

        public void OnPost() {
            var countries = Request.Form["countries"];
            VisitedCountries = countries.ToArray();
        }
    }
}
```

Bước 3. Viết mã razor cho ListBox.cshtml như sau:

```
@page
@model WebApplication1.Pages.ListBoxModel
@{
    ViewData["Title"] = "List Box";
    var method = Request.Method.ToUpper();
}

@if (method == "GET") {
    <form method="post" class="border border-light p-3">
        <h3>Which country did you visit?</h3>
        <div>
            <select class="browser-default custom-select" name="countries" multiple>
                @foreach (var p in Model.Countries) {
                    <option value="@p.value">@p.display</option>
                }
            </select>
        </div>
        <button class="btn btn-success btn-block my-2" type="submit">Submit</button>
    </form>
}
else if (method == "POST") {
    <h3>You have been to @foreach (var c in Model.VisitedCountries) {<strong>@c </str
}
```

Kết quả chạy chương trình:

Kết luận

Bài học đã hướng dẫn những kỹ thuật làm việc cơ bản với các điều khiển nâng cao, bao gồm file upload, checkbox, radio button, list.

Lưu ý rằng, đây chỉ là những kỹ thuật cơ bản nhất giúp bạn hiểu cách thức xử lý chúng trong Razor Pages. Ngoài ra còn những kỹ thuật "cao cấp" hơn như tag helper, html helper và model binding sẽ được hướng dẫn trong những bài học riêng.

Tải mã nguồn Razor Form

1 file(s) 0.00 KB

TẢI MÃ NGUỒN

+ Nếu bạn thấy site hữu ích, trước khi rời đi hãy **giúp đỡ** site bằng một hành động nhỏ để site có thể phát triển và phục vụ bạn tốt hơn.
+ Nếu bạn thấy bài viết hữu ích, hãy giúp **chia sẻ** tới mọi người.
+ Nếu có thắc mắc hoặc cần trao đổi thêm, mời bạn viết trong phần **thảo luận** cuối trang.
Cảm ơn bạn!