# Mock Server

V 1.2.1 by Allen Chen

目录

[Mock Server 1](#_Toc399745590)

[功能 2](#_Toc399745591)

[原理 2](#_Toc399745592)

[前提 2](#_Toc399745593)

[使用方法 2](#_Toc399745594)

[设置主映射规则 2](#_Toc399745595)

[设置响应映射文件 4](#_Toc399745596)

[映射文件和响应文件修改 5](#_Toc399745597)

[HTTPS支持 5](#_Toc399745598)

[限制 5](#_Toc399745599)

[与Fiddler工具共同使用 5](#_Toc399745600)

[伪造不存在或无法访问的域名的响应 6](#_Toc399745601)

## 功能

用户可以自定义在本机发送HTTP请求时，某些URL所返回的响应。支持HTTPS。

## 原理

## 前提

本机的9999端口没有被占用

Win 7以上

## 使用方法

### 设置主映射规则

在程序根目录下找到Rules.txt,设置规则。主映射规则表示某些URL的响应消息为所指定文件的内容

#### 优先级

Mock Server从上往下匹配，寻找到第一个匹配命中即返回相应结果

#### 语法

##### 精确匹配

示例（注:url和文件之间为tab,不是空格）：

https://www.jd.com/hello?a=2 2.txt

对https://www.jd.com/hello?a=2的请求将返回Response文件夹下的2.txt里的内容

##### 匹配开头

示例（注:url和文件之间为tab,不是空格）：

[startwith]https://www.jd.com/hello 2.txt

以https://www.jd.com/hello开头的请求将返回Response文件夹下的2.txt里的内容

##### 匹配包含

示例（注:url和文件之间为tab,不是空格）：

[contains] hello 2.txt

url中包含hello的请求将返回Response文件夹下的2.txt里的内容。在使用包含匹配的时候需要注意，尽量放在规则的最下方并且尽可能设置大多数情况下唯一的匹配字符串，以免因为优先级的原因篡改了其他非期望的请求的响应。

##### 附加匹配请求Body

示例（注:url和文件之间为tab,不是空格）：[startwith]http://www.thisisfakedomain123.com[contains]<funcId>10003 s.txt

url中以<http://www.thisisfakedomain123.com>开头的请求，并且请求body中包含<funcId>10003的将返回s.txt中的内容。

##### 双文件映射

示例（注:url和文件之间为tab,不是空格）：

https://www.jd.com/hello h.txt,b.txt

对https://www.jd.com/hello请求将返回:

HTTP头：Response文件夹下的h.txt里的内容

HTTP Body: Response文件夹下的b.txt里的内容

##### Verb过滤

示例（注:各个字段之间为tab,不是空格）：

http://www.baidu.com/ 1.txt GET

仅对http://www.baidu.com/的GET请求返回1.txt里的内容

##### 模拟网络延时

示例（注:各个字段之间为tab,不是空格）：

http://www.baidu.com/ 1.txt +20s

http://www.baidu.com/ +20s

http://www.baidu.com/ +200ms

在单条规则的最后附加一个+20s字段可以模拟增加20秒网络延时。支持秒，毫秒。

##### 响应内容替换

示例（注:各个字段之间为tab,不是空格）：

[startwith]https://www.jd.com/ [replaceStr]JD.COM[with]京东

表示url以https://www.jd.com 开头的请求，所有response中的JD.COM全部替换成京东。

[startwith]https://www.jd.com/ [replaceRegex] <meta([^<]\*)charset=([^<]\*)[\"'] [with]京东

表示url以https://www.jd.com 开头的请求，所有response中的符合(<meta([^<]\*)charset=([^<]\*)[\"'])正则表达式内容的全部替换成京东。

##### 启用请求和响应转换器

此功能适用于需要对请求和响应做特殊处理的情况。一个典型场景是，请求和响应都被加密，但是在设置映射规则的时候，期望请求的Body中包含指定字符才进行响应篡改，并且期望在设置篡改的响应文件的时候使用明文。此时可以使用此功能进行请求和响应篡改。

可以使用如下开关开启此功能：

<add key="EnableConvertService" value="true"/>

并且可以选择是否仅对请求/响应使用转换器：

<add key="ConvertMode" value="Both"/>

支持的选项有：

* Both
* ReqOnly
* RespOnly

要使用此功能，可以在Rules规则中的最后，延时规则（如包含）之前，增加[convert]标记：

示例（注:各个字段之间为tab,不是空格）：

[startwith]http://www.thisisfakedomain123.com[contains]<funcId>10003 s.txt [convert]

此设置说明，对于<http://www.thisisfakedomain123.com>开头的请求，先调用请求转换器进行请求Body的转换，再检查请求Body(转换后的)是否包含<funcId>10003，如果包含，则将s.txt的内容调用响应转换器得到转换的内容，将转换的内容作为最后的响应返回。

要使用此功能需要额外编写一个符合规范的RESTful Service。并在Mock Server启动时运行该Service。

**请求转换器RESTful规范：**

Mock Server的App.Config中可以设置接口地址。例如：

<add key="ConvertReqServiceUrl" value="http://localhost:18989/ConvertSvc/ConvertReq"/>

该接口接受POST请求，将请求Body中内容进行转换并返回

**响应转换器RESTful规范：**

Mock Server的App.Config中可以设置接口地址。例如：

<add key="ConvertRespServiceUrl" value="http://localhost:18989/ConvertSvc/ConvertRespEncrypt"/>

该接口接受POST请求，将请求Body中内容进行转换并返回

注释

示例（注:各个字段之间为tab,不是空格）：

#http://www.baidu.com/ 1.txt

#这是注释

在单条规则的最前面附加’#’此条规则即为注释不被应用。也可以用注释来增加说明文字。

### 设置响应映射文件

所有的响应映射文件需要放到程序根目录下Response文件夹下

#### 单文件

单文件模式映射中，映射的文件必须包括HTTP头和Body。格式跟Fiddler抓包获取的Response的Raw data完全一致。示例：

HTTP/1.1 200 OK

Cache-Control: private

Content-Length: 15445

Content-Type: text/html; charset=utf-8

Expires: Tue, 10 Jun 2014 08:16:09 GMT

Vary: Accept-Encoding

Server: BWS/1.1 Microsoft-HTTPAPI/2.0

X-Powered-By: HPHP

Date: Tue, 10 Jun 2014 08:15:53 GMT

haha

其中第一行表示HTTP状态码，之后到空行前表示HTTP头，空行后表示HTTP Body

#### 双文件模式

与单文件模式类似，区别是header和body分两个文件存放。此模式适用于图片等的mock。

### 映射文件和响应文件修改

#### Rules文件：

对文件内容的修改可以立即生效

#### Response文件夹下的响应文件：

对文件内容的修改可以立即生效

### HTTPS支持

请参考Fiddler相关HTTPS支持文档

### 限制

Content-Length头和HTTP状态码描述目前均不能修改。Content-Length会强制设置为修改后响应的大小。HTTP状态吗描述目前设置的修改无效。

### 与Fiddler工具共同使用

当需要使用Fiddler来观察被Mock Server篡改的数据的时候，请按照以下顺序执行：

1. 启动Mock Server
2. 启动Fiddler。需要确保Fiddler选项中Connections->Act as system proxy on startup没有勾选
3. 确保app使用Fiddler端口为HTTP代理
4. 确保Fiddler选项中Gateway->Manual Proxy Configuration配置为：<http://localhost:9999>。其中9999为Mock Server端口。

### 伪造不存在或无法访问的域名的响应

Mock Server启动时会在本地80端口尝试开启一个HTTP服务器，接受所有请求并返回简单的响应。因此可以通过修改本地hosts文件将需要访问的域名映射到127.0.0.1来完全绕开真实服务器进行响应伪造。