# Toolkit5.5 Url规则

Toolkit 5.5的Url规则和Toolkit5的.c扩展名的规则类似。

一个典型的Toolkit5.5的Url如下：

<http://localhost:5000/c/xml/insert/UserManager/LogOn>

第一部分localhost:5000是部署的服务器和端口，跟具体的服务器有关。

第二部分c，是HttpHandler。

第三部分xml，是ModuleCreator。

第四部分insert，是页面类型。

第五部分是余下的全部，即UserManager/LogOn，是Source。

归纳一下，就是

http://localhost:5000/{HttpHandler}/{ModuleCreator}/{页面类型}/{Source}

## HttpHandler

Toolkit 5.5定义了IHttpHandler来处理和解析Url。c是ToolkitHttpHandler类的插件注册名。每个类可以自行定义对url的解析和处理规则。我们这里说的仅仅只是c的规则。如果有人开发了自己的IHttpHandler插件，需要有相应的文档来说明其对url的解析规则。

## ModuleCreator

Toolkit5就引入了IModuleCreator来处理Source。ModuleCreator是代码插件，不同的ModuleCreator对Source的处理方式各有不同。这里xml是XmlModuleCreator类的插件注册名，它意味着Source是Xml/Module目录下定义的子路径和文件名（去掉.xml扩展名）。此外，系统还提供了plugin（Source为IModule的注册名），source（或者true，true为了兼容以前而提供。Source为ISource的注册名）。此外，可以根据需要，自行开发相应的IModuleCreator进行注册。如excel（将List数据导出到excel中）。

另外，在这个定义中，如果界面是重定向页面，需要在一开始加上~，如/~xml/。这样，系统会读取Redirector对页面进行跳转。大多数情况下，由于都是调用PageMaker输出内容，所以xml前什么都不要加。

## 页面类型

Toolkit5定义了页面类型的规则。它的结构定义如下：

public interface IPageStyle

{

PageStyle Style { get; }

string Operation { get; }

}

基本的页面类型有List，Insert，Update，Detail，Delete分别代表列表，新增，修改，详情，删除页面。如果无法用上述几个页面都不能表述出页面的类型。那么就需要使用Custom类型的页面，Custom类型的页面除了标注Style为Custom外，还需要标注Operation（内容随意），说明是怎么样的操作。不然如果出现两个Custom页面将无法区分。对于Custom页面类型的表述，系统的规定是C+Operation。如Operation为Tree，那么页面类型就是CTree，如果Operation为空，那么页面类型就是C。

## Source

Source的内容由ModuleCreator来决定。通常，如果采用xml这种ModuleCreator，那么Source就是对应的子路径+文件名。比如示例中，Source就是在Xml/Module/UserManager目录下的LogOn.xml文件。需要注意的是，Toolkit5.5支持Linux版本，那么路径名，文件名必须和实际的一致。如果大小写有任何不一致，都可能出错。

如果ModuleCreator是plugin或者source，那么Source就是插件的注册名了。

比如TracePage的Url就是

<http://localhost:5000/c/plugin/list/Trace>

这里Source就是Trace，这是TraceModule的注册名。

## 和Toolkit5 .c的Url区别

熟悉Toolkit5 .c的url，需要注意一些问题。Toolkit5.5**不再支持url的省略**，即ModuleCreator和页面类型不能省略，是什么就是什么。此外，Source不再用\_替换/。