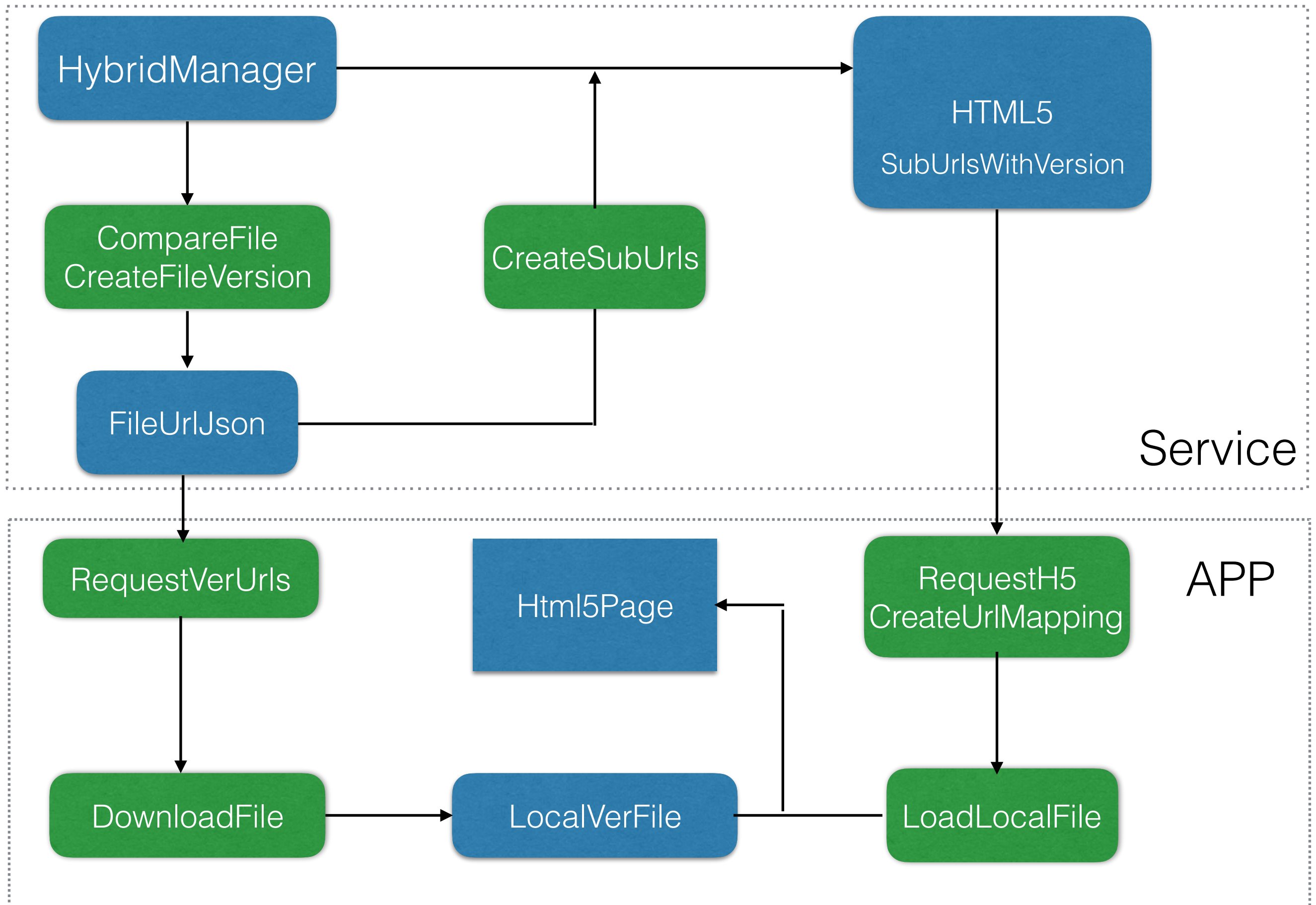


预加载框架总体架构图



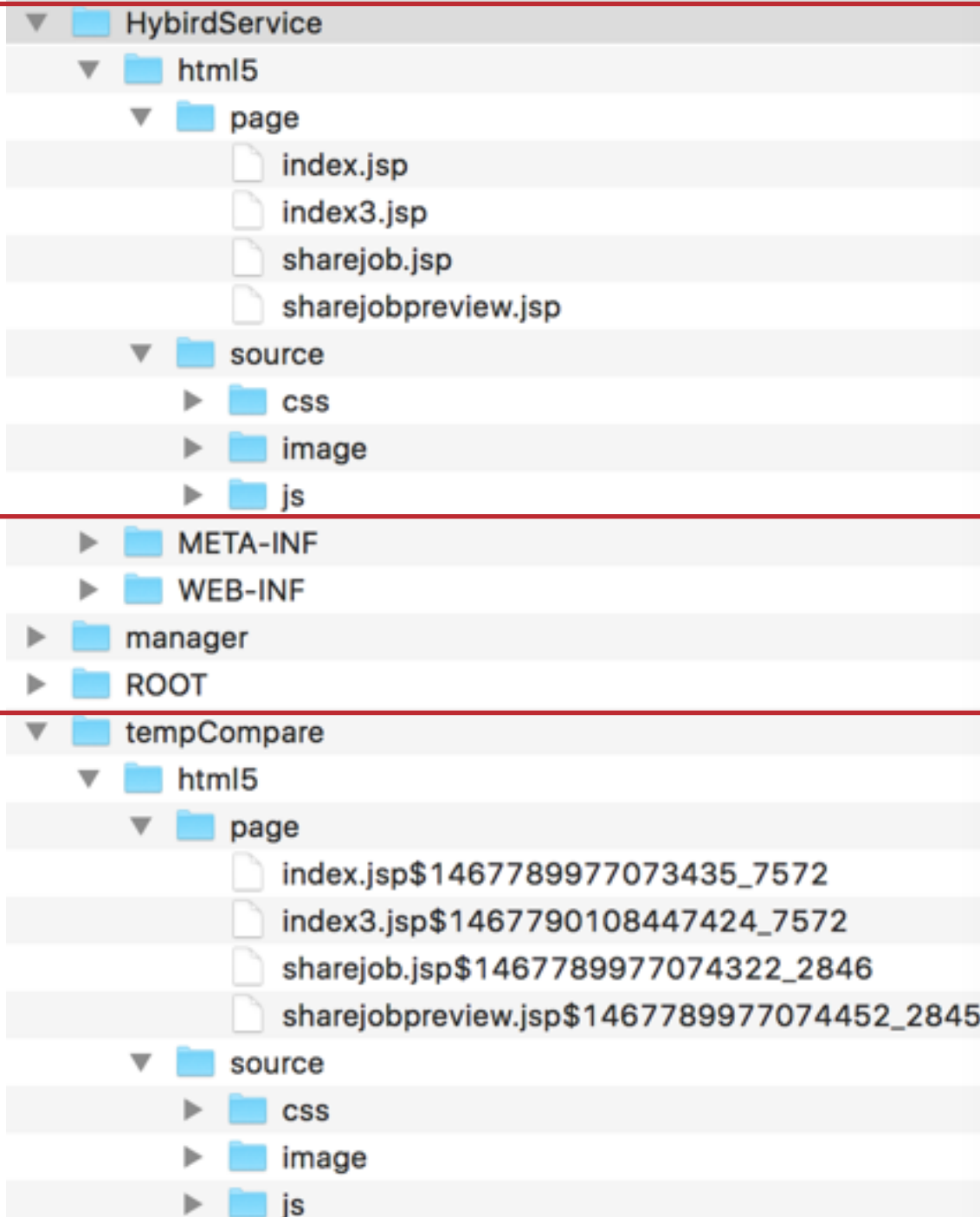
CompareFile&CreateFileVersion

文件比对以及生产版本号

版本号生产规则

根据原文件创建比对文件副本，如果有比对文件并且文件大小以及文件内容一样则认为版本没有变化，否则认为文件有版本更新添加文件后缀为
filename+\$+当前日期+1000内随机数+_文件大小

为了防止tempCompare被删除，目录放到了项目目录的同级目录





FileURLJSON格式以及RequestVerUrls

RequestVerUrls:FileUrlJson

```
{
  $_$/html5/page/index.jsp$1467789977073435_7572,
  /html5/page/sharejobpreview.jsp$1467789977074452_2845,
  /html5/source/css/index.css$146778997707596_1554
}
```

以后的扩展

RequestVerUrls:FileUrlJson

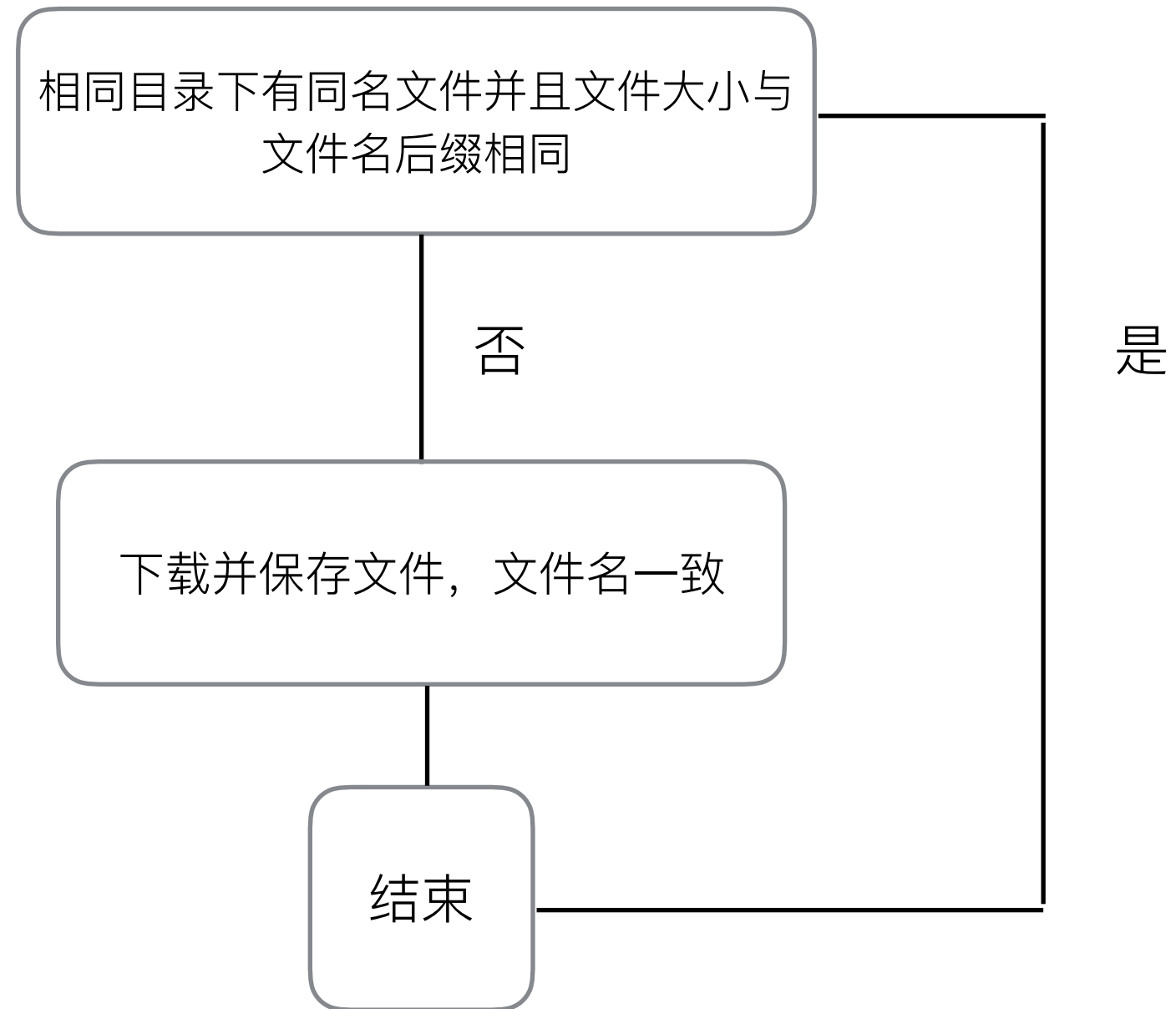
```
{
  {page1:[
    /html5/page/index.jsp$1467789977073435_7572,
    /html5/page/sharejobpreview.jsp$1467789977074452_2845,
    /html5/source/css/index.css$146778997707596_1554
  ]}

  {page2:[
    /html5/page/index.jsp$1467789977073435_7572,
    /html5/page/sharejobpreview.jsp$1467789977074452_2845,
    /html5/source/css/index.css$146778997707596_1554
  ]}
}
```



DownloadFile

FileName: /html5/page/index.jsp\$1467789977073435_7572



CreateSubUrl 生产目标Html所需的URL

正则URL检索

```
public static final String URL_RULE = "(url){1}[\\s]*[\\(]{1}[^\\)]*[\\)]{1}";  
public static final String HREF_RULE = "(href){1}[\\s]*[\\=]{1}[\\s]*[\"']{1}[^\"']*[\"']{1}";  
public static final String SRC_RULE = "(src){1}[\\s]*[\\=]{1}[\\s]*[\"']{1}[^\"']*[\"']{1}";
```

URL过滤

url (../../css/index.css) ——> ../../css/index.css

URL全路径过滤

../../css/index.css——>html5/source/css/index.css

父节点匹配

index.jsp:[html5/source/css/index.css]
index.css:[html5/source/image/bg.png]

根节点匹配

{index.jsp:[html5/source/css/index.css,html5/source/image/bg.png]
index2.jsp:[html5/source/css/index.css,html5/source/image/bg.png]}

版本号过滤

{index.jsp:[html5/source/css/index.css,html5/source/image/bg.png\$1204505_12]
index2.jsp:[html5/source/css/index.css,html5/source/image/bg.png\$1204505_12]}



RequestHTML5&CreateURLMapping 请求HTML5并建立URLMapping映射

服务器内存数据:

```
{index.jsp:[html5/source/css/index.css,html5/source/image/bg.png$1204505_12]  
index2.jsp:[html5/source/css/index.css,html5/source/image/bg.png$1204505_12]}
```

RequestHtml(index.jsp),服务器植入对应URLS 并且放到头部 保证最先加载
<link rel="stylesheet" href="<%=session.getAttribute("urls") %>">

拦截URL link 资源请求 并加同步锁创建URLMapping{
key :html5/source/css/index.css,html5/source/image/bg.png
value :html5/source/css/index.css,html5/source/image/bg.png\$1204505_12}

根据原始URL查找: html5/source/css/index.css,html5/source/image/bg.png\$1204505_12
如果有匹配URL 并且本地目录有同名文件 并且文件大小等于文件后缀那么就读取本地资源 否则读取网络资源

还需要完善的地方

1代码风格的完善

2实际使用的服务端框架整合

3url检索

4服务端和客户端垃圾文件清理机制

5文件下载分C/B 端,ISO/Android端,优先级规则

6稳定性测试

7如需要 支持服务器热部署