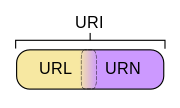
Atitit uri url格式规范与解析器 .URIparser

理解URI和URL的区别，我们引入URN这个概念。

URI = Universal Resource Identifier 统一资源**标志符**  
URL = Universal Resource Locator 统一资源**定位符**  
URN = Universal Resource Name 统一资源**名称**

三者的关系如下图：  
&amp;amp;amp;lt;img src="https://pic2.zhimg.com/f66f9f573436858aeeb2ac3da732f5a9\_b.png" data-rawwidth="180" data-rawheight="108" class="content\_image" width="180"&amp;amp;amp;gt;也就是说，URI分为三种，URL or URN or （URL and URI）也就是说，URI分为三种，URL or URN or （URL and URI）  
URL代表资源的路径地址，而URI代表资源的名称。  
通过URL找到资源是对网络位置进行标识，如：

* **http://example.org/absolute/URI/with/absolute/path/to/resource.txt**
* **ftp://example.org/resource.txt**

通过URI找到资源是通过对名称进行标识，这个名称在某命名空间中，并不代表网络地址，如：

* **urn:issn:1535-3613**

 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, 其中有些事URL. 哪些? 就是那些提供了访问机制的.

URI（IMS用户的身份标识）\_百度百科.mhtml

一.URI简介

概念：统一资源标识符（Uniform Resource Identifier）

组成部分：

1.访问资源的命名机制（scheme）

2.存放资源的主机名（authority）

3.资源自身的名称，由路径表示（path）

格式：scheme：// authority//path，其中authority中又包括了host和port两部分。

content://com.example.project:200/folder/subfolder/etc

\---------/ \-------------------/ \----------------------------/

Scheme host port path

\-------------------/

Authority

1

2

3

4

5

用处：uri主要用来表示一个资源。这个资源有很多种类，包括图片，视频，文件等。

针对资源的种类，uri用以下几种scheme标识：

1.Content：主要操作的是ContentProvider，所以它代表的是数据库中的某个资源

2.http：一个网站资源

3.file：本地机器上的某个资源

4.git：git仓库中某个资源

5.ftp：服务器上的某个资源

6.ed2k：（电驴协议）

---------------------

作者：谁的影子

来源：CSDN

二.Uri源码

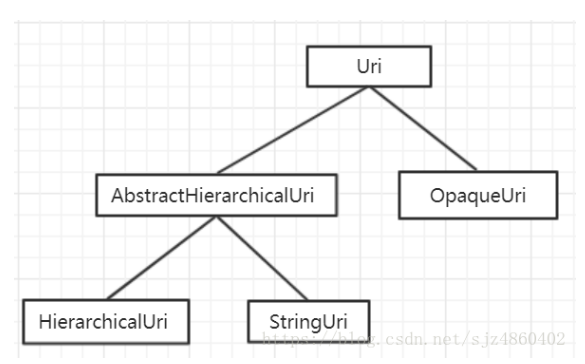
位置：frameworks\base\core\java\android\net\Uri.java

android中对Uri分为两大类：Hierarchical和opaque

Hierarchical：以”//”分级的Uri，比如：http://google.com，content://com.example.project:200/

opaque：没有使用”//”分级的Uri，比如：mailto:nobody@google.com

根据这个分类设定，Uri这个抽象类的实现类和继承关系如下：



以上是Uri.java中比较常用的几个函数。至于其子类OpaqueUri，HierarchicalUri ，StringUri，实现上并不是太难，不做展开。这里需要特殊说明的是Uri的内部类Builder，这个Builder是一个辅助类，用于Uri对象的构建，并发不安全。

还有个比较重要的概念，绝对的透明的URI和相对的URI都是分层的(hierarchical)   
去看看android的URI api吧，所有is判断都会有。   
有了上述概念，看看android中URI的具体类有：OpaqueUri，HierarchicalUri和StringUri。怎么样，能看懂了吧。   
我们继续，如果一个URI是分层的，那么这个URI的schemeSpecificPart（一般简称ssp）是如下结构：   
  
****[//authority][path][?query]**** ([...]表示可选)   
  
而对于那些基于服务器的URI来说，authority结构为：   
  
****[userinfo@]host[:port]****

面的方法吧：   
getScheme() getSchemeSpecificPart() getAuthority() getUserInfo() getHost() getPort() getPath() getQuery() getFragment() (其实是有顺序的，你能看出来吗)   
好了，所有URI的概念就介绍完了，但对于URI类，另一个作用就是处理绝对URI和相对URI。   
如果存在如下的绝对URI：   
http://docs.mycompany.com/api/java/net/ServerSocket.html   
和一个如下相对URI：   
../../java/net/Socket.html#Socket()   
那么可以将它们合并成一个绝对URI：   
http://docs.mycompany.com/api/java/net/Socket.html#Socket()   
这个过程被称为相对URL的*转换*(resolving)。   
与此相反的过程称之为*相对化*（relativization）

/bookmarksHtmlEverythingIndexPrj/src/com/attilax/net/URIparser.java

默认的uri解析器存在bug，，如果密码里面存在金号则解析出错。。

"http://root:p 1@101.132.148.11:22";

所以需要自己写了。

**void** parse(**boolean** rsa) **throws** URISyntaxException {

方法后面添加

String[] aStrings=**this**.input.split("@");

**int** protoIndex=**this**.input.indexOf("//");

**if**(protoIndex<1)**throw** **new** RuntimeException("no protocal::");

userInfo= aStrings[0].split("//")[1];

**int** syegeoStart=input.indexOf("//");

**int** hostEnd=input.indexOf(":", syegeoStart+1);

String hostportString=**this**.input.split("@")[1];

host=hostportString.split(":")[0];

port=Integer.*parseInt*(hostportString.split(":")[1]);

//input.substring(syegeoStart+2,hostEnd);

System.***out***.println("");

}

(9+条消息)Uri与UriMatcher - sjz4860402的专栏 - CSDN博客.mhtml