MSSV: 2312647

Họ tên: Lê Anh Khoa

Lớp: CTK47A

**LAB 1**

**• Tìm hiểu thư viện System.Net**

Trong .NET Framework, thư viện System.Net là một phần quan trọng hỗ trợ các lập trình viên xây dựng các ứng dụng mạng. Thư viện này cung cấp các lớp và phương thức cần thiết để làm việc với:

* Giao thức TCP/IP,
* Kết nối và truyền tải dữ liệu qua Internet,
* Làm việc với địa chỉ IP,
* Giao tiếp thông qua HTTP/HTTPS,
* Truy vấn DNS (Domain Name System),
* Xây dựng các client-server đơn giản.

Thông qua System.Net, chúng ta có thể thực hiện các thao tác như: xác định địa chỉ IP của một tên miền, lấy tên máy tính từ địa chỉ IP, tạo kết nối mạng, gửi và nhận dữ liệu,...

**• Tìm hiểu các lớp: IPAddress, IPHostEntry, DNS**

**1. Lớp IPAddress**

Lớp IPAddress là lớp đại diện cho một địa chỉ IP trong .NET. Địa chỉ IP có thể thuộc loại IPv4 (ví dụ: 192.168.1.1) hoặc IPv6 (ví dụ: fe80::1).

Lớp này cung cấp các phương thức và thuộc tính giúp chúng ta:

* Phân tích (parse) địa chỉ IP từ chuỗi,
* Chuyển đổi địa chỉ IP thành dạng chuỗi,
* Truy cập các địa chỉ IP đặc biệt như địa chỉ localhost (127.0.0.1), địa chỉ broadcast,...

Ví dụ, để tạo một địa chỉ IP từ chuỗi, ta có thể sử dụng phương thức Parse:

IPAddress ip = IPAddress.Parse("192.168.1.100");

Console.WriteLine("Địa chỉ IP: " + ip.ToString());

Ngoài ra, IPAddress cũng cung cấp sẵn các địa chỉ IP đặc biệt thông qua thuộc tính tĩnh:

* IPAddress.Loopback: đại diện cho địa chỉ localhost (127.0.0.1),
* IPAddress.Any: đại diện cho bất kỳ địa chỉ IP nào (0.0.0.0), thường dùng trong server khi lắng nghe tất cả các card mạng,
* IPAddress.Broadcast: đại diện cho địa chỉ broadcast (255.255.255.255),
* IPAddress.None: giá trị không hợp lệ của địa chỉ IP.

Thông qua lớp IPAddress, việc xử lý và thao tác với địa chỉ IP trong C# trở nên đơn giản và rõ ràng hơn.

**2. Lớp IPHostEntry**

Khi thực hiện truy vấn DNS để tìm kiếm địa chỉ IP từ một tên miền, kết quả trả về không phải là một địa chỉ IP đơn lẻ mà thường là một tập hợp các địa chỉ IP cùng với các thông tin liên quan. Trong .NET, kết quả của một truy vấn DNS như vậy được đóng gói trong một đối tượng IPHostEntry.

Lớp IPHostEntry đại diện cho một mục nhập DNS, bao gồm các thông tin như:

* **HostName**: Tên máy chủ (tên miền) được truy vấn,
* **AddressList**: Một mảng chứa các địa chỉ IP liên quan đến tên miền đó,
* **Aliases**: Danh sách các tên thay thế (alias) cho tên miền, nếu có.

Ví dụ, nếu ta thực hiện truy vấn tên miền google.com, đối tượng IPHostEntry sẽ chứa danh sách các IP mà máy chủ của Google sử dụng để xử lý các yêu cầu:

IPHostEntry hostEntry = Dns.GetHostEntry("google.com");

Console.WriteLine("Tên máy chủ: " + hostEntry.HostName);

foreach (IPAddress ip in hostEntry.AddressList) {

Console.WriteLine("Địa chỉ IP: " + ip.ToString());

}

Đoạn mã trên sẽ in ra tên máy chủ chính thức mà DNS trả về, cùng với danh sách các địa chỉ IP được liên kết với tên miền đó.

**3. Lớp Dns**

Lớp Dns là lớp trung tâm trong System.Net dùng để thực hiện các thao tác liên quan đến **Domain Name System (DNS)**. Lớp này cung cấp nhiều phương thức tĩnh để tra cứu địa chỉ IP từ tên miền và ngược lại.

**Các phương thức phổ biến:**

* Dns.GetHostEntry(string hostname): Trả về đối tượng IPHostEntry chứa các địa chỉ IP tương ứng với tên miền đã chỉ định.
* Dns.GetHostEntry(IPAddress address): Truy vấn DNS ngược để lấy tên miền từ địa chỉ IP.
* Dns.GetHostName(): Trả về tên máy tính cục bộ đang chạy chương trình.

**Ví dụ 1: Lấy địa chỉ IP từ tên miền**

IPHostEntry entry = Dns.GetHostEntry("microsoft.com");

foreach (IPAddress ip in entry.AddressList) {

Console.WriteLine("IP của Microsoft: " + ip.ToString());

}

**Ví dụ 2: Lấy tên máy tính cục bộ và các địa chỉ IP của nó**

string localHost = Dns.GetHostName();

Console.WriteLine("Tên máy tính cục bộ: " + localHost);

IPHostEntry localEntry = Dns.GetHostEntry(localHost);

foreach (IPAddress ip in localEntry.AddressList) {

Console.WriteLine("Địa chỉ IP nội bộ: " + ip.ToString());

}

Như vậy, với lớp Dns, lập trình viên có thể dễ dàng tương tác với DNS để lấy thông tin liên quan đến mạng, phục vụ cho việc xây dựng các ứng dụng client-server, truy cập website, xử lý dữ liệu mạng,...