URI、URL

URI Represents a Uniform Resource Identifier (URI) reference.

URL Class URL represents a Uniform Resource Locator, a pointer to a "resource" on the World Wide Web.

# URI

## 继承关系

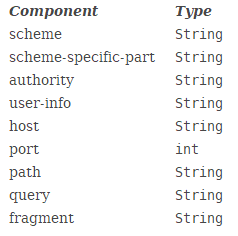
public final class URI extends Object implements Comparable<URI>, Serializable



All Implemented Interfaces:**Serializable, Comparable<URI>**

Represents a **Uniform Resource Identifier (URI)** reference.

## 组成要素



## 构造方法

URI(String str)

Constructs a URI by parsing the given string.

URI(String scheme, String ssp, String fragment)

Constructs a URI from the given components.

URI(String scheme, String userInfo, String host, int port, String path, String query, String fragment)

Constructs a hierarchical URI from the given components.

URI(String scheme, String host, String path, String fragment)

Constructs a hierarchical URI from the given components.

URI(String scheme, String authority, String path, String query, String fragment)

Constructs a hierarchical URI from the given components.

## 一般方法

### 静态方法create

static URI create(String str)

Creates a URI by parsing the given string.

### 实现了**Comparable**接口：compareTo

int compareTo(URI that)

Compares this URI to another object, which must be a URI.

### equals

boolean equals(Object ob)

Tests this URI for equality with another object.

### resolve方法

URI resolve(String str)

Constructs a new URI by parsing the given string and then resolving it against this URI.

URI resolve(URI uri)

Resolves the given URI against this URI.

### toURL

URL toURL() Constructs a URL from this URI.

### getXxx方法

# URL

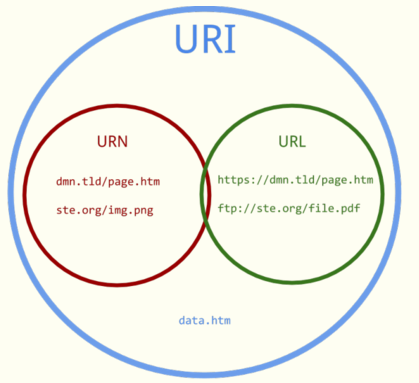
# URIs, URLs, and URNs

A URI is a **uniform resource identifier** while a URL is **a uniform resource locator**. **Hence every URL is a URI, abstractly speaking, but not every URI is a URL.** This is because there is another subcategory of URIs, uniform resource names (URNs), which name resources but do not specify how to locate them. The mailto, news, and isbn URIs shown above are examples of URNs.

The conceptual distinction between URIs and URLs is reflected in the differences between this class and the URL class.

An instance of this class represents a URI reference in the syntactic sense defined by RFC 2396. **A URI may be either absolute or relative**. A URI string is parsed according to the generic syntax without regard to the scheme, if any, that it specifies. No lookup of the host, if any, is performed, and no scheme-dependent stream handler is constructed. Equality, hashing, and comparison are defined strictly in terms of the character content of the instance. In other words, a URI instance is little more than a structured string that supports the syntactic, scheme-independent operations of comparison, normalization, resolution, and relativization.

An instance of **the URL class**, by contrast, represents the syntactic components of a URL together with some of the information required to access the resource that it describes. A URL must be absolute, that is, it must always specify a scheme. A URL string is parsed according to its scheme. A stream handler is always established for a URL, and in fact it is impossible to create a URL instance for a scheme for which no handler is available. Equality and hashing depend upon both the scheme and the Internet address of the host, if any; comparison is not defined. In other words, a URL is a structured string that supports the syntactic operation of resolution as well as the network I/O operations of looking up the host and opening a connection to the specified resource.



URI，URL，URN

从上面的那幅图可以看出来，**一共有三个不同的概念URI,URL,URN。**这讨论这样的问题时，最好的方法就是回到原点啊，这里我们在RFC 3986: Uniform Resource Identifier (URI): Generic Syntax里面收集了点资料：

“A Uniform Resource Identifier (URI) 是一个紧凑的字符串用来标示抽象或物理资源。”

“*A URI 可以进一步被分为定位符、名字或两者都是.* 术语“Uniform Resource Locator” (URL) 是URI的子集, 除了确定一个资源,还提供一种定位该资源的主要访问机制(如其网络“位置”)。“

那我们无所不知的维基百科把这段消化的很好，并描述的更加形象了：

“**URI可以分为URL,URN或同时具备locators 和names特性的一个东西**。URN作用就好像一个人的名字，URL就像一个人的地址。换句话说：URN确定了东西的身份，URL提供了找到它的方式。”

通过这些描述我们可以得到一些结论：

首先，URL是URI的一种（通过那个图就看的出来吧）。所以有人跟你说URL不是URI，他就错了呗。但也不是所有的URI都是URL哦，就好像蝴蝶都会飞，但会飞的可不都是蝴蝶啊，你让苍蝇怎么想！

让URI能成为URL的当然就是那个“访问机制”，“网络位置”。e.g. http:// or ftp://.。

URN是唯一标识的一部分，就是一个特殊的名字。

　　下面就来看看例子吧，当来也是来自权威的RFC：

ftp://ftp.is.co.za/rfc/rfc1808.txt (also a URL because of the protocol)

http://www.ietf.org/rfc/rfc2396.txt (also a URL because of the protocol)

ldap://[2001:db8::7]/c=GB?objectClass?one (also a URL because of the protocol)

mailto:John.Doe@example.com (also a URL because of the protocol)

news:comp.infosystems.www.servers.unix (also a URL because of the protocol)

tel:+1-816-555-1212

telnet://192.0.2.16:80/ (also a URL because of the protocol)

urn:oasis:names:specification:docbook:dtd:xml:4.1.2

　　这些全都是URI, 其中有些是URL. 哪些? 就是那些提供了**访问机制**的.

总结

下面到了回答问题的时候了：

当我们替代web地址的时候，URI和URL那个更准确？

基于我读的很多的文章，包括RFC，我想说**URI更准确**。

别急，我有我的理由：

我们经常使用的URI不是严格技术意义上的URL。例如：你需要的文件在files.hp.com. 这是URI，但不是URL--系统可能会对很多协议和端口都做出正确的反应。

你去*http://files.hp.com 和ftp://files.hp.com.可能得到完全不同的内容。*这种情况可能更加普遍，想想不同谷歌域名上的不同服务啊。

所以，用URI吧，这样你通常技术上是正确的，URL可不一定。最后“URL”这个术语正在被弃用。所以明智吧少年！

结语

If you don’t mind being “that guy”, URI is probably the more accurate term to use. But if you are in the linguist / “use what’s understood” camp, feel free to go with URL.