**请求响应用例（修订）**

**角色：web应用使用者（【角色类别】：web应用使用者，web应用开发者（mvc框架使用者））**

**子用例１：输入web应用的域名，得到web主页。**

**子用例２：输入web应用的url对应资源不存在，返回人性化提示的404页面。**

**子用例３：输入web应用的url对应资源存在【不带参数】，返回对应的html,xml,txt,json,jsp。**

**子用例４：输入web应用的url对应资源存在【带合法参数】，返回经过参数处理的对应的html,xml,txt, json,jsp。**

**子用例５：输入web应用的url对应资源存在【带非法参数】，返回提示非法参数的405错误页面。**

**子用例６：长时间的保持登录状态。（cookie / session)**

**子用例７：隐私保护。对数据进行加密。**

**子用例８：快速响应。尽快的返回页面。（页面缓存，计算，网络带宽）**

**涉及的类：**

**Cookies, Sessions,Cryptor,HomePage,404Page, 405Page,**

**HttpRequestParser, HttpResponseWrapper, UrlHandler**

**【用例】：**

**请求响应：**

**DO:**

**in:用户输入**

**if 输入web应用的域名**

**得到web主页**

**else if 输入web应用的url且对应资源不存在**

**返回人性化提示的404页面。**

**else if 输入web应用的url对应资源存在【不带参数】**

**返回对应的html,xml,json,jsp**

**else if 输入web应用的url对应资源存在【带合法参数】**

**返回经过参数处理的对应的html,xml,json,jsp。**

**else if 输入web应用的url对应资源存在【带非法参数】**

**返回提示非法参数的405错误页面。**

**长时间的保持登录状态。设置Cookie和Session。**

**隐私保护。对传输的数据加密。**

**快速响应。尽快的交付页面。**

**【时序图语言描述】**

**in:用户输入【method: sendRequest, Class: UserAgent】->**

**if 输入web应用的域名**

**得到web主页**

**【method:isDomain, Class:HttpRequestParser】->(msg:是域名）-> 【method: getPage, Class: HomePage】**

**else if 输入web应用的url且对应资源不存在**

**返回人性化提示的404页面。**

**【method:isUrl, Class:HttpRequestParser】->(msg:是url)-> 【method: validateUrl, Class: UrlHandler】->(msg:url资源不存在)->【method: getPage, Class: 404Page】**

**else if 输入web应用的url对应资源存在【不带参数】**

**返回对应的html,xml,json,jsp**

**【method:isUrl, Class:HttpRequestParser】->(msg:是url且不带参数)-> 【method: validateUrl, Class: UrlHandler】->(msg:url资源存在)->【method: getPage, Class: Page】->(msg:返回对应Page)->【method: returnPage, Class: UrlHandler】**

**else if 输入web应用的url对应资源存在【带合法参数】**

**返回经过参数处理的对应的html,xml,json,jsp。**

**【method:isUrl, Class:HttpRequestParser】->(msg:是url且带参数)-> 【method: validateUrl, Class: UrlHandler】->(msg:url资源存在且参数合法)->【method: getPage, Class: Page】->(msg:返回对应Page)->【method: returnPage, Class: UrlHandler】**

**else if 输入web应用的url对应资源存在【带非法参数】**

**返回提示非法参数的405错误页面。**

**【method:isUrl, Class:HttpRequestParser】->(msg:是url且带参数)-> 【method: validateUrl, Class: UrlHandler】->(msg:url资源存在且参数非法)->【method: getPage, Class: 405Page】->(msg:返回对应Page)->【method: returnPage, Class: UrlHandler】**

**【公共序列】**

**->(msg:返回了页面内容)->【method:wrap(), Class: HttpResponseWrapper】->(msg:返回了http响应）->【method:recieveResponse, Class: UserAgent】**

**【以下几个功能并行，理论上具有全过程生命周期】**

**长时间的保持登录状态。设置Cookie和Session。**

**【method:addCookie, Class:Cookies】,【method:addSession, Class:Sessions】**

**隐私保护。对传输的数据加密。**

**【method:encrypt(), Class:Cryptor】，【method:decryptor,Class:Cryptor】**

**快速响应。尽快的交付页面。（页面缓存，计算，压缩，网络带宽）【未列出可能的相关类】**