**Ajax là gì : Ajax post là truyền dữ liệu (Object ) từ Ajax (trong View) lên Controller /Action mà không cần reload lại trang kết hợp với form**

# **⭐ TÓM TẮT NGẮN GỌN NHẤT**

|  |  |  |
| --- | --- | --- |
| **Loại POST** | **Binding** | **Validation** |
| Form POST | Chỉ bind field trong form | ModelState.IsValid (Validate) tất cả Property trong Models , nhưng bỏ qua navigation Property |
|  |  |  |

**Sử dụng Ajax post kết hợp với form post để truyền các dữ liệu nguyên thủy lên Controller/ Action xử lí mà không bị reload lại trang**

**Th1 : Ajax trả về kiểu dữ liệu nguyên thủy**

**Controllers**

[HttpPost]

public async Task<IActionResult> Edit( int Status, int Id )

{

var productId = await \_context.Products.FindAsync(Id);

if (productId == null)

{

return Json(new { success = false }); // tự động convert kiểu dữ liệu nguyên thủy sang Object json

}

else

{

productId.Status = Status;

\_context.Products.Update(productId);

await \_context.SaveChangesAsync();

return Json(new { success = true });

}

}

* Convert : Boolean success = true sang Object json

{

" success ": true

}

**Views**

@model Check.Models.Product

<h2>Nhập Status</h2>

<form id="formCreate">

<div class="w-25">

<select class="form-control" id="Status">

<option value="1">Đơn hàng mới</option>

<option value="0">Đã xử lý</option>

</select>

</div>

<div >

<input type="hidden" class="getId" value="5" />

</div>

<button id="btnSave" type="button">Submit</button>

</form>

@section Scripts {

<script>

$('#btnSave').click(function (e) {

e.preventDefault();

var status = $('#Status').val(); // Nó sẽ lấy value của option đang được chọn trong Select

var id = $('.getId').val(); //Nó sẽ lấy value trong thẻ input

$.ajax({

url: '/Item/Edit', // Controller/Action

type: 'POST',

dataType: "json",

data: { Status: status, Id: id }, // gửi status và id lên server

success: function (res) {

if (res.success) { // gủi lên Server thành công

Swal.fire("Thêm thành công!"); // sử dụng Thông báo Sweet Alert

} else {

Swal.fire("Thêm không thành công!");

}

},

error: function () { // gủi lên Server thất bại

Swal.fire("Có lỗi xảy ra!");

}

});

});

</script>

}

**Th2 : Ajax trả về kiểu dữ liệu Tham chiếu (Object)**

Models

public class UserInfo

{

public int Id { get; set; }

public string Name { get; set; }

public bool IsActive { get; set; }

}

Controllers

namespace Check.Controllers

{

public class TestController : Controller

{

[HttpPost]

public IActionResult Index1( )

{

UserInfo user = new UserInfo()

{

Id = 1,

Name = "Dat",

IsActive = true

};

return Json(new { data = user }); // tự động convert kiểu dữ liệu tham chiếu (Object) sang json Object

}

}

}

* Convert : var user sang Object json

{

"data": {

"id": 1,

"name": "Dat",

"isActive": true

}

}

Views

<button id="btnSave" type="button">Submit</button>

@section Scripts {

<script>

$('#btnSave').click(function (e) {

e.preventDefault();

$.ajax({

url: '/Test/Index1',

type: 'POST',

dataType: "json",

success: function (result) {

alert(

result.id + " " +

result.name + " " +

result.isActive

);

}

});

});

</script>

}

**Th3 : Ajax trả về kiểu dữ liệu Tham chiếu (Object) và kiểu dữ liệu nguyên thủy**

Controllers

**public class TestController : Controller**

**{**

[HttpPost]

public IActionResult Index1( )

{

UserInfo user = new UserInfo

{

Id = 1,

Name = "Dat",

IsActive = true

};

return Json(new

{

success = true, // primitive

data = user // object

});

}

}

* Convert : var data và success sang Object json

{

"success": true,

"data": {

"id": 1,

"name": "Dat",

"isActive": true

}

}

Views

<button id="btnSave" type="button">Submit</button>

@section Scripts {

<script>

$('#btnSave').click(function (e) {

e.preventDefault();

$.ajax({

url: '/Test/Index1',

type: 'POST',

dataType: 'json', // ⭐ BẮT BUỘC

success: function (result) {

alert(

result.success + " " +

result.data.id + " " +

result.data.name + " " +

result.data.isActive

);

}

});

});

</script>

}

TH4 : **: Ajax trả về kiểu dữ liệu tập hợp các tham chiếu (List <T> )**

public class TestController : Controller

{

[HttpPost]

public IActionResult Index1( )

{

var users = new List<UserInfo>

{

new UserInfo { Id = 1, Name = "A", IsActive = true },

new UserInfo { Id = 2, Name = "B", IsActive = false },

new UserInfo { Id = 3, Name = "C", IsActive = false },

new UserInfo { Id = 4, Name = "D", IsActive = false }

};

return Json(new { data = users });

}

}

* Convert : var users sang Object Json

{

"data": [

{ "id": 1, "name": "A", "isActive": true },

{ "id": 2, "name": "B", "isActive": false },

{ "id": 3, "name": "C", "isActive": false },

{ "id": 4, "name": "D", "isActive": false }

]

}

* var Object = new

{

data = list // Mảng của bạn nằm ở đây

};

Viết đầy đủ của new { data = list } // Mảng của bạn nằm ở đây

Views

<button id="btnSave" type="button">Submit</button>

@section Scripts {

<script>

$('#btnSave').click(function (e) {

e.preventDefault();

$.ajax({

url: '/Test/Index1',

type: 'POST',

dataType: 'json', // ⭐ BẮT BUỘC

success: function (result) {

// dùng map

const data = result.data.map(u => ({

Id: u. Id,

Name: u. Name ,

IsActive : u. IsActive

}));

// dùng foreach

data.forEach(u => {

alert( u.Id + “ “ + u.Name + “ “ + u. IsActive)

});

}

});

});

</script>

}

Note : những điểm chú ý

dataType: "json" :nó sẽ parse ( JSON.parse() ) tự động Json Object thành Javascript Object

JSON Object

{

"success": true,

"data": {

"id": 1,

"name": "Dat",

"isActive": true

}

}

* Sau khi JSON.parse() hoặc dùng jQuery/Axios với dataType: "json" → JS Object:

result = {

success: true,

data: {

id: 1,

name: "Dat",

isActive: true

}

};

console.log(result.success);      // true

console.log(result.data.id);      // 1

console.log(result.data.name);    // "Dat"

console.log(result.data.isActive);// true

\* Đây là **JSON Object**  
vì có dạng

{

“Key” : value

}

* Sau new thì phải có Đối tượng cụ thể để tăng tính tường mình