

Bài thực hành 03

Tạo ứng dụng Client để gọi API

MỤC TIÊU

- Cấu hình Cross-Origin Resource Sharing
- Sử dụng các Api thông qua jQuery Ajax, Angular app

CHÚ Ý: LÀM TIẾP TỪ LAB 11-12

Bài 1: Cấu hình Cross-Origin Resource Sharing

Bước 1: Mở lại dự án Api đã tạo bài trước, mở file **Startup** lên tìm đến phương thức **ConfigureServices** và thêm code sau, xem hình

```
// This method gets called by the runtime. Use this method to add services to the container.
public void ConfigureServices(IServiceCollection services)
{
    services.AddMvc().SetCompatibilityVersion(CompatibilityVersion.Version_2_1);
    // Cấu hình kết nối Database
    services.AddDbContext<AppDbContext>(op => op.UseSqlServer(Configuration.GetConnectionString("DefaultConnection")));

    services.AddCors(options =>
    {
        options.AddPolicy("AllowOrigin", builder => builder.AllowAnyOrigin()
            .AllowAnyMethod()
            .AllowAnyHeader()
            .AllowCredentials());
    });
}
```

Bước 2 Tiếp theo tìm tới phương thức Configure và thêm code sau

```
// This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
public void Configure(IApplicationBuilder app, IHostingEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }
    else
    {
        app.UseHsts();
    }

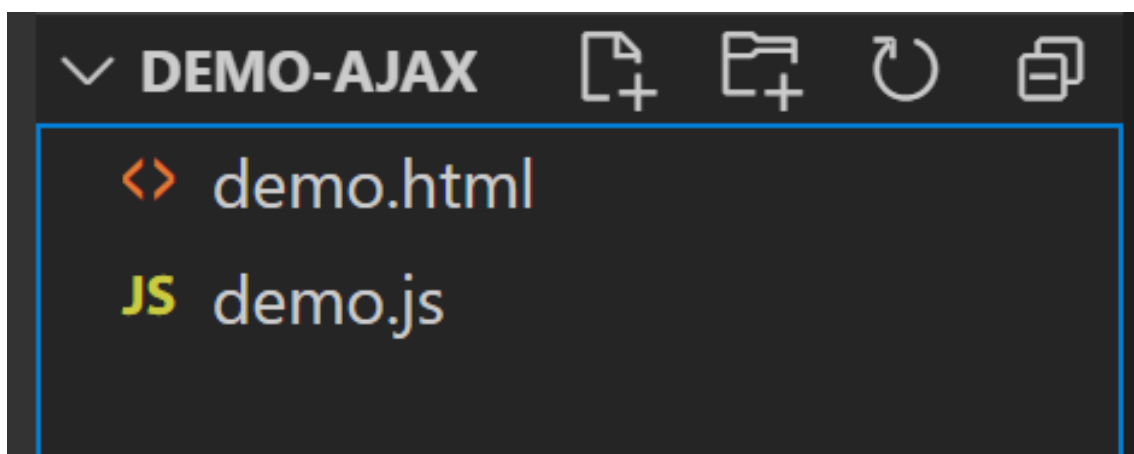
    app.UseCors("AllowOrigin");
    app.UseHttpsRedirection();
    app.UseMvc();
}
```

Build lại dự án và nhớ khởi chạy ứng dụng

Bài 2: Sử dụng Ajax để gọi api hiển thị danh sách

Bước 1: Tạo một thư mục mới đặt tên là **demo-ajax** ở đâu đó trong máy của bạn

Sau đó tạo file demo.html và file **demo.js** trong thư mục demo-ajax vừa tạo
Cấu trúc thư mục demo-ajax có dạng



Bước 2: Mở file demo.html và soạn code có dạng sau

(Sử dụng phím tắt bs3-template:html5)

```
<!DOCTYPE html>
<html lang="">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Title Page</title>
    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" /
  >
</head>
<body>
  nội dung giao diện sẽ soạn thảo tại đây

  <!-- jQuery -->
  <script src="https://code.jquery.com/jquery.min.js"></script>
  <!-- Bootstrap JavaScript -->
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
  <script src="demo.js"></script>
</body>
</html>
</html>
```

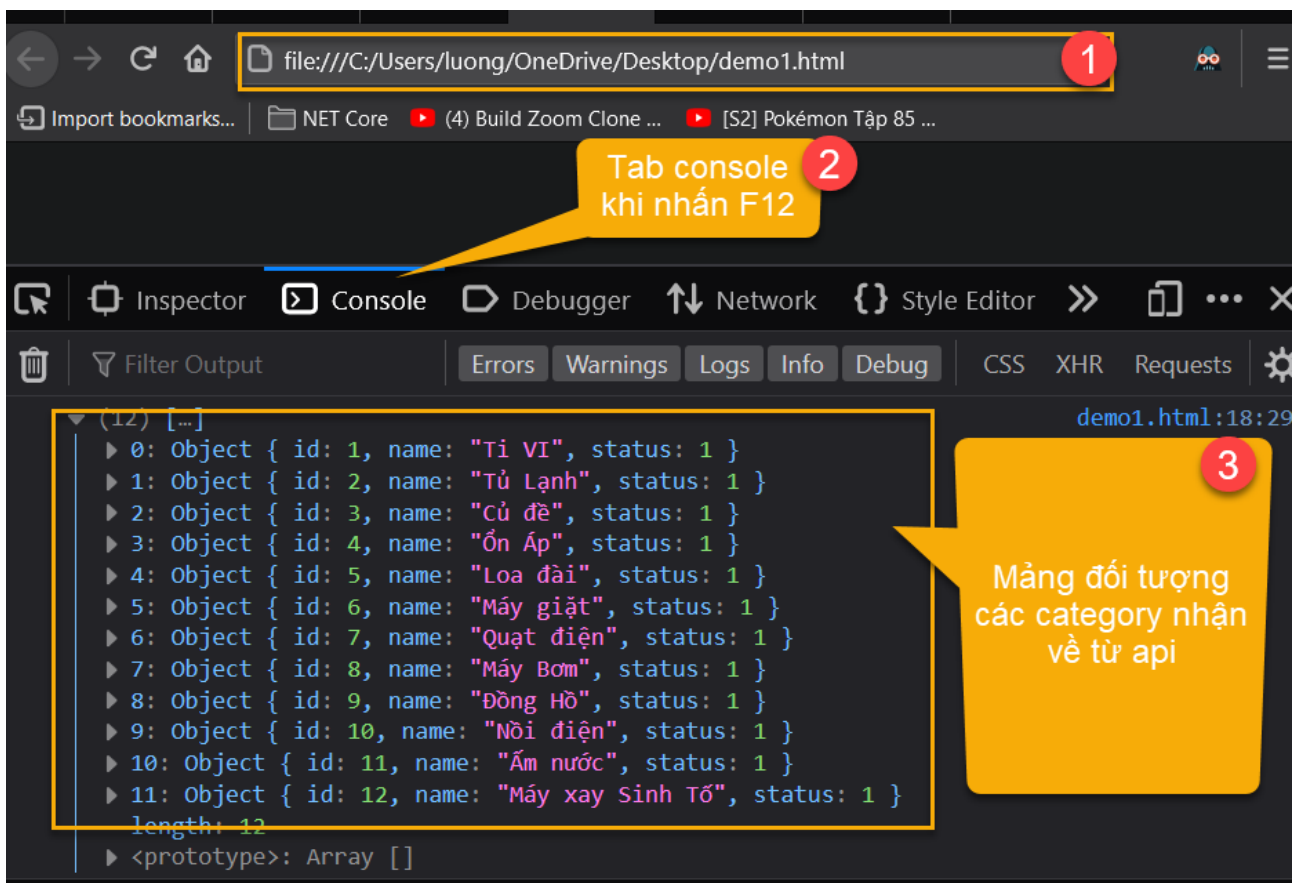
Bước 3: Mở file demo.js soạn code jQuery Ajax để gọi api như sau

```
$(document).ready(function(){
  // Phương thức GET => Lấy về danh sách danh mục
  $.get('https://localhost:44341/api/categories', function(res){
    console.log(res)
  })
})
```

Chú ý: Như lab 01 đã tạo các api

Link	Phương thức	Trả về
https://localhost:44341/api/categories	GET	Danh sách
https://localhost:44341/api/categories/{id}	GET	Chi tiết
https://localhost:44341/api/categories	POST	Thêm mới
https://localhost:44341/api/categories/{id}	PUT	Cập nhật
https://localhost:44341/api/categories/{id}	DELETE	Xóa bỏ

Bước 4: mở file **demo.html** trên trình duyệt như bình thường và nhấn F12 vào tab console xem kết quả có dạng như sau



Bước 4: Hiển thị dữ liệu từ Ajax lên giao diện Html

Mở lại file **demo.html**, nhúng thư css, và js của bootstrap vào, sau đó tạo bảng html

Code tham khảo

```
<body>
  <div class="container">
    <h1>Danh mục</h1>
    <table class="table table-bordered table-hover">
      <thead>
        <tr>
          <th>ID</th>
          <th>Tên Danh mục</th>
          <th>Trạng thái</th>
          <th>Lựa chọn</th>
        </tr>
      </thead>
      <tbody id="list">
        <!-- Dữ liệu load vào đây -->
      </tbody>
    </table>
  </div>
  <!-- jQuery -->
  <script src="https://code.jquery.com/jquery.min.js"></script>
  <!-- Bootstrap JavaScript -->
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
  <script src="demo.js"></script>
</body>
```

Bước 5: Mở lại demo.js sử dụng vòng lặp for để duyệt dữ liệu và đổ lên giao diện

```
$(document).ready(function(){
  // Phương thức GET => Lấy về danh sách danh mục
  $.get('https://localhost:44341/api/categories', function(res){
    var html = ''; // biến để chứa cấu trúc tr, td của bảng
    for (let i = 0; i < res.length; i++) {
      var cat = res[i];
      var status = cat.status == 1 ? 'Hiển thị' : 'Đang ẩn';
      html += '<tr>';
      html += '<td>' + cat.id + '</td>';
      html += '<td>' + cat.name + '</td>';
      html += '<td>' + status + '</td>';
      html += '<td>';
      html += ' <button type="button" data-id="' + cat.id + '" class="btn btn-xs btn-';
      html += 'primary show-detail">Xem</button>';
      html += ' <button type="button" data-id="' + cat.id + '" class="btn btn-xs btn-danger';
      html += ' delete">Xóa</button>';
      html += '</td>';
      html += '</tr>';
    }
    // gán nội dung của biến html vào thẻ có id="list"
    $('#list').html(html);
  })
})
```

Trong đoạn code trên mục đích duyệt mảng các đối tượng và cứ mỗi lần duyệt cộng chuỗi một cấu trúc dòng và cột của bảng html

Chú ý: có đoạn button có thuộc tính **data-id** để chuẩn bị dữ liệu id cho các sự click xem và xóa



Bước 6: F5 để xem kết quả có dạng (Phải có dữ liệu trong CSDL để API trả về cho client thì mới có danh sách có dạng như sau)



Danh mục

ID	Tên Danh mục	Trạng thái	Lựa chọn
1	Ti VI	Hiện thị	<button>Xem</button> <button>Xóa</button>
2	Tủ Lạnh	Hiện thị	<button>Xem</button> <button>Xóa</button>
3	Củ dề	Hiện thị	<button>Xem</button> <button>Xóa</button>
4	Ôn Áp	Hiện thị	<button>Xem</button> <button>Xóa</button>
5	Loa đài	Hiện thị	<button>Xem</button> <button>Xóa</button>
6	Máy giặt	Hiện thị	<button>Xem</button> <button>Xóa</button>

Bài 3: Sử dụng Ajax gọi api hiển thị chi tiết một bản ghi

Bước 1: Mở lại file demo.html, thêm thành phần modal của bootstrap 3 (Chú ý lệnh gõ tắt để tạo modal là bs3-modal)

Code tham khảo

```
<body>
  <!-- Code phần danh sách cứ ữu nguyên ở đây -->
  <!-- Thêm Modal ngay dưới phần danh sách-->
  <div class="modal fade" id="modal-id">
    <div class="modal-dialog">
      <div class="modal-content">
        <div class="modal-header">
          <button type="button" class="close" data-dismiss="modal" aria-
hidden="true">&times;</button>
          <h4 class="modal-title">Thông tin chi tiết</h4>
        </div>
        <div class="modal-body" id="detail">
          <!-- nội dung load từ ajax sẽ đổ vào đây -->
        </div>
      </div>
    </div>
  </div>
  <!-- jQuery -->
  <script src="https://code.jquery.com/jquery.min.js"></script>
  <!-- Bootstrap JavaScript -->
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
  <!-- Cú pháp jquery Ajax gọi api -->
  <script src="demo.js"></script>
</body>
```

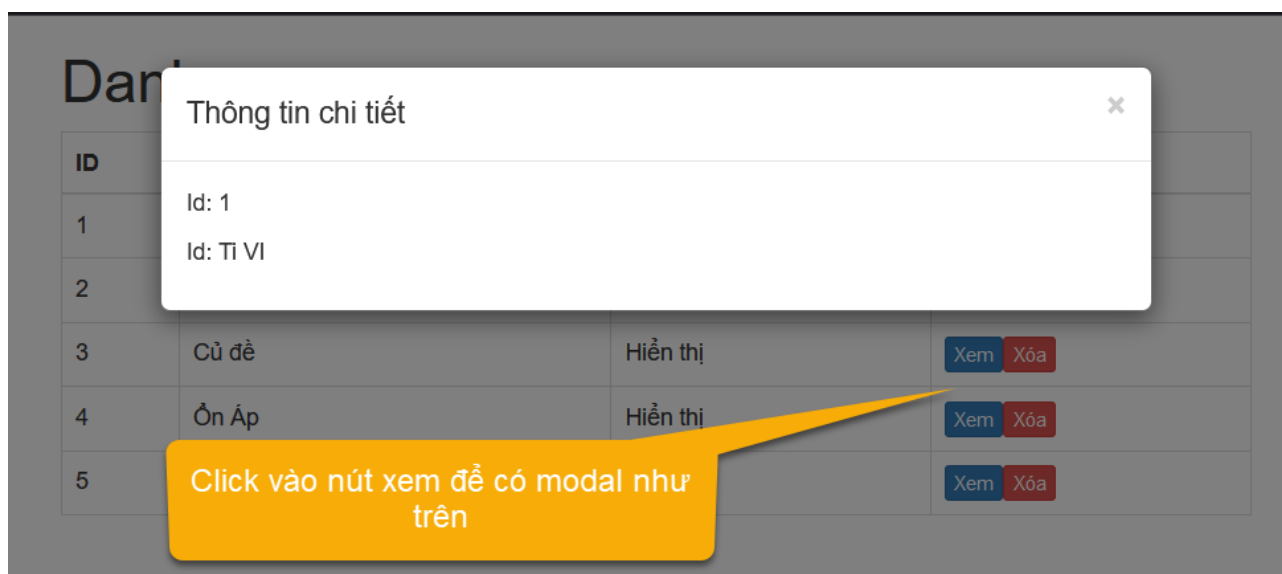
Bước 2: Mở file demo.js và thêm code bắt sự kiện click trên nút bấm có class show-detail tại table danh sách

Code tham khảo

```
$(document).ready(function(){
    // code phần hiển thị danh sách cứ để nguyên ở đây

    // Sự kiện để lấy ra chi tiết
    $(document).on('click', '.show-detail', function(){
        var id = $(this).data('id'); // lấy giá trị từ thuộc tính data-id của thẻ button
        // Phương thức GET => Lấy về chi tiết danh mục
        $.get('https://localhost:44341/api/categories/'+id, function(res){
            var _detail = '<p>Id: ' + res.id + '</p>';
            _detail += '<p>Id: ' + res.name + '</p>';
            $('#detail').html(_detail); // Gán giá trị vào cho phần tử có id="detail"
            $('#modal-id').modal('show'); // Nổi modal lên
        });
    });
});
```

Bước 3: F5 lại trình duyệt và click lên nút xem sẽ nổi modal với chi tiết của danh mục



Bài 4: Gọi api xóa dữ liệu theo id thông qua ajax với kiểu DELETE

Bước 1: Mở lại file demo.js, code bắt sự kiện cho nút xóa (Nút bấm class là delete)

```
// Sự kiện để gọi api xóa dữ liệu
$(document).on('click', '.delete', function(){
    var _this = $(this); // lưu lại phần tử hiện đang click
    var id = $(this).data('id'); // lấy giá trị từ thuộc tính data-id của thẻ button
    // Gọi ajax
    $.ajax({
        url: 'https://localhost:44341/api/categories/'+id,
        type: 'DELETE', // Phương thức Delete => gọi api xóa danh mục
        success: function(res){
            _this.closest('tr').remove(); // loại bỏ thẻ tr của dòng dữ liệu bị xóa
        }
    });
});
```

Bước 2: F5 lại dự án và click vào nút xóa để test

Bài 5: Tạo form thêm mới dữ liệu sử dụng Ajax gọi api thêm mới

Bước 1: Mở lại file **demo.html** và tạo form thêm mới ngay trên phần danh sách

Code form tham khảo

```
<div class="container">
  <h1>Thêm mới danh mục</h1>
  <form action="" method="POST" class="form-inline" id="form-add">
    <div class="form-group">
      <label>Tên danh mục</label>
      <input class="form-control" id="name">
    </div>
    <div class="form-group">
      <label>Trạng thái</label>
      <input type="number" class="form-control" id="status">
    </div>
    <button type="button" class="btn btn-primary add-new">Thêm mới</button>
  </form>

  <h1>Danh mục</h1>
  <table class="table table-bordered table-hover" id="table-list">
    <thead>
      <tr>
        <th>ID</th>
        <th>Tên Danh mục</th>
        <th>Trạng thái</th>
        <th>Lựa chọn</th>
      </tr>
    </thead>
    <tbody id="list">
      <!-- Dữ liệu load vào đây -->
    </tbody>
  </table>
</div>
```

Bước 2: Mở file **demo.js** thêm đoạn code bắt sự kiện nút bấm thêm mới

```
$(document).ready(function(){
  // Các sự kiện khác để nguyên ở đây
  $('.add-new').click(function(){
    var _name = $('input#name').val(); // lấy dữ liệu trên input có id="name"
    var status = $('input#status').val(); // lấy dữ liệu trên input có id="status"
    var _data = {name: _name, status: status}; // chuẩn bị đối tượng dữ liệu
    $.ajax({
      url: 'https://localhost:44341/api/categories',
      type: 'POST', // Phương thức POST => gọi api thêm danh mục,
      data: JSON.stringify(_data), // truyền data name lên cho server,
      contentType: "application/json; charset=utf-8", // gửi dữ liệu tiến việt có dấu
      success: function(res){
        location.reload();// Load lại trang web
      },
      error: function (req, status, error) {
        console.log(status);
      }
    });
  });
});
```



Bước 3: F5 để refresh lại trình duyệt và test thử kết quả

Thêm mới danh mục

Tên danh mục

Trạng thái

Thêm mới

Danh mục

ID	Tên Danh mục	Trạng thái	Lựa chọn
1	Ti Vi	Hiển thị	<button>Xem</button> <button>Xóa</button>
2	Tủ Lạnh	Hiển thị	<button>Xem</button> <button>Xóa</button>
3	Củ đề	Hiển thị	<button>Xem</button> <button>Xóa</button>
4	Ổn Áp	Hiển thị	<button>Xem</button> <button>Xóa</button>

Bài 7: Sử dụng Ajax để chỉnh sửa thông qua phương thức put

Bước 1: Mở lại file demo.js, tìm đến đoạn code tạo thêm nút bấm chỉnh sửa

```
$.get('https://localhost:44341/api/categories', function(res){
    var html = ''; // biến để chứa cấu trúc tr, td của bảng
    for (let i = 0; i < res.length; i++) {
        var cat = res[i];
        var status = cat.status == 1 ? 'Hiển thị' : 'Đang ẩn';
        html += '<tr>';
        html += '<td>' + cat.id + '</td>';
        html += '<td>' + cat.name + '</td>';
        html += '<td>' + status + '</td>';
        html += '<td>';
        html += '<button type="button" data-id="' + cat.id + '" class="btn btn-xs btn-';
        success '<button type="button" data-id="' + cat.id + '" class="btn btn-xs btn-primary show-';
        html += '<button type="button" data-id="' + cat.id + '" class="btn btn-xs btn-';
        danger delete">Xóa</button>';
        html += '</td>';
        html += '</tr>';
    }
    // gán nội dung của biến html vào thẻ có id="list"
    $('#list').html(html);
});
```

Giao diện có dạng

Thêm mới danh mục

Tên danh mục Trạng thái

Danh mục

ID	Tên Danh mục	Trạng thái	Lựa chọn
1	Ti Vi	Hiện thị	<input type="button" value="Chỉnh sửa"/> <input type="button" value="Xem"/> <input type="button" value="Xóa"/>
2	Tủ Lạnh	Hiện thị	<input type="button" value="Chỉnh sửa"/> <input type="button" value="Xem"/> <input type="button" value="Xóa"/>
3	Củ dề	Hiện thị	<input type="button" value="Chỉnh sửa"/> <input type="button" value="Xem"/> <input type="button" value="Xóa"/>
4	Ổn Áp	Hiện thị	<input type="button" value="Chỉnh sửa"/> <input type="button" value="Xem"/> <input type="button" value="Xóa"/>

Bước 2: Tại file **demo.js** thêm đoạn code bắt sự kiện khi click vào nút edit gọi ajax để lấy về chi tiết 1 bản ghi theo id qua api

<http://localhost:44341/categories/id> (id sẽ được thay thế tùy vào nút click)

Show khi gọi dữ liệu về sẽ đổ lên form và hiển thị modal của bs3

Code có dạng

```
// Sự kiện để gọi api chi tiết 1 bản ghi theo id
$(document).on('click', '.edit', function () {
    var $_this = $(this); // lưu lại phần tử hiện đang click
    var id = $(this).data('id'); // lấy giá trị từ thuộc tính data-id của thẻ button
    // Gọi ajax
    $.ajax({
        url: 'https://localhost:44341/api/categories/' + id,
        type: 'GET', // Phương thức Delete => gọi api xóa danh mục
        success: function (res) {
            $('#input#edit_id').val(res.id); // Gán dữ liệu cho input id="edit_id"
            $('#input#edit_name').val(res.name); // Gán dữ liệu cho input id="edit_name"
            $('#input#edit_status').val(res.status); // Gán dữ liệu cho input id="edit_status"
            $('#modal-edit').modal('show'); // một modal có id="modal-edit"
        }
    });
});
```

Bước 3: Tạo modal có id="modal-edit" và có form giống như form thêm mới

```
<!-- Thêm Modal chỉnh sửa-->
<div class="modal fade" id="modal-edit">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <button type="button" class="close" data-dismiss="modal" aria-hidden="true">&times;</button>
```

```
<h4 class="modal-title">Chỉnh sửa thông tin</h4>
</div>
<div class="modal-body">
  <form action="" method="POST" class="form-inline" id="form-add">
    <input type="hidden" id="edit_id" class="form-control">
    <div class="form-group">
      <label>Tên danh mục</label>
      <input class="form-control" id="edit_name">
    </div>
    <div class="form-group">
      <label>Trạng thái</label>
      <input type="number" class="form-control" id="edit_status">
    </div>
    <button type="button" class="btn btn-primary update">Cập nhật</button>
  </form>
</div>
</div>
</div>
</div>
```

Khi click vào nút Chỉnh sửa modal hiện lên như sau

Chỉnh sửa thông tin ×

Tên danh mục

Trạng thái

Cập nhật

Bước 4: Mở lại file demo.js thêm code bắt sự kiện cho nút cập nhật lấy dữ liệu trên form và gọi api có phương thức PUT

```
$('.update').click(function () {
  var _id = $('#input#edit_id').val(); // lấy dữ liệu trên input có id="edit_id"
  var _name = $('#input#edit_name').val(); // lấy dữ liệu trên input có id="edit_name"
  var status = $('#input#edit_status').val(); // lấy dữ liệu trên input có id="edit_status"
  var _data = {id:_id, name: _name, status: status }; // chuẩn bị đối tượng dữ liệu
  if (_name != '') {
    $.ajax({
      url: 'https://localhost:44341/api/categories/'+_id,
      type: 'PUT', // Phương thức PUT => gọi api thêm danh mục,
      data: JSON.stringify(_data), // truyền data name lên cho server,
      contentType: "application/json; charset=utf-8",
      success: function (res) {
        location.reload();
      },
      error: function (req, status, error) {
        console.log(status);
      }
    });
  } else {
    alert('Vui lòng nhập tên danh mục');
  }
});
```

Bước 5: Click vào nút edit nhập dữ liệu mới và test thử

BÀI 8: TẠO ỨNG DỤNG ANGULAR ĐỂ ÁP DỤNG .NET CORE API

Chú ý: máy của bạn phải cài đặt angular-cli trước khi thực hiện các bước này
Xem list video sau để biết thêm về kiến thức angular

<https://www.youtube.com/watch?v=Qdt0rQTNDfg&list=PLFWDoeAHRLTb2WpindrtFmA6zLn8Afnx3>

Bước 1: Tạo ứng dụng angular

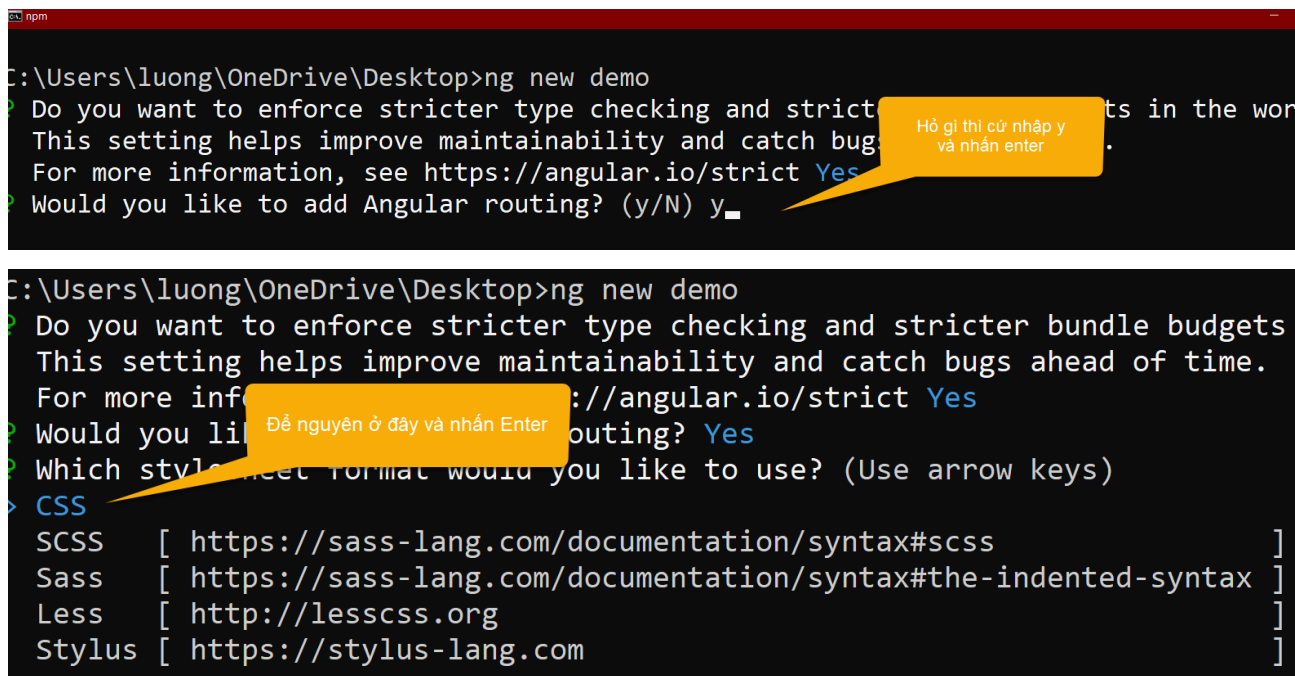
Mở cmd của windows lên và gõ lệnh sau

```
ng new demo
```

Trong đó

ng new là từ khóa bạn cần phải viết đúng

demo là tên thư mục dự án tùy chọn (viết thường, không dấu, không khoảng cách)



```
C:\Users\luong\OneDrive\Desktop>ng new demo
? Do you want to enforce stricter type checking and stricter bundle budgets? This setting helps improve maintainability and catch bugs ahead of time. For more information, see https://angular.io/strict Yes
? Would you like to add Angular routing? (y/N) y
? Which style format would you like to use? (Use arrow keys)
> CSS
SCSS [ https://sass-lang.com/documentation/syntax#scss ]
Sass [ https://sass-lang.com/documentation/syntax#the-indented-syntax ]
Less [ http://lesscss.org ]
Stylus [ https://stylus-lang.com ]
```

Hỗ gì thì cứ nhập y và nhấn enter

Để nguyên ở đây và nhấn Enter

Sau khi cài xong nhớ gõ cd ten_dự_án để vào thư mục dự án

Xem ví dụ ở đây thư mục dự án là demo => cd demo

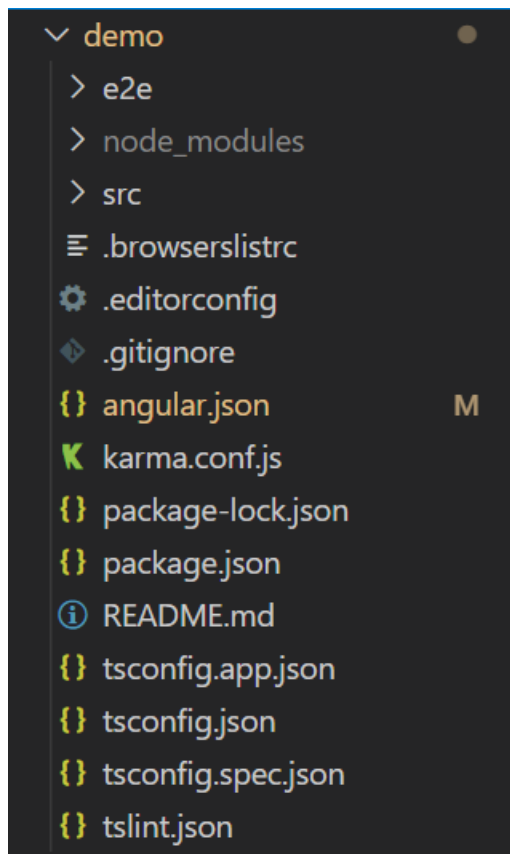
The file will have its original line endings in your working directory.
Successfully initialized git.

```
C:\Users\luong\OneDrive\Desktop>cd demo
```

Sau khi cài xong gõ
cd tên_du_án
nhấn Enter
Để đi vào thư mục dự án

```
C:\Users\luong\OneDrive\Desktop\demo>
```

Thư mục code có dạng như sau



Bước 2: khởi động dự án bằng lệnh `ng serve -o`

```
C:\Users\luong\OneDrive\Desktop\demo>ng serve -o
? Would you like to share anonymous usage data about this project with the Angular Team
Google under Google's Privacy Policy at https://policies.google.com/privacy?details
details and how to change this setting, see http://angular.io/analytics. (y/N)
```

Nhập Y nhấn Enter

```
** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **
```

```
✓ Compiled successfully.
✓ Browser application bundle generation complete.
```

Dự án tự mở theo link

Initial Chunk Files	Names	Size
styles.css, styles.js	styles	346.70 kB

4 unchanged chunks

Build at: 2021-03-05T09:50:36.731Z - Hash: 0...ms

Build thành công

```
✓ Compiled successfully.
```

Chú ý: Trình duyệt sẽ tự mở link dự án angular

<http://localhost:4200/>

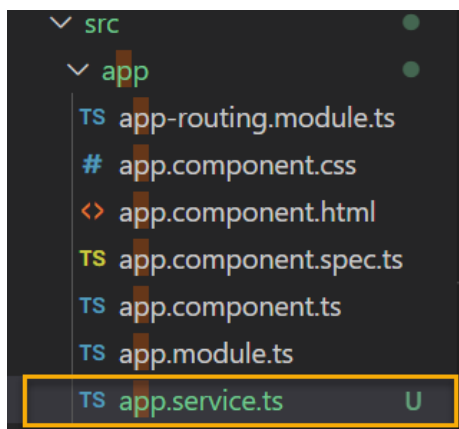
Bước 3: mở file **src/app/app.module.ts** và thêm module HttpClientModule như sau

```
import { HttpClientModule } from '@angular/common/http';
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    HttpClientModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

Bước 4: Tạo service để gọi api

Mở thư mục code của ứng dụng angular của bạn lên

Trong thư mục **src/app** tạo thêm file **app.services.ts** (có đuôi mở rộng là .ts là ngôn ngữ typescript),



Nội dung file **app.service.ts** như sau

```
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
```

```
import { Observable } from 'rxjs';
const _api = "https://localhost:44341/api/"; // khai báo link api cơ bản
@Injectable({
  providedIn: 'root'
})
export class AppService {
  constructor(private http: HttpClient) { }
  // phương thức getList gọi api categories GET trả về danh sách đối tượng categories
  getList(): Observable<any> {
    return this.http.get(_api + 'categories');
  }

  // phương thức addNew gọi categories POST thêm mới 1 ban ghi mới
  addNew(data:any): Observable<any> {
    return this.http.post(_api + 'categories', data);
  }

  // phương thức updateById gọi categories PUT cập nhật 1 ban ghi mới
  updateById(id:any, data:any): Observable<any> {
    return this.http.put(_api + 'categories/'+id, data);
  }

  // phương thức getById gọi api categories/:id GET trả về 1 đối tượng category
  getById(id:any): Observable<any> {
    return this.http.get(_api + 'categories/'+id);
  }

  // phương thức deleteById gọi api categories/:id DELETE xóa 1 đối tượng category
  deleteById(id:any): Observable<any> {
    return this.http.delete(_api + 'categories/'+id);
  }
}
```

Bước 5: Mở file src/app/app.component.ts và cập nhật code như sau

```
import { Component, OnInit } from '@angular/core';
import { AppService } from '../app.service';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent implements OnInit {
  title = 'demo';
  categoris: any; // khai báo biến chứa danh sách categoies
  constructor(private app: AppService) {}
  /*
   * khai báo hàm getList lấy về danh sách
   * hàm này có thể sử dụng ở nhiều trường hợp khác nhau
   */
  getList(){
    // sử dụng app.services đã tạo ở trên
    this.app.getList().subscribe(res => { //phương thức lấy về danh sách
      this.categoris = res;
    })
  }

  ngOnInit() :void {
    this.getList(); // gọi lại hàm getList để lấy về danh sách categoris
  }
}
```



Bước 6: Mở file `src/app/app.component.html` và thay thế toàn bộ code như sau

```
<div class="container">
<h2>Danh sách danh mục</h2>
<table class="table table-bordered table-hover">
  <thead>
    <tr>
      <th>STT</th>
      <th>Name</th>
      <th>Status</th>
      <th>Action</th>
    </tr>
  </thead>
  <tbody>
    <!-- Duyệt dữ liệu ngay trên cấu trúc html sử dụng *ngFor => vòng lặp for -->
    <tr *ngFor="let cat of categoris; let key = index">
      <td>{{key+1}}</td>
      <td>{{cat.name}}</td>
      <td>{{cat.status == 1 ? 'Hiển thị' : 'Tạm ẩn'}}</td>
      <td>
        <button class="btn btn-xs btn-success">Chỉnh sửa</button>
        <button class="btn btn-xs btn-primary">Xem</button>
        <button class="btn btn-xs btn-danger">Xóa</button>
      </td>
    </tr>
  </tbody>
</table>
</div>
```

Bước 7: Nhúng thêm css của bootstrap để áp dụng giao diện cơ thể nhìn

Mở file `src/index.html` nhúng bootstrap css online

```
<link rel="stylesheet" href="https://netdna.bootstrapcdn.com/bootstrap/3.3.4/css/bootstrap.min.css">
```

Nhúng thêm jquery của bootstrap online

```
<script src="https://code.jquery.com/jquery.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
```

Mở lên trình duyệt của ứng dụng angular và xem kết quả



Danh sách danh mục

STT	Name	Status	Action
1	Ti Vi 12	Hiện thị	<button>Chỉnh sửa</button> <button>Xem</button> <button>Xóa</button>
2	Tủ Lạnh	Hiện thị	<button>Chỉnh sửa</button> <button>Xem</button> <button>Xóa</button>
3	Củ đề	Hiện thị	<button>Chỉnh sửa</button> <button>Xem</button> <button>Xóa</button>
4	Ồn Áp	Hiện thị	<button>Chỉnh sửa</button> <button>Xem</button> <button>Xóa</button>

Bước 8: Sự kiện khi click vào nút xóa

Mở file **src/app/app.component.html** tìm đến nút xóa và thêm sự kiện click với code như sau

