

## 1. URI

URI, viết tắt của Uniform Resource Identifiers, tạm dịch là *nh danh tài nguyên*. *Nh danh tài nguyên* trong mạng Internet là một trong rất nhiều vấn đề mà một hệ thống phân tán phải giải quyết. Tất nhiên URI không chỉ trong việc truy cập tới các trang web, mà còn dùng trong việc truy cập tới các server FTP, trong các dịch vụ web và cả trong việc truy cập các tệp tin cục bộ.

URI được định nghĩa trong RFC 2396. Cùng với URI có hai thuật ngữ khác gồm URL và URN. URL (Uniform Resource Locator) trong các tài liệu chuẩn hiện nay không còn được dùng nữa. Người ta dùng URI thay cho URL với ý nghĩa rộng hơn. URN (Uniform Resource Name) là một URI đã được chuẩn hóa, nó được dùng khi nói tới một tài nguyên độc lập trong mạng chứa nó.

Trên thực tế, ta phân tích các thành phần của một URI. Cấu trúc hay gọi tắt của một URI là:

`http://www.wrox.com` hay `mailto:christian@nagel.net`

Cấu trúc cơ bản của URI có dạng:

`http://www.globalknowledge.net:80/training/generic.asp?pageid=1078&country=DACH`

- Phần đầu của URI là định danh, tức scheme. Nó xác định không gian làm việc của URI bằng những ký hiệu cú pháp phải tuân thủ. Dấu `://` phân cách định danh với phần còn lại của URI.
- Sau dấu `://` là địa chỉ IP hay tên của server. Đây là `www.globalknowledge.net`.
- Sau tên hay địa chỉ server là số hiệu cổng kết nối tới server. Nếu ta không chỉ rõ số hiệu cổng định danh thì giao thức định danh sẽ gán cổng mặc định, chẳng hạn giao thức HTTP mặc định dùng cổng 80.
- Sau địa chỉ cổng là đường dẫn tới tài nguyên mong muốn. Ví dụ `/training/generic.asp`.
- Dấu `?` phân cách phần đầu URI với phần truy vấn (query). Trong ví dụ, truy vấn xác định là `pageid=1078&country=DACH`.
- Một tài nguyên mà truy vấn muốn truy cập tới có thể được phân chia thành các mảnh với các tên gọi riêng, hay bookmark trong các trang HTML. Ký tự `#` dùng để phân mảnh.

Theo đó, nếu một URL có dạng `http://www.microsoft.com/net/basics/glossary.asp#.NETFramework`, thì phần mà truy vấn muốn truy cập là `#.NETFramework`.

Lưu ý là, nếu `#` có mặt trong phần query thì nó không phải là phần mã tài nguyên, mà nó là một bộ phận của chuỗi truy vấn. Trong một URL, ta có thể có query, hay một mã, nhưng không phải cả hai. Một số dấu có thể sử dụng trong URI, mà ký tự có chức năng riêng. Chúng bao gồm: `;/?:@&=+$` và ,

## Uri

Uri thuộc lớp `System`. Nó có các thuộc tính và phương thức truy vấn tham số, so sánh và thao tác các URI.

Ta có thể tạo một Uri bằng cách truy vấn chuỗi URI cho hàm tạo:

```
Uri uri = new Uri("http://msdn.microsoft.com/code/default.asp");
```

Nếu ta đã có một Uri cũ thì có thể tạo mới URI bằng cách thao tác URI cũ và vị trí URI khác:

```
Uri baseUri = new Uri("http://msdn.microsoft.com");  
Uri newUri = new Uri(baseUri, "code/default.asp");
```



Nếu URI cơ bản đã có chứa sẵn đường dẫn thì nó có thể bỏ qua khi tạo mới đối tượng URI liên quan. Khi tạo URI liên quan, chỉ cần hình thức, số hiệu cổng và tên server tham gia dưới dạng các tham số tạo mới.

## Kiểm tra tên host và hình thức dịch vụ

Lớp Uri có các phương thức tĩnh kiểm tra tên của một host và dịch vụ cung cấp. Phương thức Uri.CheckSchemeName() trả về giá trị true nếu có tồn tại dịch vụ, và Uri.CheckHostName() và Uri.IsHostValid() kiểm tra tên host có hợp lệ theo kiểu UriHostNameType. Chúng có thể là một trong các giá trị sau:

Giá trị UriHostNameType	Mô tả
Basic	Tên tồn tại tên host, nhưng không xác định kiểu.
Dns	Kiểu này thường được sử dụng để kiểm tra tên host.
IPv4	Nếu chuỗi trả về đúng định dạng địa chỉ IP, phân cách bởi các dấu chấm, đó là kiểu IPv4.
IPv6	Nếu chuỗi trả về đúng định dạng địa chỉ IP, phân cách bởi các dấu chấm, đó là kiểu IPv6.
Unknown	Giá trị này được trả về nếu địa chỉ host có chứa các ký tự không hợp lệ.

Có thể dùng Uri.CheckHostName() để kiểm tra tính hợp lệ của chuỗi tên do người dùng nhập vào, nhưng nó không cho biết tên host có tồn tại trên thực tế hay không và có thể nhanh chóng thay đổi. Sử dụng Uri.IsHostValid() để kiểm tra tên host có hợp lệ hay không. Sử dụng Uri.IsUri() để kiểm tra chuỗi địa chỉ IP rõ ràng hơn trong trường hợp này khi ta chắc chắn chuyển đổi từ tên miền host sang địa chỉ IP.

## Các tính chất của Uri

Lớp Uri có nhiều thuộc tính để truy cập các thành phần của URI. Ta sẽ dùng các thuộc tính này trong bài tiếp theo.

<http://www.globalknowledge.net:80/training/generic.asp?pageid=1078&country=DA>  
CH

Thuộc tính	Mô tả
AbsoluteUri	Địa chỉ URI đầy đủ mô tả một URL. Nếu có chứa rõ sơ hiệucôngtrùng vớicôngchunquy nh trong giao thệcthì hàm trả về giá trị true. Khi đó, thuộc tính Uri.IsUri() trả về giá trị true. Ví dụ: http://www.globalknowledge.net/training/generic.asp?pageid=1078country=DACH

	Nếu tên tập tin được truy cập cho hàm tạo của lớp Uri, thì AbsoluteUri sẽ thêm tên tập tin mới từ file:///.
Scheme	Là hình thức dịch vụ mà ta thấy nó ngay đầu dòng. Đây là http.
Host	Thuộc tính này thể hiện tên host của URI. www.globalknowledge.net
Authority	Còn gì nữa không? Thuộc tính Host nêu sự khác biệt giữa các quy định giao thức. Nếu dùng một sự khác biệt thì thuộc tính này chỉ ra tên host và sự khác biệt.
HostNameType	Kiểu tên host. Kiểm tra giá trị của s để biết kiểu UriHostNameType. Tr�ng hợp này có kiểm tra là UriHostNameType.Dns
Port	Số của thuộc tính này để chỉ ra cổng. Đây là 80.
AbsolutePath	Đường dẫn tài nguyên, tính từ sau sự khác biệt cho tới chuỗi truy cập. Ví dụ này là /training/generic.asp
LocalPath	Đường dẫn cục bộ. Chính là /training/generic.asp. Như ta thấy trong mô tả yêu cầu HTTP, không có sự khác biệt giữa AbsolutePath và LocalPath. Nhưng sự khác biệt sẽ thể hiện nếu tài nguyên được tham chiếu qua mô tả mô tả. Vì mô tả tài nguyên chia sẻ qua mạng, thuộc tính LocalPath trở về có đường file:\\server\share\directory\file.txt, thì AbsolutePath lại trở về tên server và các tên GUID để định danh, chia sẻ.
Query	Chuỗi truy cập theo sau đường dẫn. Ví dụ này là ?pageid=1078&country=DACH
PathAndQuery	Thuộc tính này trở về tập hợp đường dẫn và chuỗi truy cập.