Coding

博客园 首页 新随笔 管理

URL的结构介绍

URL的结构，关于这方面的blog也能一搜一大把。。。最近也在看方面的基础，也记录下。。。

参考：Web之困

1. 什么是URL

URL(Uniform Resource Locator)统一资源定位符，就是由一串简单的文本字符组成。**一条符合规范的URL对应的是服务器的一个特定的资源(如HTML页面，一张图片等)。**

Scheme:	//	Login:password@	Address	:port	/path/to/resource	?query_string	#fragment

按照完整的URL的结构定义，一共分成上述8个部分，其中各部分有着一定的特征，并且有些部分是必须的，有些是可选的。而标准的定义与浏览器对这些标准的实现又有着很大的差异。

Scheme部分：协议部分。

协议名称是由一串**不区分大小写的字母**组成，以**:**作为结束符。协议所表示的是获取该资源需要使用的协议。如HTTP、HTTPS等。而浏览器将支持一些额外的协议，如data:和javascript:等。

//部分：层级URL标识符号

基本上每个URL中都会包含这个符号，是固定的；可以理解为把协议与后面的信息进行分隔开的一个符号。按照书上的说明，一个好处是Web应用无需关注某个协议的具体实现，而只需要关注于’//’符号后面的指向地址即可。

但是也存在这非层级结构的URL：例如，mailto:协议。当使用

mailto:user@example.com?....的时候，该URL将能够传递到默认的邮件客户端程序而无需其他的解析。

Login:password@部分：身份验证

其实这一块信息我们看到的比较少，这是一个可选部分，一般的协议(http\https之类)都会使用默认的匿名形式进行数据获取，该部分使用的是**@**作为该部分的结束符号。

Address部分：服务器地址

这是一个很关键的部分，这关系到你需要从哪个服务器上去获取资源。而我们看到的比较多的是这部分以域名(htc.org)的形式呈现，还有以Ipv4(220.181.111.188)的地址呈现。当然也能够以Ipv6的形式呈现。

按照标准的描述是这部分只能用：数字、“.”、“-”组成。但浏览器对这支持的字符会比较多。

Port部分： 服务器端口

这里是属于网络端口，16位， 因此可选为[0~2^16)，这里的端口并不是物理端口，而是逻辑端口；只要是为了处理多进程时数据进行传输的时候，保证各进程中数据不会发生紊乱，能够传送到相应的进程中所设定的（参考：http://baike.baidu.com/link?url=MDeMzLjNepWAvUUhGaHPFZMnUk8z3oFGIVz_qqPkOr_HEFxGDf6Gf5pdJd7lVc_XOAfgta2D0augSiFRadL0Kq）

不同网络协议都有自己特定的端口号：如http 80

随笔分类

Algorithm(2)
ISSUES(3)
POJ(2)
REMAININGS(2)
Timus(3)
WEB(2)

/path/to/resource：文件路径

前面提到的URL指向的是一个唯一确定的资源，而这里指向的是资源的完整路径（即存储的位置），一般都是用 / 进行分层描述。

?query_string：查询字符串

这里的查询字符串是用于参数传递给服务器端。但标准没有对这一部分有着特别严格的规定。这一部分是以？开始作为标识，而现在一般的用法都是类似于以下的形式，?name=hello&id=5&… ,并且这种用法也被服务器端语言（如PHP等）所支持，如PHP获取该查询值的方法是：

```
$_GET['id']
$_POST['id']
```

#fragment: 片段ID

该部分与上面的?后面的表单信息本质的区别就是这部分内容不会被传递到服务器端。一般用于页面的锚。就是我们常见的网站右下脚一般有一个回到顶部的按钮，一般就是使用其实现的。

例如：

```
<!DOCTYPE HTML>
<html>
    <head>
        <title>    return </title>
        <script>
            function file(){
                var xx = "hello<br/>world<br/>" ;
                for( var i = 0; i < 100; i++ ){
                    xx += "<br/>" ;
                }
                xx += "ni<br/>hao<br/>" ;
                document.getElementById('aa').innerHTML = xx ;
            }
        </script>
    </head>
    <body onload = "file();">
        <p> nihao </p>
        <br>
        <br>
        <a name='hello'> hello hello hello </a> <br/>
        <p id = 'aa'>
        </p>
        <a href="#hello"> 返回HELLO </a>
        <a href="#" target="_self">返回顶部</a>
    </body>
</html>
```

Python获取URL并处理

参考：

http://www.cnblogs.com/qq78292959/archive/2013/04/07/3005763.html

http://my.oschina.net/guol/blog/95699

python里面获取这一部分信息使用的是urlparse模块。

解析成为6部分，返回元组(scheme, netloc, path, parameters, query, fragment)

Scheme	//	Login:passw	Address	:port	/path/to/reso	?query_stri	#fragme

:		ord@			urce	ng	nt

```
from urlparse import urlparse


url = "https://www.zhangsanlisi.com/questions/1000;hello_world?id=10&name=zhangsan#hello"

end_url = urlparse(url)

print end_url
```

```
>>>
ParseResult(scheme='https', netloc='www.zhangsanlisi.com', path='/questions/1000
', params='hello_world', query='id=10&name=zhangsan', fragment='hello')
>>>
```

可以与上面对应，其中netloc描述的是包括验证信息+服务器地址+端口号；而params用的比较少，基本上比较难看到，用于指定特定的参数，参考（http://blog.csdn.net/yueguanghaidao/article/details/16368399）

用；（分号）作为开始标识。

分类: WEB

标签: URL, Web

« 上一篇: POJ2449 Remmarguts' Date
» 下一篇: Python程序退出方式(sys.exit() os._exit() os.kill() os.popen(...))

posted @ 2015-08-16 20:46 ct_usl 阅读(4263) 评论(0) 编辑 收藏

· 程序员的工作、学习与绩效
· 软件开发为什么很难
· 唱吧DevOps的落地，微服务CI/CD的范本技术解读
» 更多知识库文章...