

如何启动F12

- 键盘按F12启动
- 鼠标右键选检查页面启动
- 菜单里更多工具里->开发者工具
- ctrl+ shift+i快捷键启动

Elements(查看器)

用途

1. 查看元素代码
2. 查看元素属性
3. 修改元素代码和属性(仅对当前页面生效)

左上角箭头 / ctrl+shift+c: 选取页面元素

UI自动化测试工具: selenium 语言: python/java

八大元素定位方法

1. id定位元素
2. name定位元素
3. Xpath定位元素
4. class_name定位元素
5. tag_name定位元素
6. link_text定位元素
7. partial_link_text定位元素
8. css选择器定位

Console(控制台)

用途: --开发用的测试

打印控制台的查看, 用于测试查看报错内容(前端页面报错), 前端开发编写js脚本调试前端代码, 或则打印分析问题

1. console里验证xpath表达式: `$x('//input[@name="phone"]')` UI自动化测试
2. 执行js语句
3. 查看cookies信息: `document.cookie`

Source(源代码/调试器)

查看css, js等文件源码

Network(网络)

记录前后端的交互, 接口测试, 抓包工具

http请求消息: 请求行(请求方法)、请求头、请求体--客户端发送

http响应消息: 响应行(响应状态码), 响应头, 响应体--服务器回应

抓到报文分析

- 状态: http的响应状态码: 200(正常)、404(资源不存在或地址错误)、500/503(服务器问题)、302/304(重定向)
- 方法: get post put patch
- 域名: 域名/IP+端口
- 文件和类型: html css js
- 发起者: 请求怎么发起的, 比如script: 页面是由script脚本处理时发送的
- 传输和代销: 传输的文件及大小
- 消息头: 响应头、请求头

详细报文内容:消息头, cookies, 请求, 响应, 耗时

login请求抓包不到: 打开保留日志

禁用缓存: 开启后, 页面资源不会存入缓存

如何定位前端和后端问题:

1.明显的前端js问题

2.抓包分析:

- 前端没有发送请求 --前端问题
 - 前端发送请求了, 但是前端发送内容的数据错误 --前端问题
 - 前端发送了正确请求, 后端没有给响应消息 --后端问题
 - 前端发送了正确请求, 后端也响应消息了, 但是响应的数据错误 --后端问题
 - 前端发送了正确请求, 后端给出了正确响应, 但前端没有正确显示 --前端问题
- 有时需要配合数据库确认+Linux服务器分析项目日志

搜索和过滤

1. 请求过滤

HTML、CSS、JS等

XHR: XMLHttpRequests

WS: WebSocket连接

2. 报文过滤 ctrl+f: 调用搜索框

url: 域名或文件部分包含这个url

status-code: 404

method: post

domain: 8.129.91.152.8765

remote-ip: 124.237.176.160

mime-type: image/png

接口测试和数据篡改

火狐右键编辑并重发

chrome右键以fetch格式复制, 控制台修改

以cURL(bash)复制可导入到posman

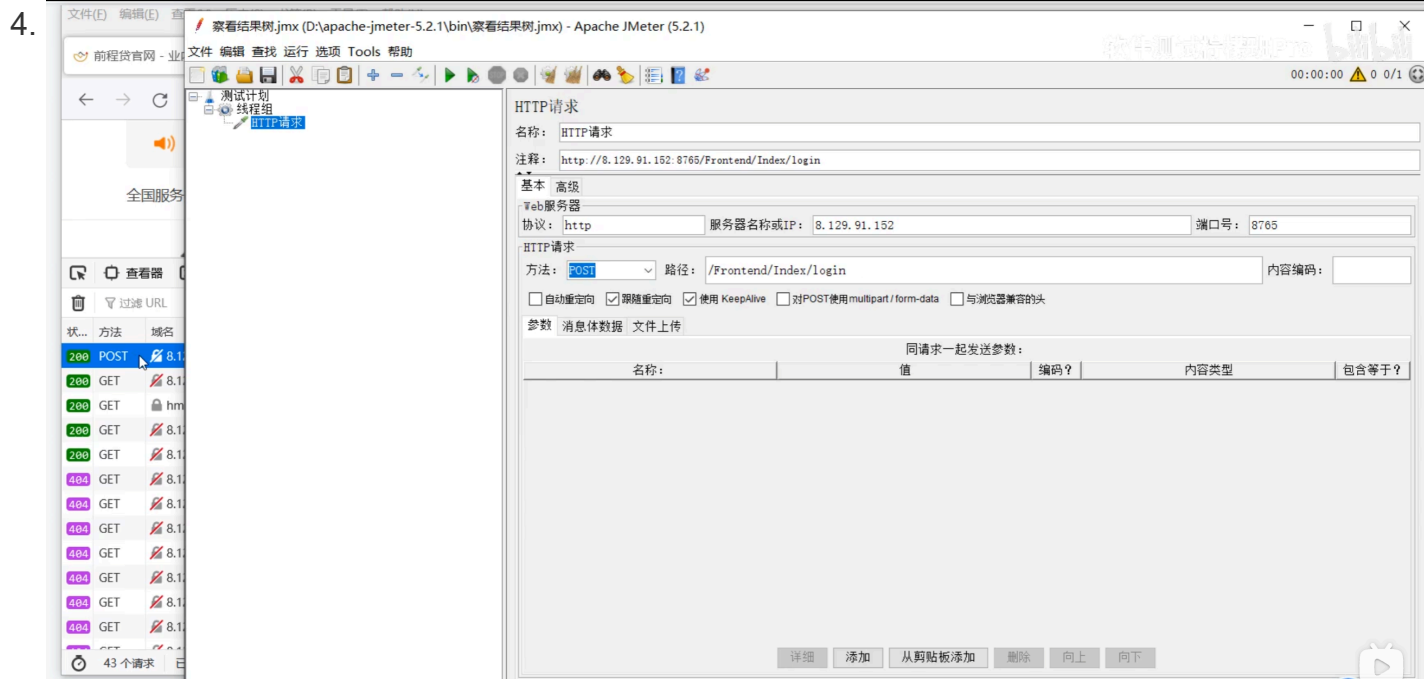
返回网络检查响应结果

jmeter安装, jdk环境配置

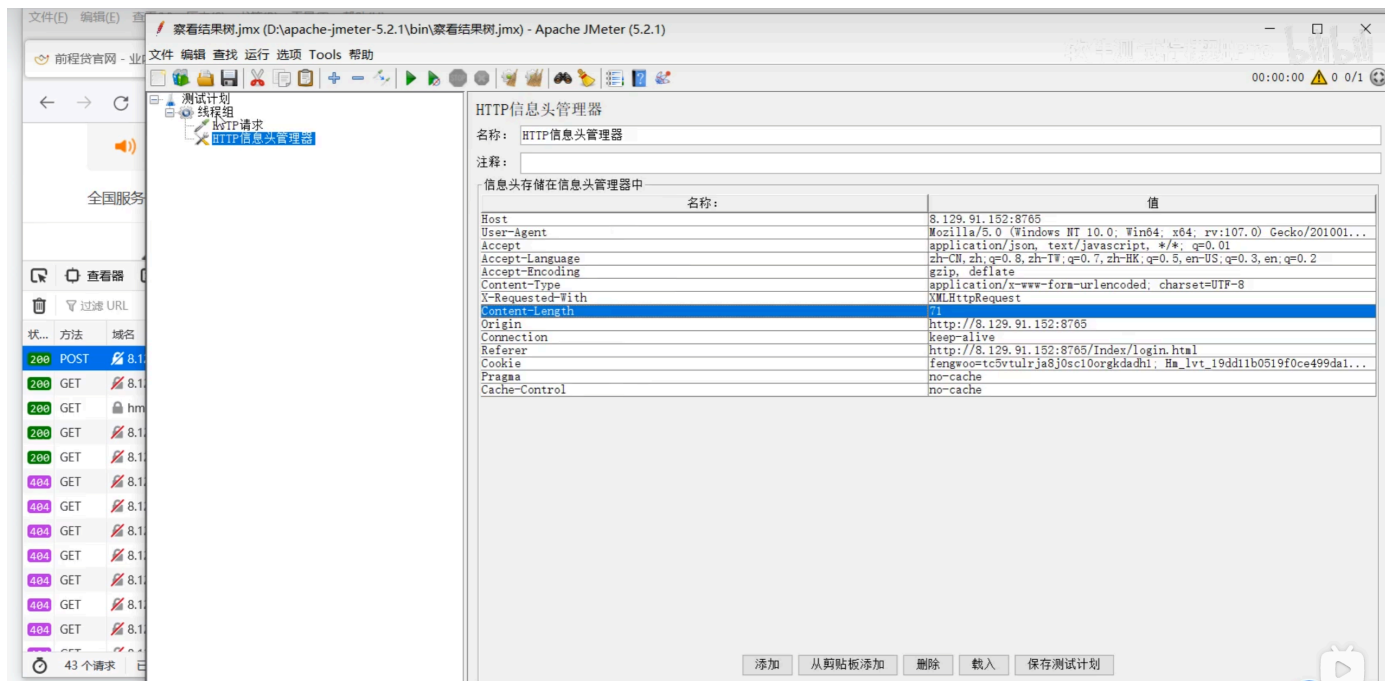
1. 添加线程组

2. HTTP请求

3. 复制网址到注释



5. 复制post参数，从剪贴板添加
6. 右键线程组，配置元件HTTP cookie管理器，打开
7. 复制请求头，从剪贴板添加



8. 右键线程组添加监听器查看结果树，点击查看结果树，再点击运行

导出/入HAR文件

弱网测试

2G

3G

离线

4G

wifi等

用途: 模拟用户网络不佳的场景(web应用 手机模式), 保证软件可以正常处理并给出友好的反应和提示信息。

禁用缓存右侧无限制切换4G、3G、脱机状态、自定义测试预设情况

录制脚本和回放

录制脚本的用途 (导入和导出) :

测试的时候一些重复性的工作可以录制下来进行回放;也可以保存后导入进行replay (重放)。

- 1、录制并进行replay
- 2、导出给开发进行replay复现bug
- 3、导入进行replay
- 4、还可以进入分析性能页面

可导出/入json格式文件

Performance(性能)

性能测试分为前端性能 (页面的性能) 和后端性能 (接口-jmeter等)

Performance/性能测试

用途: 监控该网页的性能各种指标信息

场景: 当我们在进行大并发的直播互动这种类似的场景下, 对浏览器的负荷是比较大的, Performance对网页进行分析就可以评估实时网页性能以及相关的数据。(直播或者游戏类)

案例: 点击●可以开始录制, 它会记录用户的交互以及这些交互对页面性能数据的影响, 当交互完成后, 点击Stop来停止Record ,

Performance面板会展示出刚才录制的页面性能数据

得到如下的性能数据:

火焰图: 显示性能报告, 我们根据性能报告用来分析页面得性能, 从而找到性能瓶颈

FPS (帧率): 主要和动画性能有关, 代表每秒帧数。图表中的绿色长条越高, 说明FPS越高, 用户体验越好。如果其中有红色长条, 代表着这部分帧数有卡顿, 需要优化

NET (网络): 主要展示了网络请求的先后顺序以及各自的请求耗时, 可以被Network面板完美替代, 建议直接查看Network

summary (摘要): cpu图表的各种颜色表示在这个时间段内, CPU在处理各种时间, 上面所花费的时间, 如果你看到某段颜色占据了大量的比重, 就说明这个地方可能存在优化的空间

Main (主要) : 主要用来分析页面性能的部分, 它展现了主线程在Record过程中做的所有事情, 包括: Loading、Scripting、Rendering等; 横轴代表着时间, 纵轴代表着调用堆栈。每一个长条代表执行了一个事件或函数, 长条的长度代表着耗时的长短。如果某个长条右上角是红色的则表示该函数存在性能问题, 需要重点关注。

导出&导出火焰图: 导入和导出脚本进行分析 --提供给开发进行问题分析/测试导入进行问题复现

Memory(存储)

用途: 主要存储一些cookies和前端数据等。

本地存储和会话存储主要是前端开发人员在前端设置, 一旦数据保存在本地后, 就可以避免再向服务器请求数据, 因此减少不必要的请求, 减少数据在浏览器和服务端间不必要地来回传递。 ---提高浏览器访问速度

本地存储和会话存储的操作代码完全相同, 它们的区别仅在于数据的寿命。

本地存储: 保存不受时间限制, 用于长期保存网站的数据; 除非用户自己删除, 否则不会消失; 并且站内任何页面都可以访问该数据; 主要用来保存访客将来还能看到的数据。

会话存储: 用于临时保存针对一个窗口(或标签页)的数据, 保存受时间限制, 当前窗口一旦关闭, 内容就会被浏览器删除。会话存储则用于保存那些需要从一个页面传递给下一个页面的数据。

案例场景: 比如页面上输入一些内容信息(注册信息), 中途被打断, 页面关闭, 然后重新打开, 能保留上次输入的内容。

Cookies: 是下载到您计算机上的小文件。比如访问过的链接, 单击的按钮, 登录信息或过去访问过的页面, 当我们下次访问该网站时, 该网站将使用该信息来跟踪我们先前在其网站上的活动。它还可以记住以前输入到表单字段中的信息, 例如姓名, 日期, 出生详细信息, 信用卡号等。

Cookies的组成部分:

expires =date:表示cookie的失效时间, 默认是浏览器关闭时失效(可选) --默认是会话 (和服务器的session时效同步) --session相关

path=路径:访问路径, 默认为当前文件所在目录(可选); 设置了之后只有设置的那个路径文件 才可以访问cookies

domain=域名:访问域名, 限制在该域名下访问(可选) =设置的域名才可以访问cookies 没有设置默认当前域名

secure:安全设置, 如果设置了则必须使用https协议才可获取cookie(可选)

cookies用户鉴权: 用户信息以session (会话--失效) 形式保存到服务器里, session id通过set cookie 字段发送给客户端, 以cookie形式保存到浏览器