应用系统性能测试实验

# ****1.目标****

教学目标是学生通过学习应用系统性能测试工具WebRunner接口录制、脚本编辑、脚本增强、脚本调试、性能测试、测试报告完整的测试流程，掌握性能测试核心概念（Http接口、测试脚本、负载、事务、TPS、RPS、响应时间）和性能优化的相关技能，培养系统性能瓶颈分析思维与测试报告解读能力。

* 掌握性能测试核心概念（Http接口、测试脚本、负载、事务、TPS、RPS、响应时间）；
* 掌握性能测试完整流程,独立完成测试脚本录制、脚本增强、脚本调试、配置测试负载、设计测试场景, 采集性能测试指标和生成专业测试报告；
* 培养系统性能瓶颈分析思维与测试报告解读能力

# ****2.知识点****

性能测试核心在于评估系统在压力下的表现，基础是理解通信协议（如HTTP用于请求/响应交互，JSON作为主流数据格式），并据此构建测试脚本，通过参数化（模拟多用户数据）、关联（处理动态值如Session）、逻辑控制和断言（验证正确性）进行脚本增强，并通过脚本调试确保测试脚本的正确性；测试的核心是设计和施加负载（如并发用户数、请求速率），并规划测试场景（如梯形负载、阶梯负载、循环负载）来模拟真实压力模式，通过采集关键指标（如平均响应时间、吞吐量（TPS/RPS）、失败率）和系统监控资源利用率（CPU、内存、I/O、网络），综合分析定位系统性能瓶颈、验证性能目标并为系统优化提供数据支撑，驱动系统性能的持续改进。

## 2.1 http协议

应用层协议，用于客户端（如浏览器、性能测试工具模拟的虚拟用户）和服务器之间传输超文本（如 HTML）和其他资源（如图片、JSON 数据等）的通信基础。它是 Web 应用和 API 交互的基石（特别是 RESTful API）。

* ****模拟真实用户行为：**** 绝大多数 Web 应用的交互都基于 HTTP(S)。性能测试工具必须能准确模拟客户端发送 HTTP 请求（GET, POST, PUT, DELETE 等）和接收 HTTP 响应的过程。
* ****理解请求/响应结构：**** 测试脚本需要构建正确的 HTTP 请求（包含 URL、Headers、Cookies、Body 等），并解析响应（状态码、Headers、Body）。错误的结构会导致脚本失败或不真实的负载。
* ****识别性能瓶颈：**** 分析 HTTP 交互（请求发送时间、接收到第一个字节时间、接收完成时间）是定位网络延迟、服务器处理时间、响应大小等问题的关键。
* ****状态管理：**** HTTP 本身是无状态的，但 Web 应用常使用 Session（通常依赖 Cookies）或 Token（通常在 Header 中）来维持状态。脚本必须能处理这些机制（如自动管理 Cookie 或手动设置 Token）

## 2.2 JSON协议；

JSON协议一种轻量级的数据交换格式。它使用易于人阅读和编写的文本格式，同时也易于机器解析和生成。它基于键值对的结构。

* ****API 交互的主要载体：**** 现代 Web API（尤其是 RESTful API）几乎都使用 JSON 作为请求体（Request Body）和响应体（Response Body）的标准格式。
* ****脚本参数化和数据驱动：**** 测试数据（如用户信息、产品 ID）常存储在 JSON 文件中，供脚本读取和使用。
* ****响应验证和关联：**** 需要解析 JSON 格式的响应，提取关键数据（如订单号、认证令牌）用于后续请求（关联），或验证响应内容是否符合预期。

## 2.3. 脚本增强

脚本增强是对录制的基础的测试脚本进行修改和优化，使其更健壮、更灵活、更符合实际业务逻辑，并能模拟更复杂的用户行为，设计符合业务逻辑的测试脚本。

* ****处理动态数据：**** 应用往往包含会话 ID、令牌、时间戳等动态值，录制脚本无法直接重用。增强（主要是关联）解决此问题。
* ****参数化：**** 避免所有用户使用相同数据（导致缓存失效或数据冲突），使用不同数据模拟真实用户（如不同用户名、搜索关键词）。
* ****添加逻辑：**** 模拟用户决策（如根据前一个响应结果决定下一步操作）、循环、条件判断、等待时间（思考时间）。
* ****增加检查点：**** 验证服务器返回的内容是否正确，而不仅仅是 HTTP 状态码 200。
* ****错误处理：**** 使脚本在遇到非预期响应时能优雅处理（如重试、记录错误、停止迭代）。
* ****模块化和重用：**** 将通用功能（如登录）封装，便于复用和维护。

## 2.4. 脚本调试

脚本调试是指在正式执行大规模负载测试之前，通过小规模、可控的运行来验证单个虚拟用户（单线程/单用户）执行脚本的正确性、逻辑和数据处理是否如预期。目的是确保测试脚本的正确性。

* ****确保测试脚本逻辑正确：**** 避免将脚本本身的错误（如错误的 URL、参数、关联失败、断言错误）带入负载测试，导致结果无效或难以分析。
* ****验证关联和参数化：**** 确认动态数据是否被正确提取和替换。
* ****验证业务逻辑：**** 检查条件判断、循环等逻辑是否按预期执行。
* ****观察请求/响应细节：**** 查看实际的请求内容（Headers, Body）和完整的响应内容（Headers, Body），确保与预期一致。
* ****提高效率：**** 尽早发现并修复测试脚本可能存在的问题，比在负载测试运行中或运行后才发现要节省大量时间和精力。

## 2.5. 测试负载

测试负载指在性能测试中模拟并发用户施加到被测系统上的压力或工作量，量化了有多少“用户”或“操作”在同时（或一定速率下）访问系统。

* ****定义测试强度：**** 负载是性能测试的核心，不同的负载水平（如 100 并发用户 vs 10000 并发用户）会揭示系统在不同压力下的表现。
* ****目标导向：**** 负载设计应基于测试目标（如验证系统能否处理预期高峰流量、找出最大容量、评估稳定性）。
* ****模拟真实场景：**** 负载模型应尽可能贴近生产环境的用户行为模式（用户数、操作频率、业务混合比例）。

## 2.6. 测试场景

测试场景定义了性能测试的具体执行方式和负载变化模式。它描述了负载如何随时间施加到系统上以及模拟了哪些用户行为组合。

* ****模拟真实世界：**** 用户访问不是恒定不变的。场景模拟了高峰、低谷、平稳期、突发流量等真实模式。
* ****测试特定目标：****
* ****基准测试：**** 低负载，建立性能基线。
* ****负载测试：**** 模拟预期或略高于预期的负载，评估常规性能。
* ****压力测试：**** 逐步增加负载至超过系统容量，找出瓶颈和极限。
* ****稳定性/耐力测试：**** 长时间（数小时或数天）施加稳定负载（通常是预期高峰），检查是否有内存泄漏、资源耗尽等问题。
* ****尖峰测试：**** 短时间内突然施加极高负载，测试系统的弹性。
* ****组合业务：**** 一个场景通常包含多个不同的业务操作（如登录、增加、删除、查询、修改），并按一定比例混合。

## 2.7. 性能测试指标

性能测试指标用于量化衡量系统性能表现和资源使用情况的核心，主要用户评估测试结果、识别瓶颈、促进系统性能优化。

* ****事务指标：****
* ****响应时间：**** 用户发起请求到收到完整响应所经历的时间。关键指标包括：平均响应时间、最小响应时间、最大响应时间、百分位响应时间（如 T50响应时间, T90响应时间等）。
* ****事务吞吐量：**** 单位时间内系统成功完成的事务数量（Transactions Per Second - TPS），衡量系统处理能力。
* ****事务错误率：**** 失败的事务数占总事务数的百分比，高于阈值通常表示系统不稳定或过载。
* ****资源指标 ：****
* ****CPU 利用率：**** 服务器 CPU 忙碌的程度。持续高利用率（如 > 80%）可能成为瓶颈。
* ****内存利用率：**** 物理内存和虚拟内存的使用情况。关注可用内存、Swap 使用率（过高严重影响性能）、内存泄漏趋势。
* ****磁盘 I/O：**** 读写速率（IOPS）、磁盘队列长度、磁盘利用率。高 I/O 等待时间是常见瓶颈。
* ****网络 I/O：**** 网络带宽使用率、网络错误/重传率， 可能成为跨机房或带宽受限系统的瓶颈。

# ****实验环境****

## 3.1 软硬件环境

实验环境的描述需涵盖硬件、软件、运行配置等方面，确保实验可复现、环境稳定。

1）客户端硬件环境：

* PC计算机；
* CPU：主频2.00GHz以上；
* 内存：16G以上；
* 硬盘：500G以上；
* 操作系统：Windows7、Windows 10、Windows 11，64位操作系统/麒麟/统信桌面操作系统；

1. 服务器硬件环境
   * PC计算机；
   * CPU：主频2.00GHz以上；
   * 内存：32G以上；
   * 硬盘：500G以上；
   * 操作系统：Linux/centos/麒麟/统信/欧拉服务器操作系统
2. 软件环境：
   * 浏览器：Chrome 90以上。
   * WebRunner V6.0
   * 日志管理系统 V1.0（内置在WebRunner中）

## 3.2 实验说明

### 3.2.1 业务场景

本测试场景以应用系统最常见的登录接口为常见，以WebRunner工具内置的日志管理系统V1.0为测试样品，评估登录接口的性能表现。

### 3.2.2 工具简介

WebRunner应用系统性能是一款web应用系统性能测试工具。该工具完全基于现实应用系统，通过测试用例及其比例等业务场景设置来模拟现实用户的操作，不断对服务器造成压力进而测试应用系统的性能。工具提供可视化的配置和过程监控，将脚本录制与性能测试进行分离，达到一次录制重复使用的效果，不同的场景设置勾画用户行为，实现测试的快速准确定制，自动化输出报告，快速定位系统性能瓶颈，并提供缺陷建议。



图 3-1功能模块图

### 3.2.3 场景约束

1. 并发用户数：50、100、200。
2. 梯形负载模式，用户创建速率每秒钟5个；
3. 单次测试时长10分钟。

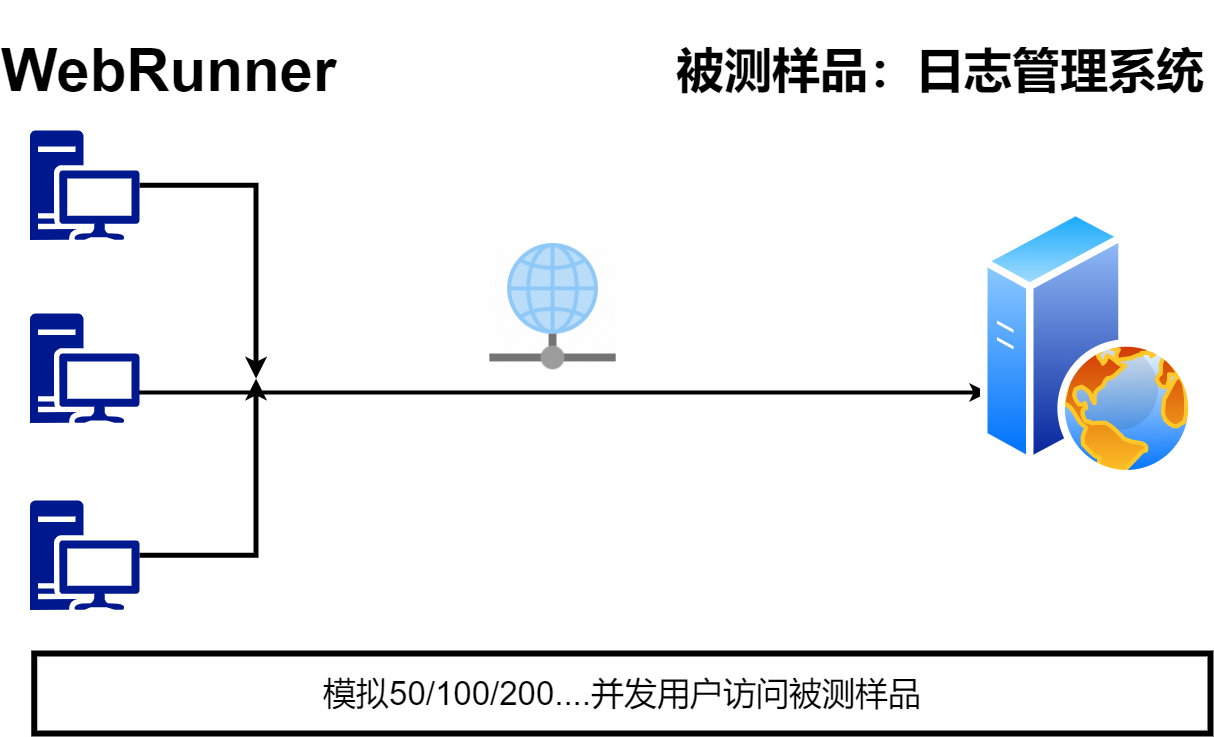


图3-2网络拓扑图

### 3.2.4 性能指标

性能测试核心关注指标:

（1）并发用户数：模拟多个用户同时访问被测系统；

（2）TPS：每秒完成的事务个数；

（3）事务响应时间: 从客户端发出事务到完成的时间；

（4）标准差: 事务响应时间稳定性指标；

（5）错误率: 请求失败的比例，包括超时、无法连接等错误；

### 3.2.5 测试流程

测试分为准备、测试、报告三个阶段：

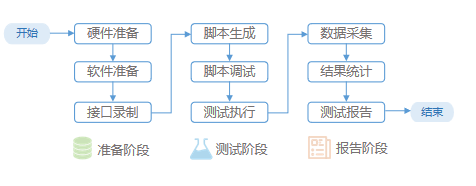


图3-3测试流程

测试流程描述如下：

（1）硬件准备：准备测试需要的测试机、服务器、网络等硬件环境。

（2）软件准备：准备被测样品以及测试工具等软件环境。

（3）脚本生成：生成性能测试脚本。

（4）测试执行：启动测试创建并发用户执行性能压测。

（5）报告生成：根据报告要求，生成测试结果。

### 3.2.6 测试报告

测试报告展示如下信息：

（1）水印；

（2）报告日期；

（3）被测样品、版本等；

（4）测试工具名称、版本等；

（5）测试的开始时间、结束时间、测试时长；

（6）测试环境配置信息；

（7）测试最终结果，包括并发用户数、TPS、事务响应时间、标准差等；

（8）其他重要输出信息。

# ****4.实验步骤****

实验步骤需以清晰、有序的方式呈现操作流程，让学习者能够按步骤完成实验并达成目标。

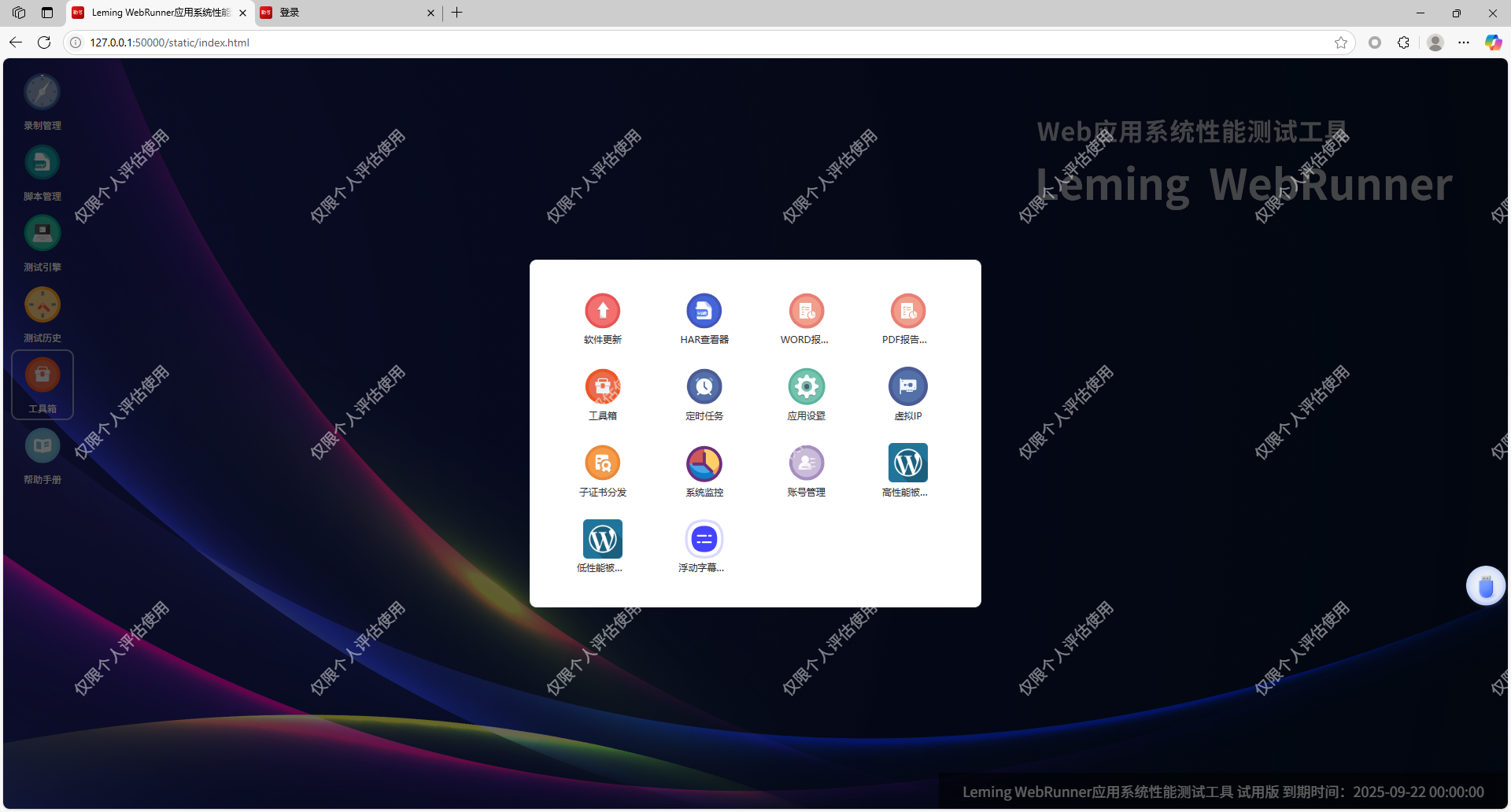
## 4.1测试工具的安装与启动

* 1. Windows系统下的安装与启动

①下载软件安装包，存放至不含中文字符的路径中，解压之后会生成1个文件夹。

②解压文件后打开webrunner-x86\_64-windows文件夹，双击 （webrunner.exe）即可启动工具。

* 1. 访问内置高性能和低性能的被测系统



* 点击【高性能被测系统】---【点击增删查改被测系统】--【点击生成账号数据】--获取【被测地址】：http://192.168.2.156:50050/static/login.html
* 点击【低性能被测系统】---【点击增删查改被测系统】--【点击生成账号数据】--获取【被测地址】：http://192.168.2.156:50051/static/login.html

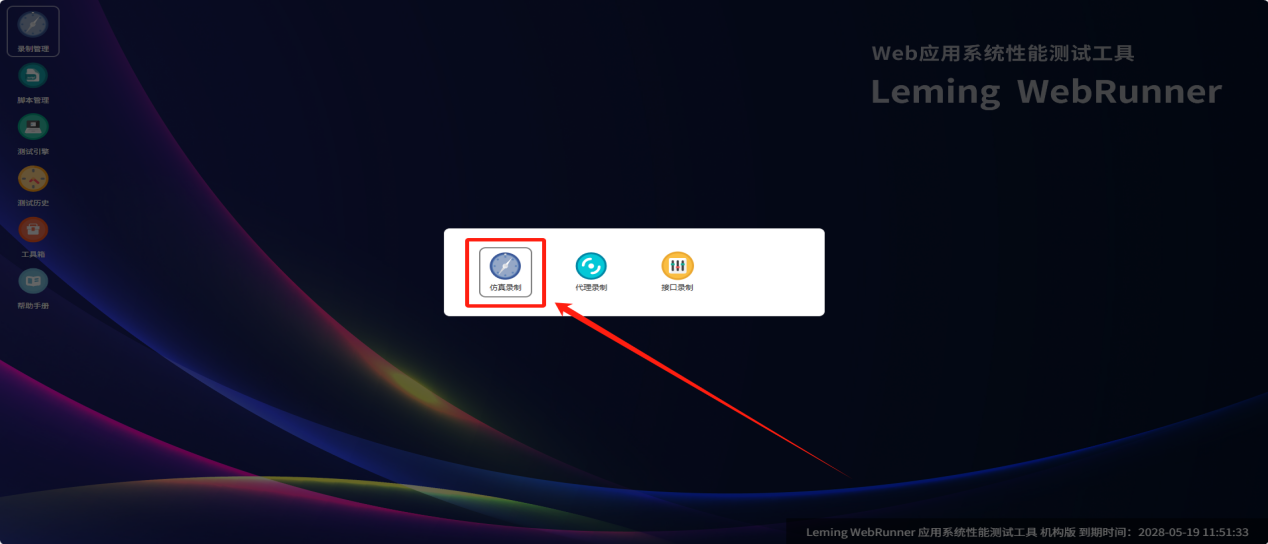
## 4.2脚本录制-仿真录制

启动工具后

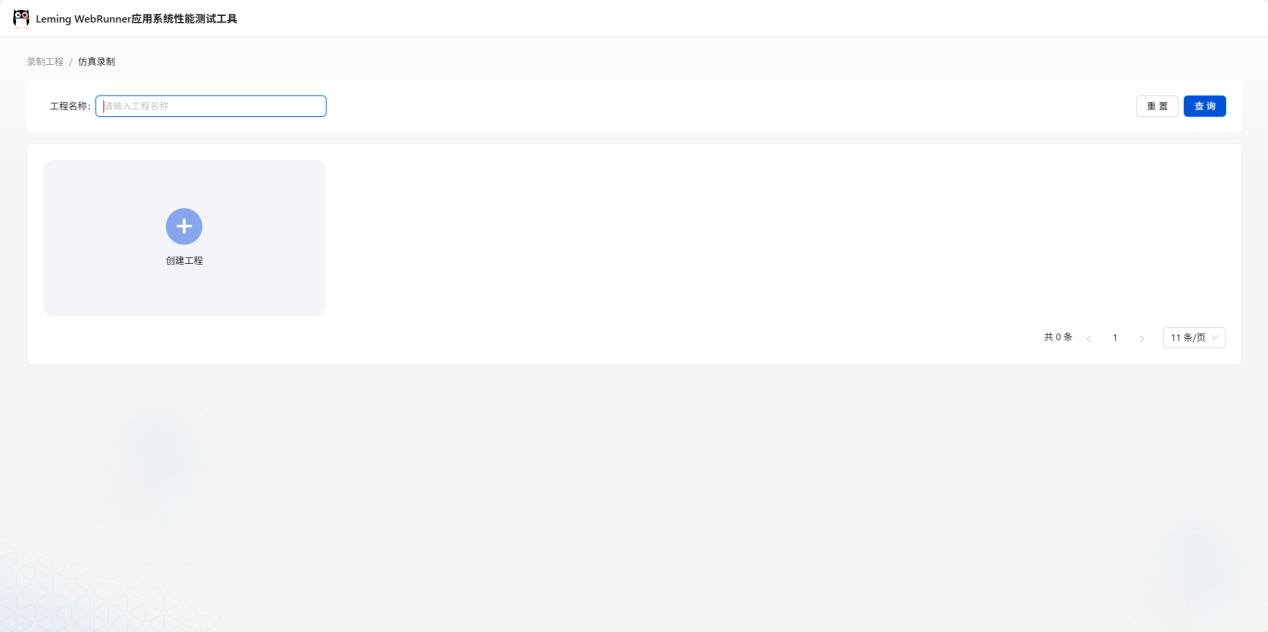
双击工具首页中的“录制管理”



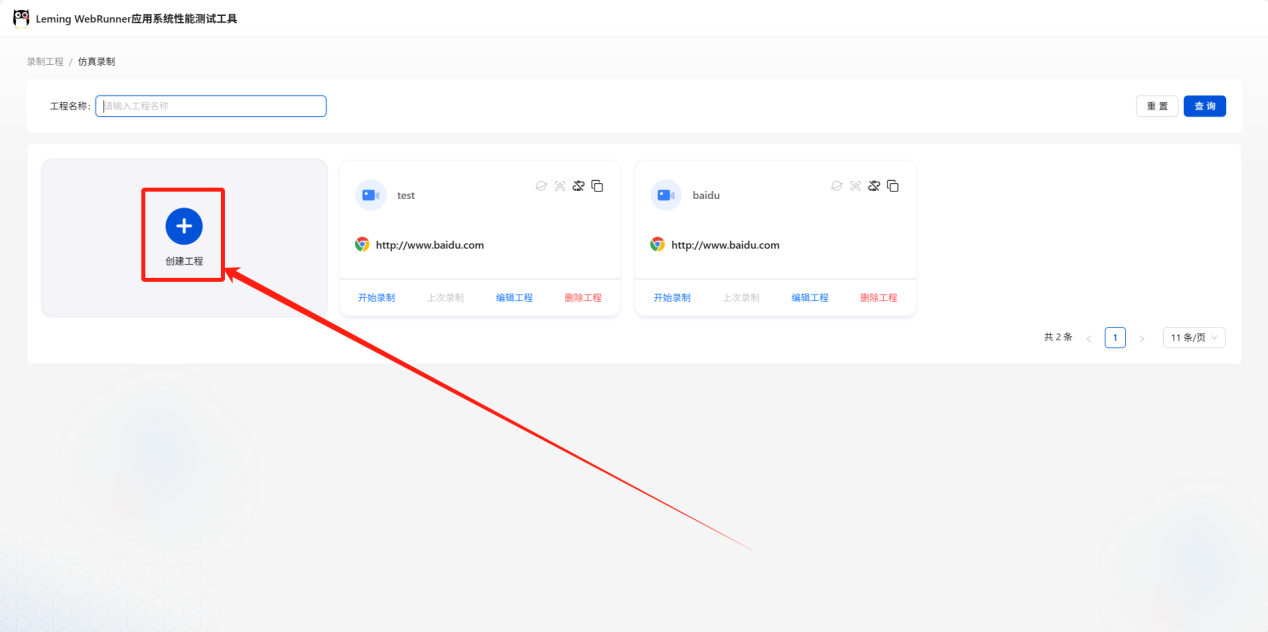
双击“仿真录制”



进入到仿真录制界面



点击“创建工程”



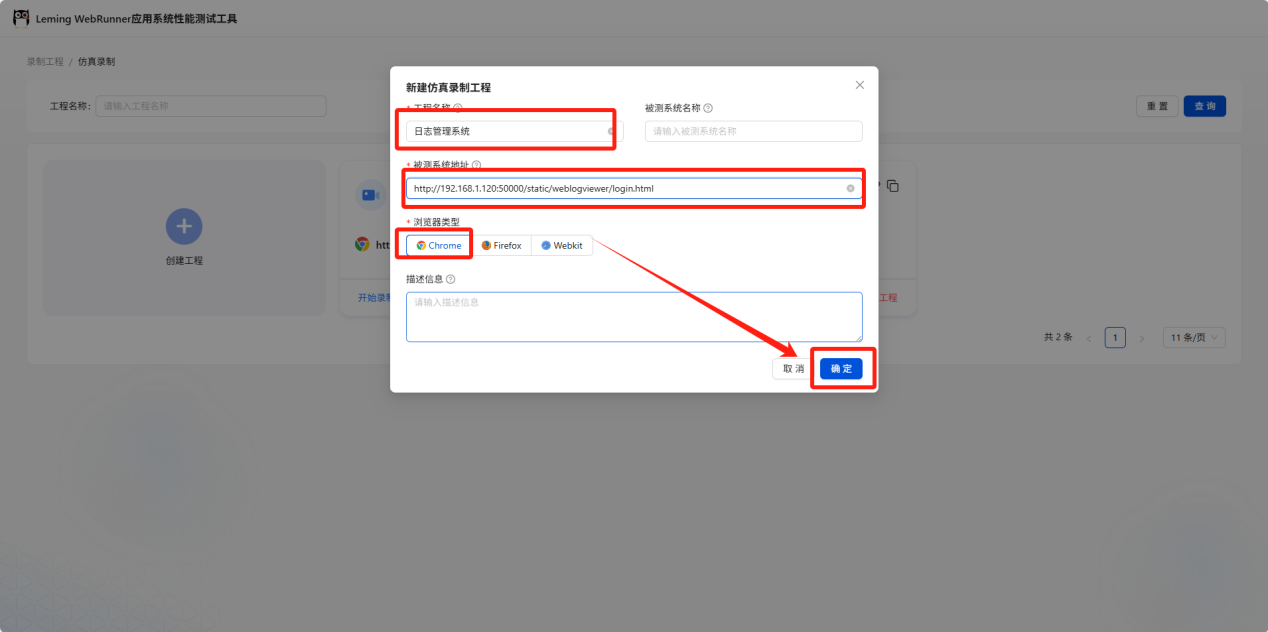
进入到工程创建界面



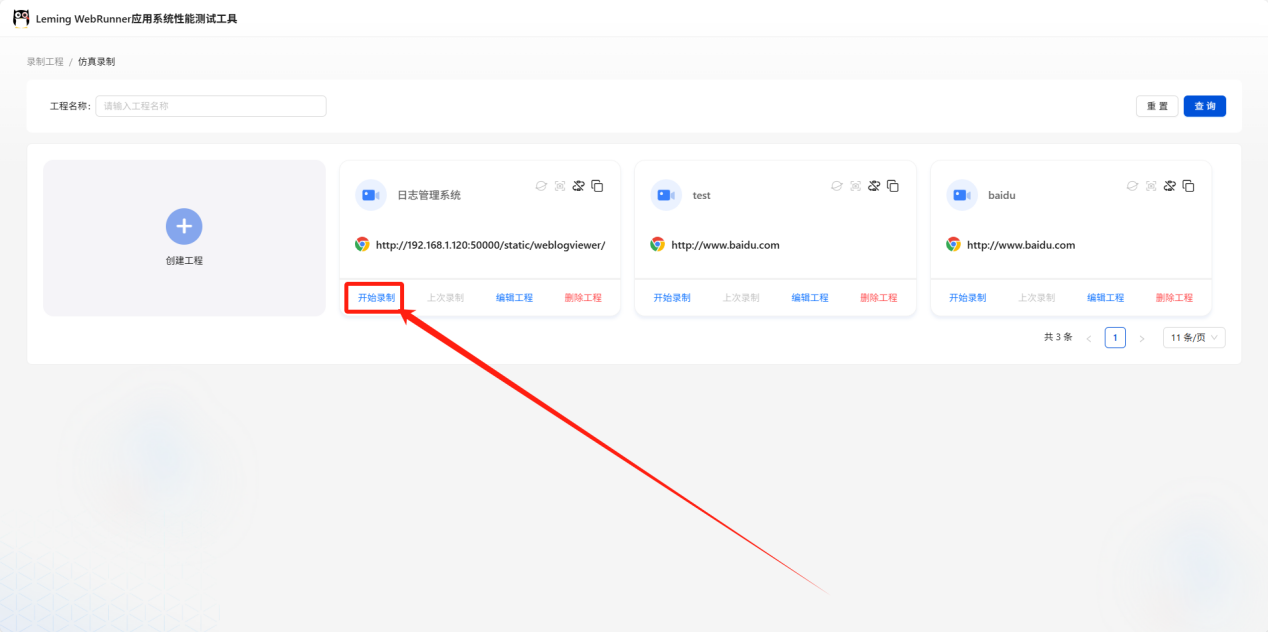
输入工程名称“日志管理系统”（名称可自取）

输入被测系统地址“http://192.168.1.120:50000/static/weblogviewer/login.html”（可更换其他地址）

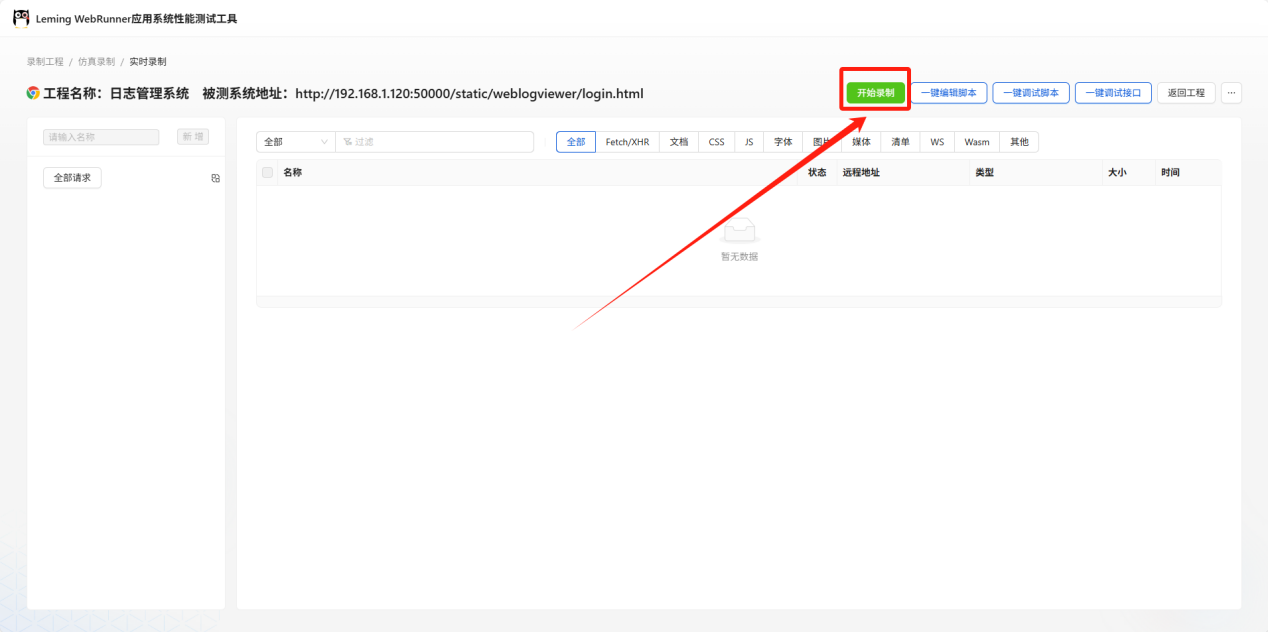
选择“”谷歌浏览器（也可选取其他两个浏览器），点击“确定”按钮



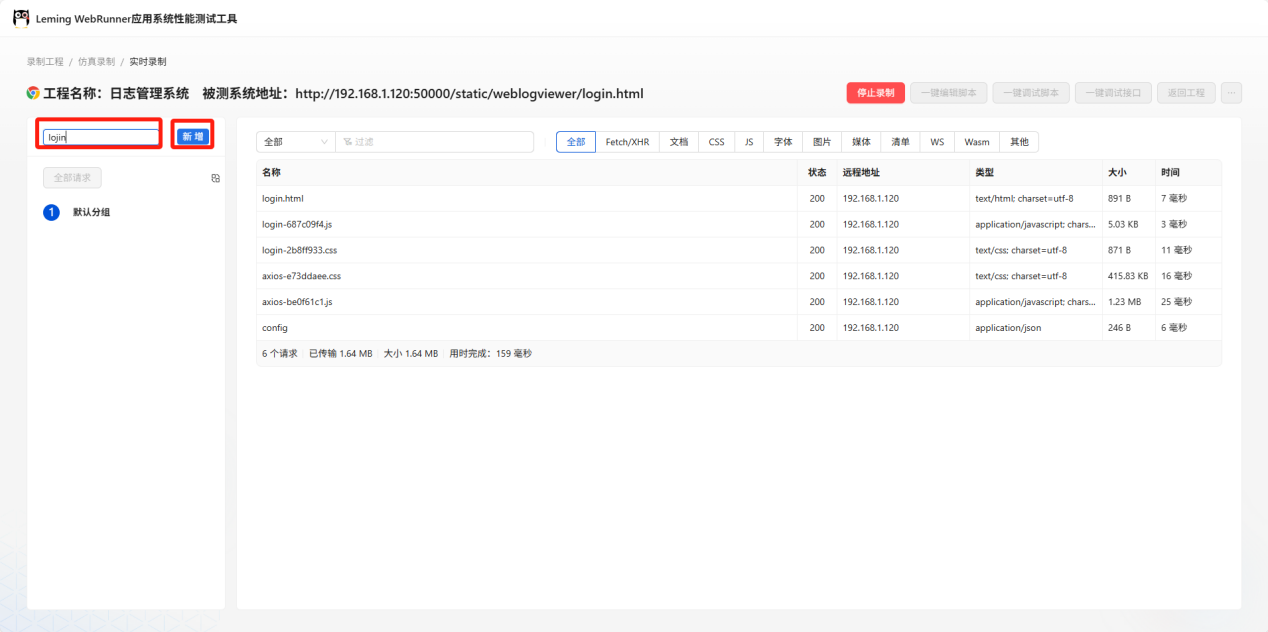
点击“开始录制”按钮



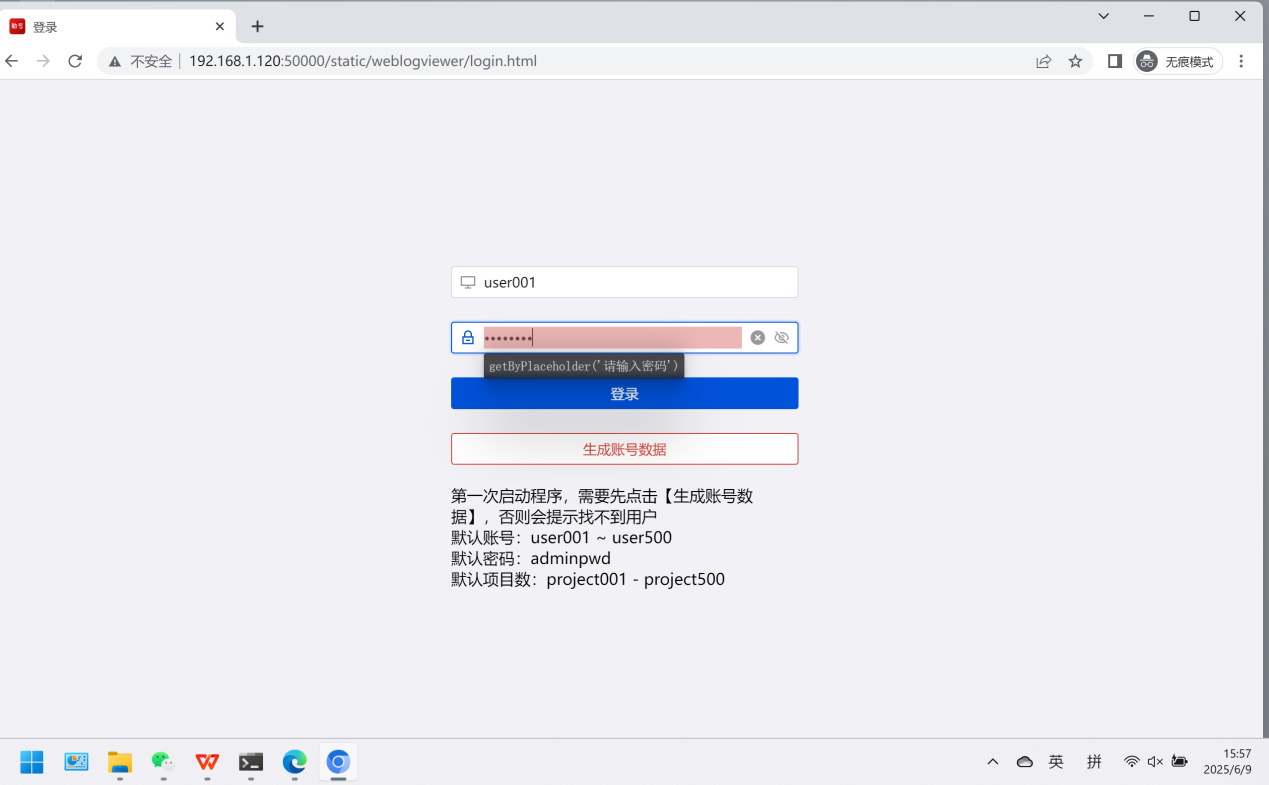
进入到录制界面，点击“开始录制”



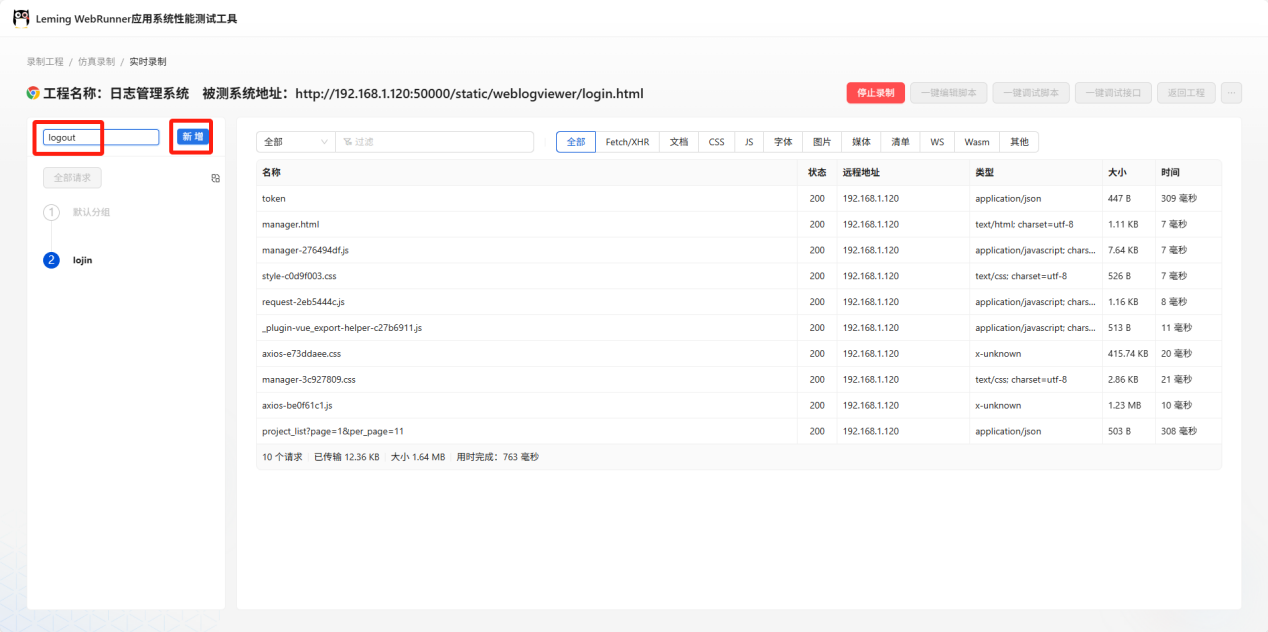
在左侧输入框中输入“login”，点击“新增”按钮（分组操作）



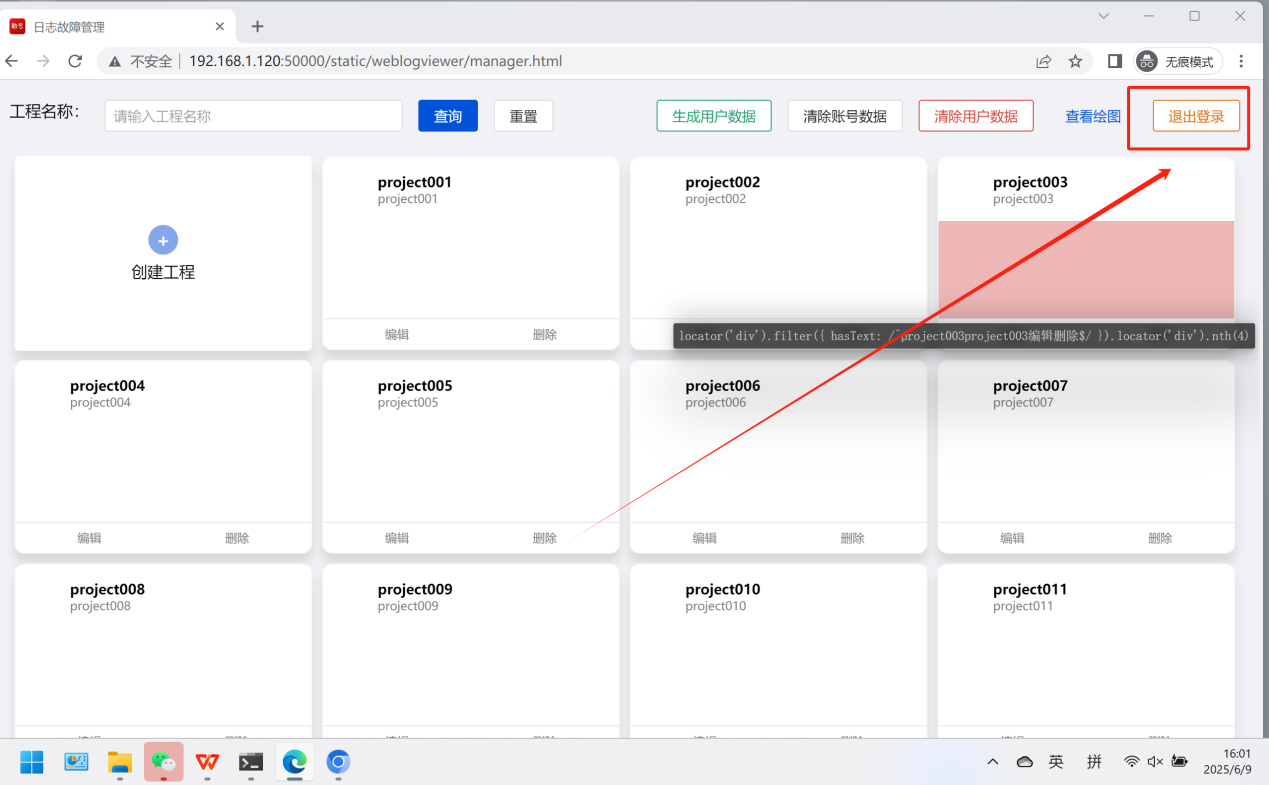
在仿真录制界面输入登录名称“user001”，登录密码“adminpwd”，点击“登录”



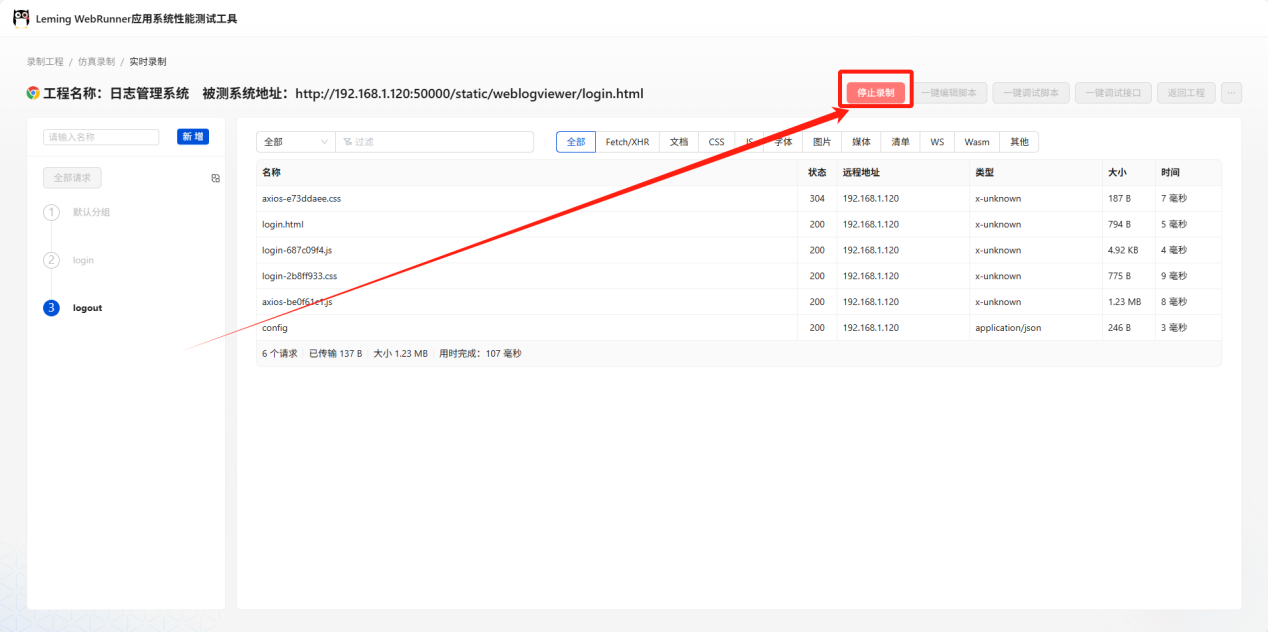
再次在输入框中输入“logout”，点击“新增”按钮（分组操作）



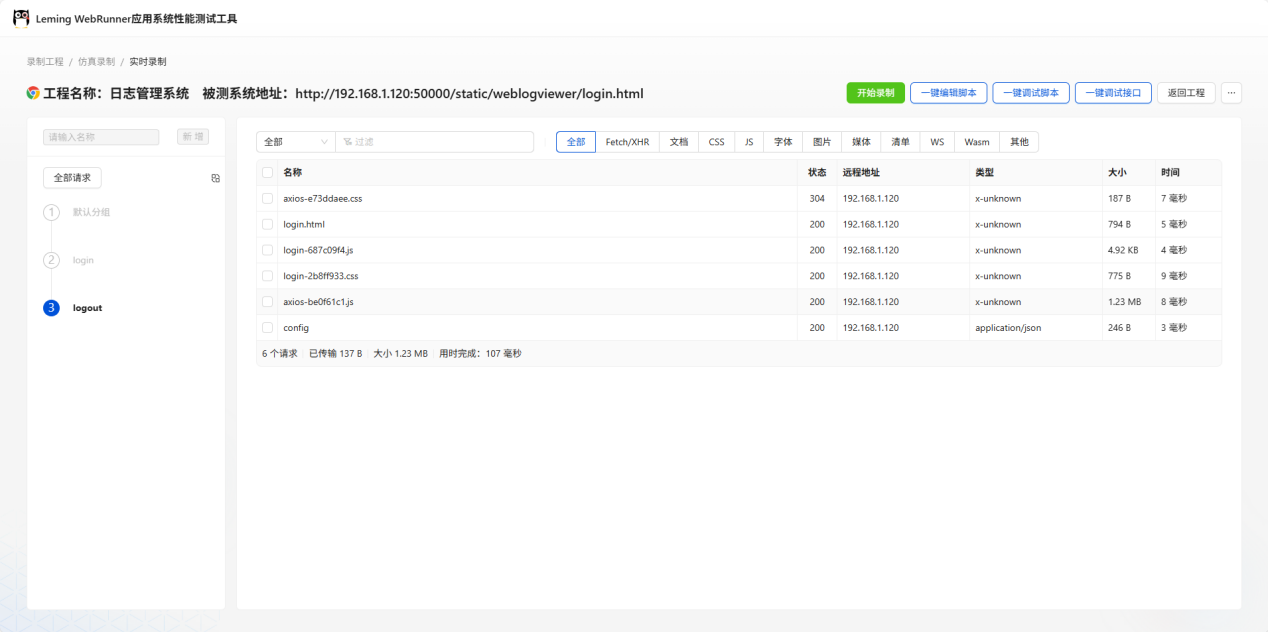
在录制界面中，点击“退出登录”



回到仿真录制实时录制界面，点击“停止录制”

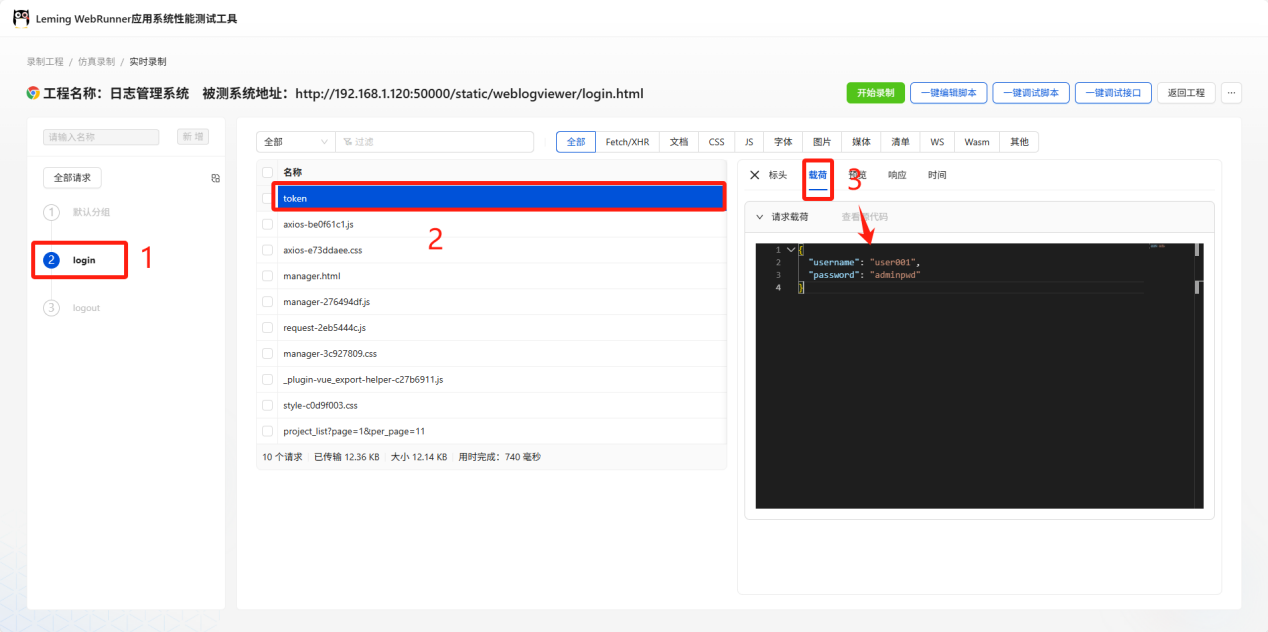


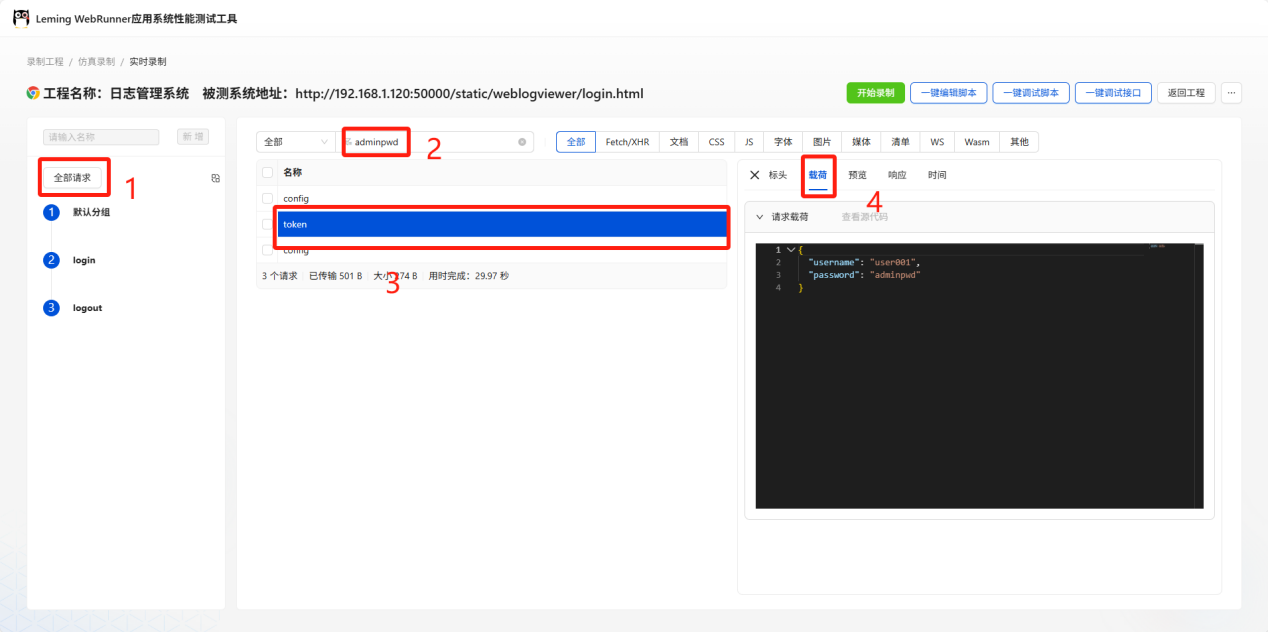
仿真录制停止后的界面

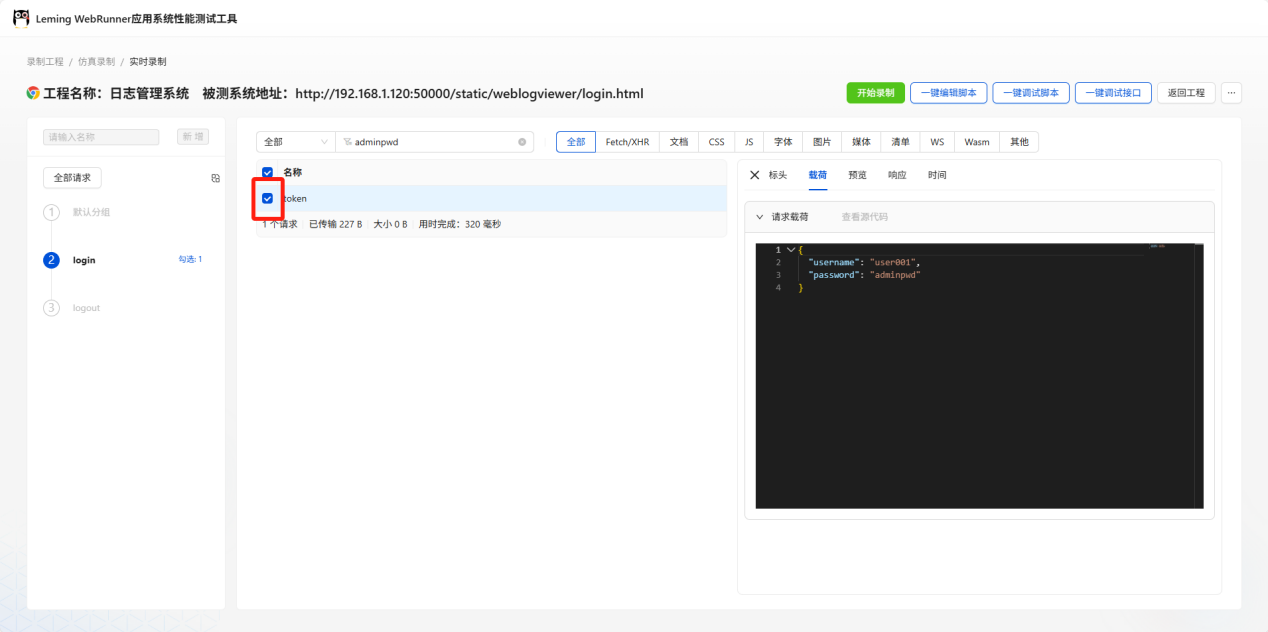


点击左侧分组中的“login”，点击名称为“token”的请求，点击右侧的“载荷”

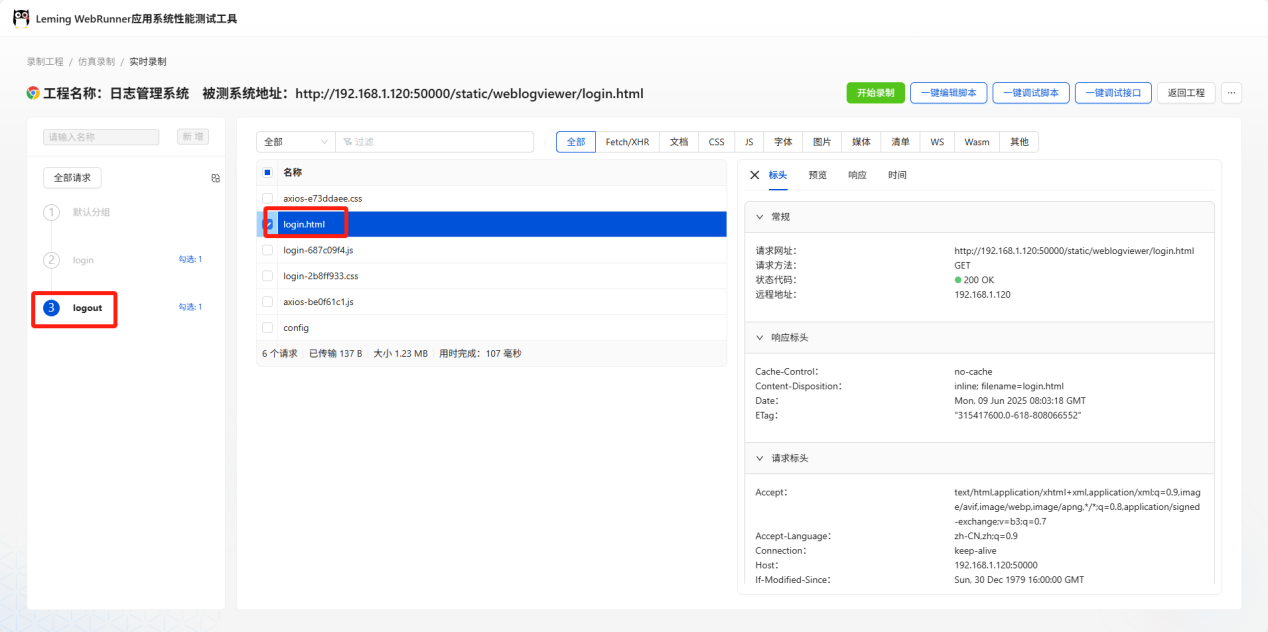
或是点击左侧“全部请求”，在输入框中输入登录密码“adminpwd”，点击名称为“token”的请求，点击右侧的“载荷”（此操作为获取登录请求，排斥其他请求），勾选该条请求





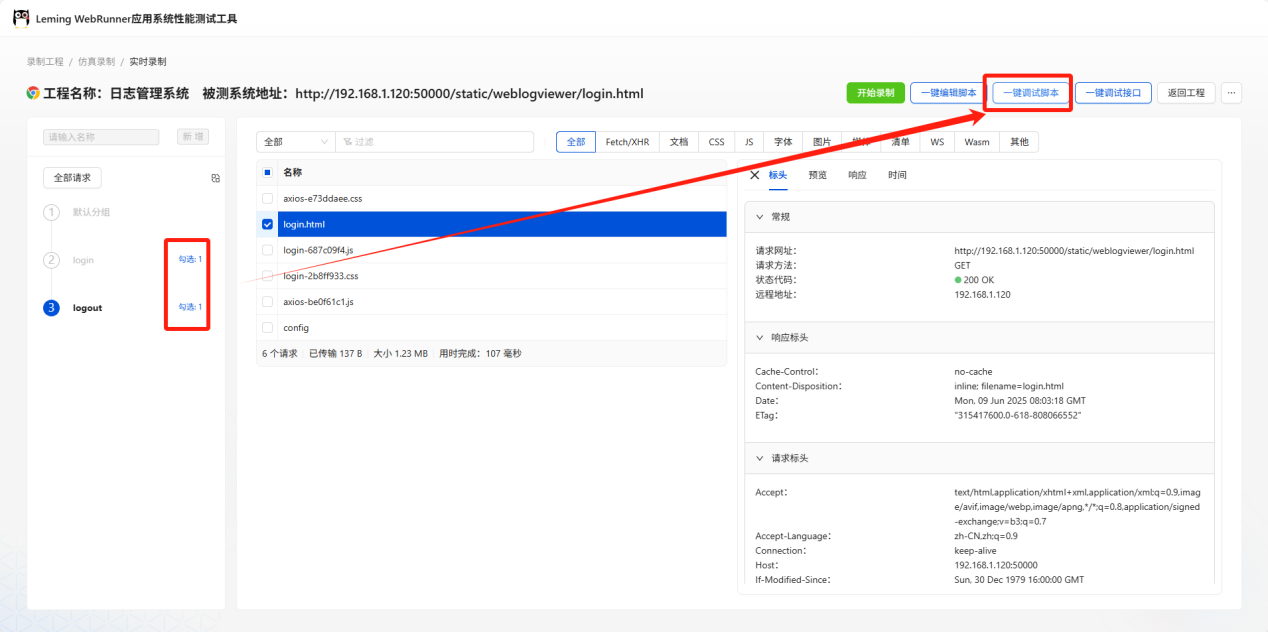


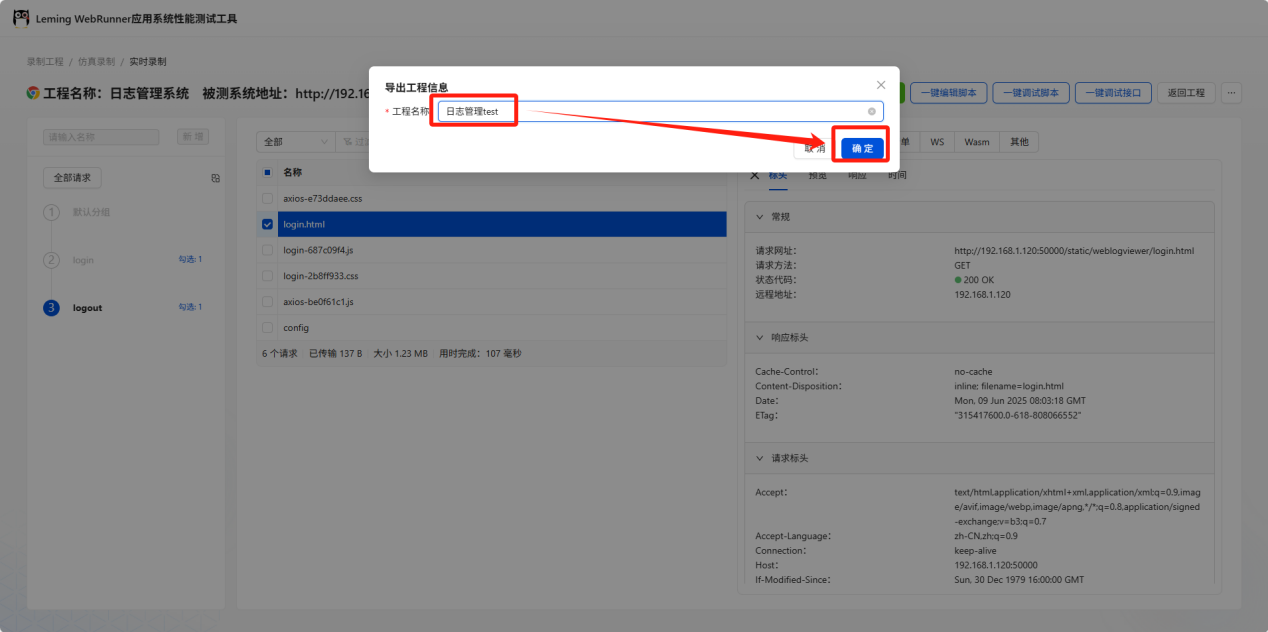
点击左侧分组中的“logout”，点击名称为“login.html”的请求（此操作为获取登录请求，排斥其他请求），勾选该条请求（此请求为退出登录的请求与开始登录的地址一致，其他地址可根据上述获取登录请求操作方式获取请求）



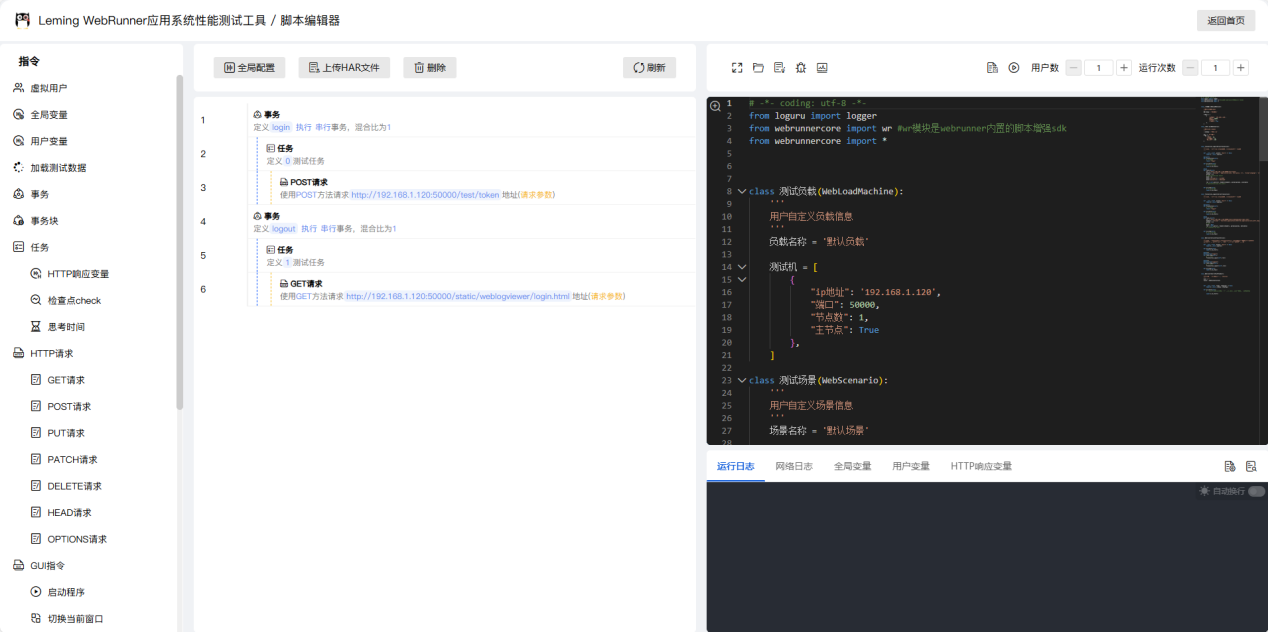
## 4.3脚本编辑

勾选请求后，点击“一键编辑脚本”



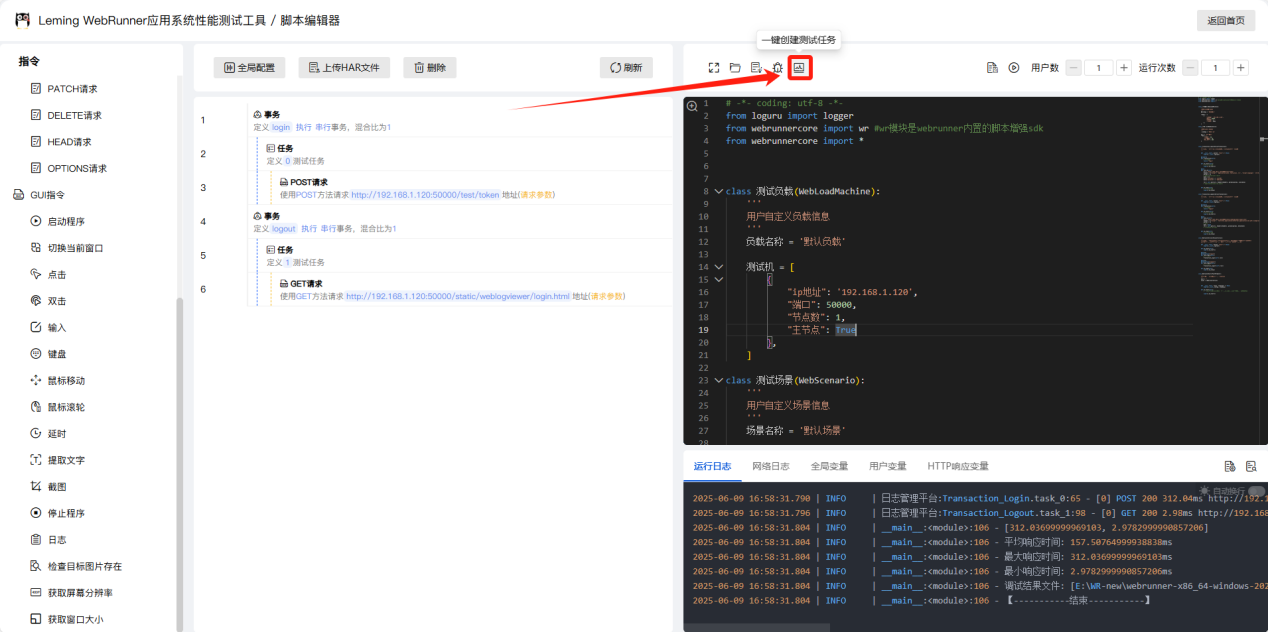


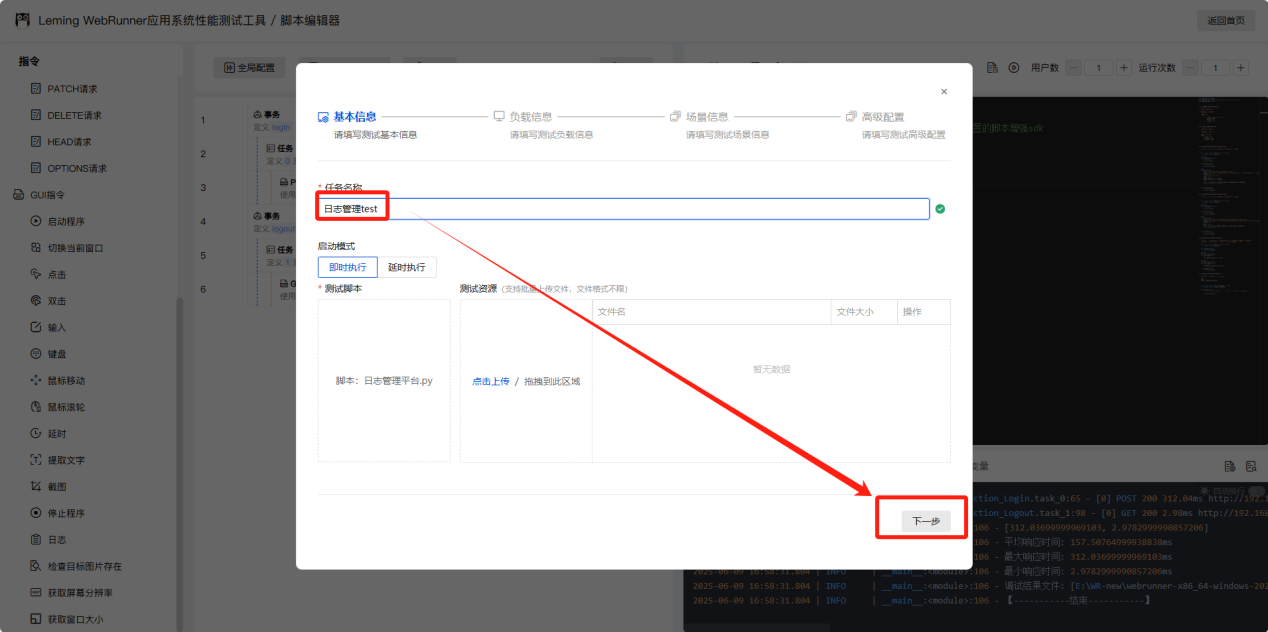
跳转至脚本编辑界面（左侧菜单栏，可根据需求对脚本进行修改）



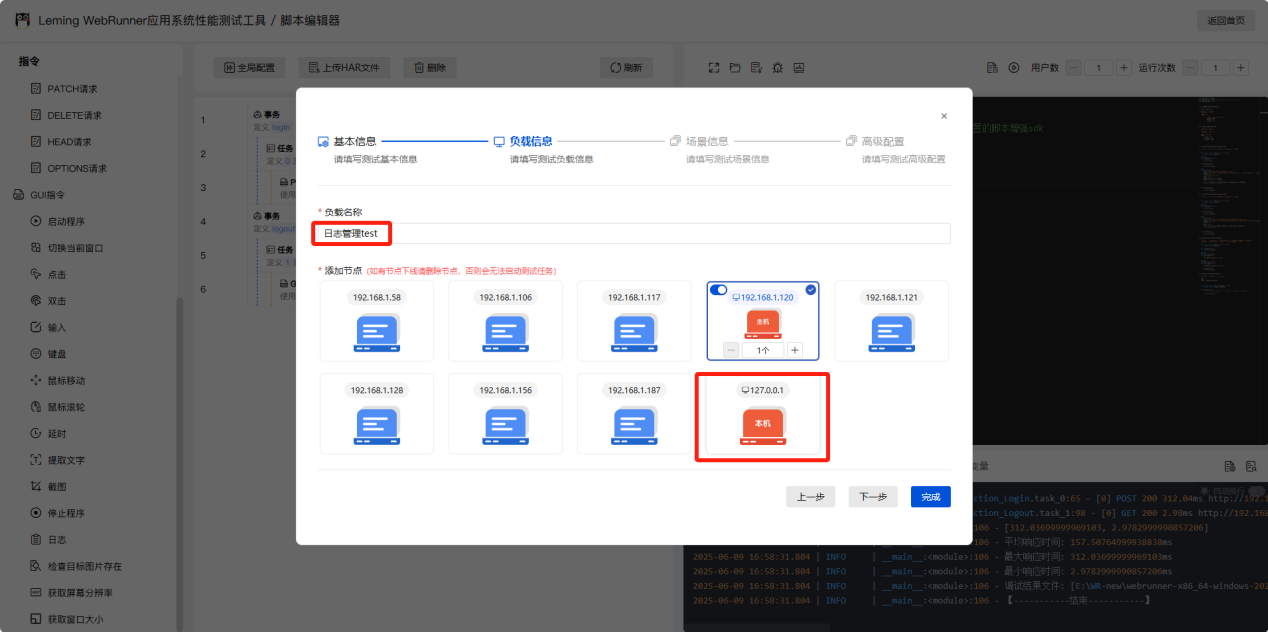
## 4.4测试任务创建

点击“ ”图标，输入任务名称“日志管理test”，点击下一步

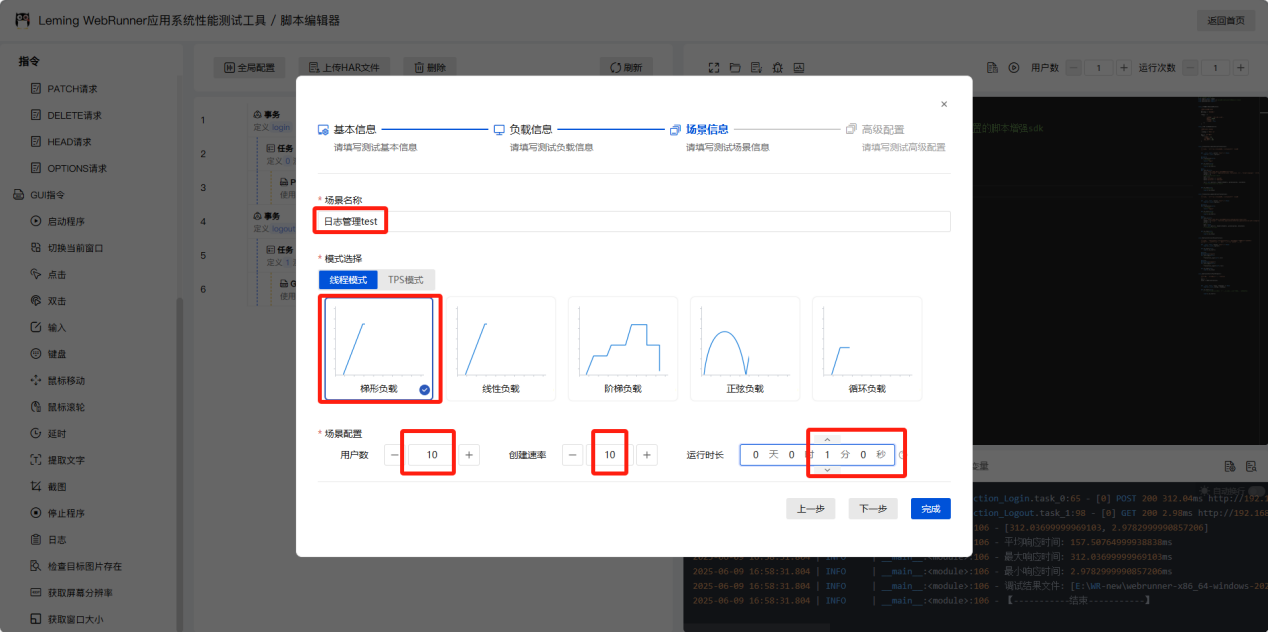




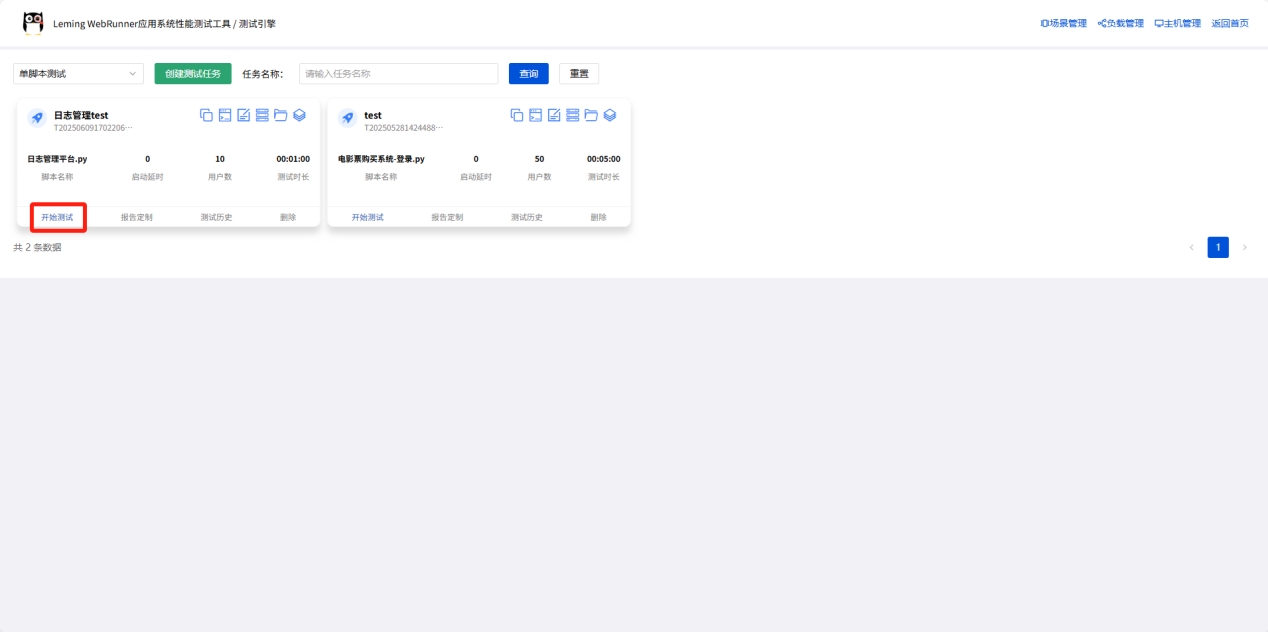
输入负载名称“日志管理test”，添加节点处可默认，也可以选择“本机”，点击下一步



输入场景名称“日志管理test”，选择线程模式下“梯形负载”，输入用户数为“10”，创建速率为“10”，运行时长为“1分”，点击完成按钮

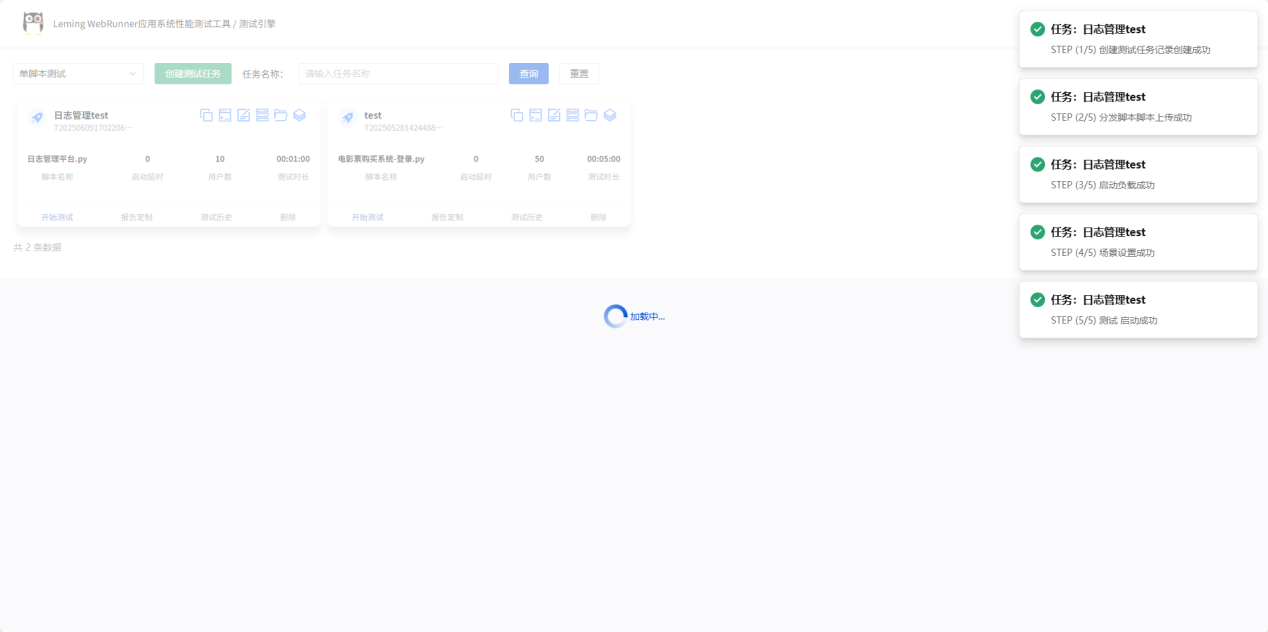


跳转至测试引擎界面，选择所创建的名称为“日志管理test”的任务



## 4.5测试任务启动

点击“开始测试”，测试任务启动界面

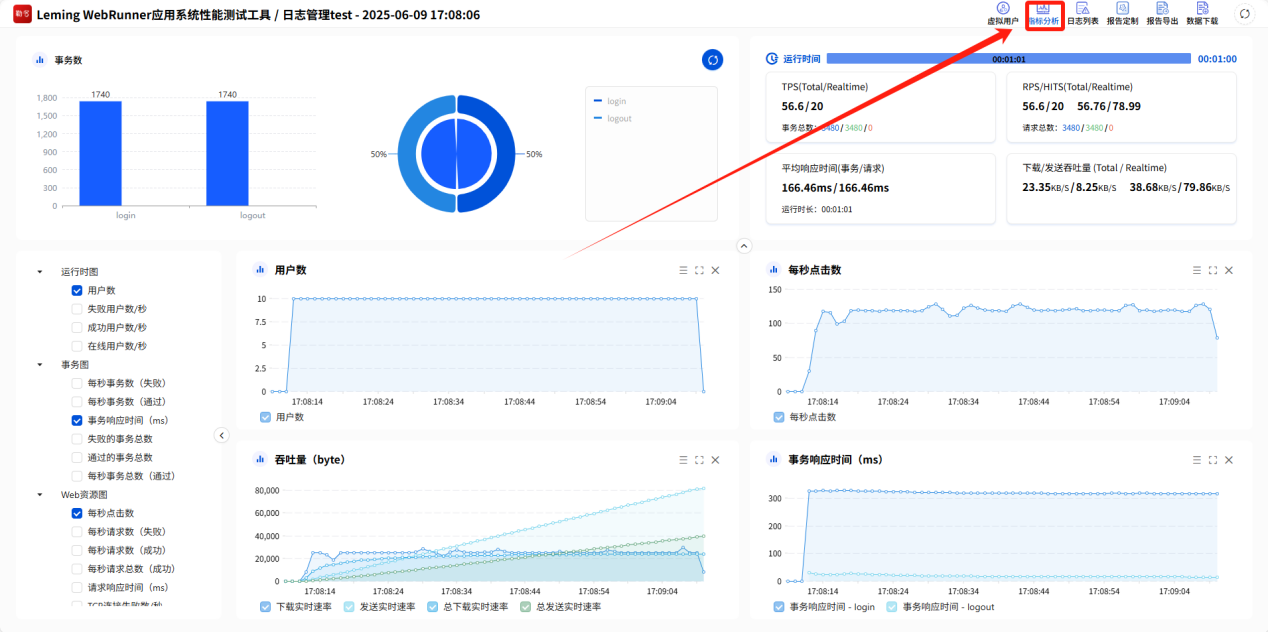


跳转至监控页面

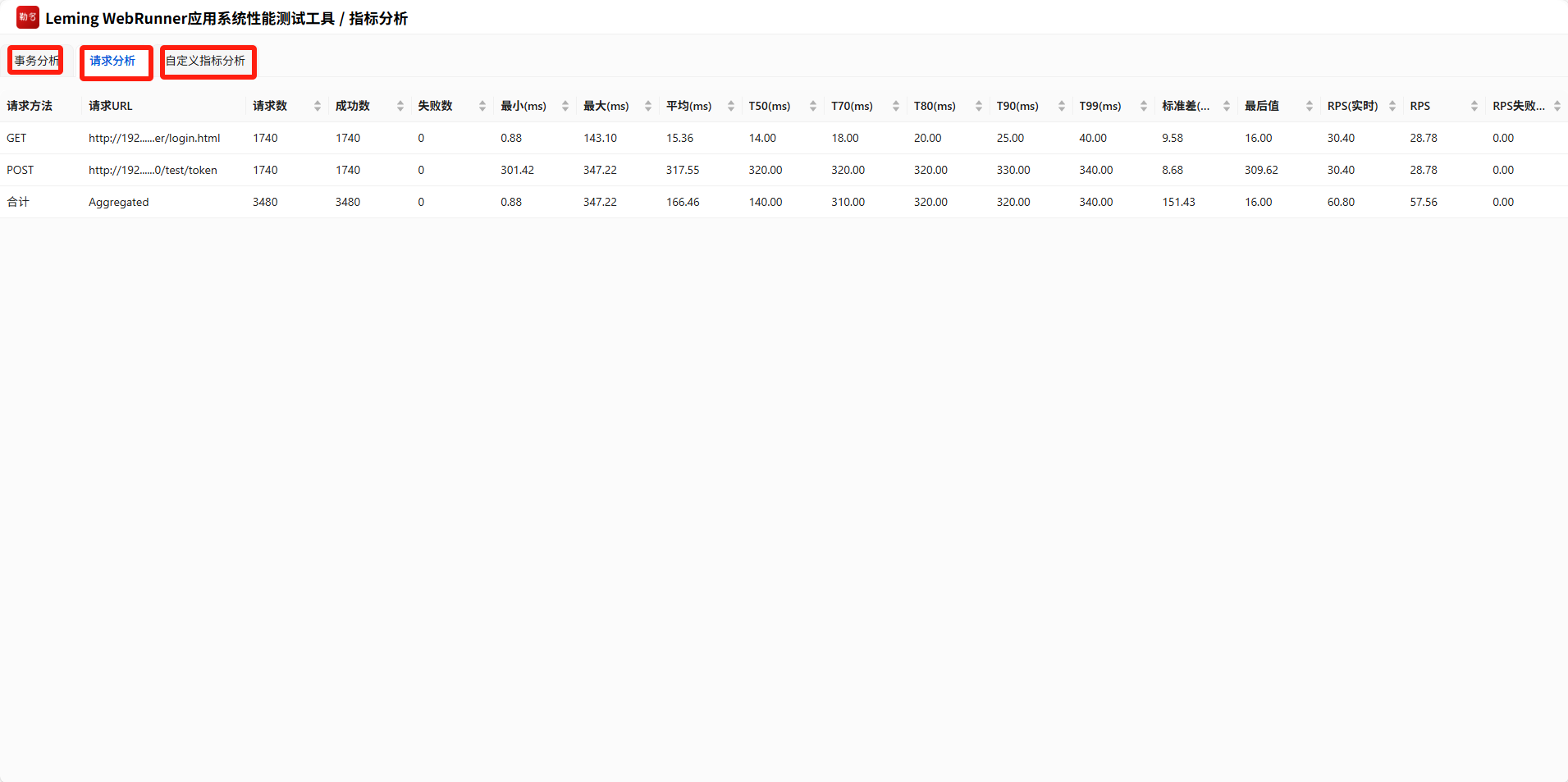


## 4.6指标查看与分析

测试过程中或是测试结束后，点击右上角菜单栏中的“指标分析”，查看测试过程中各接口数据

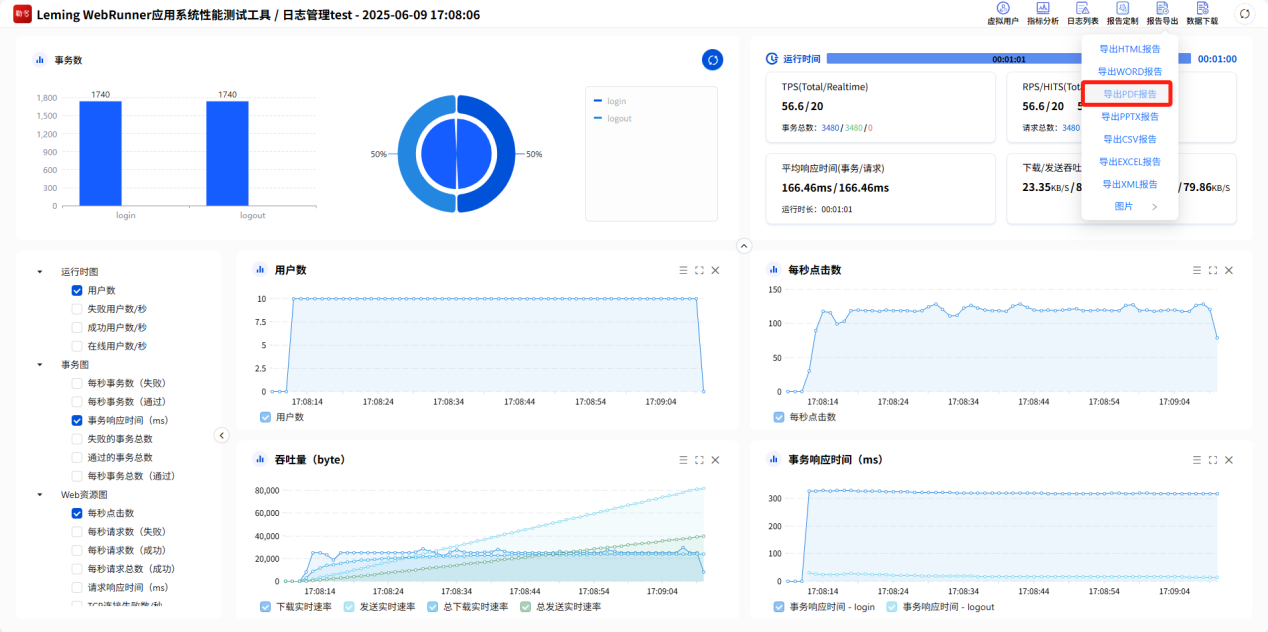


可查看事务分析，请求分析，也可以自定义指标分析



## 4.7测试报告下载

在监控界面，点击右上角菜单栏中的“报告导出”，选择PDF格式（根据需要选择报告格式）



报告导出成功



打开文件，查看报告内容



# ****5.预期结果****

实验预期结果需基于实验目的与步骤，清晰呈现理想状态下应达成的成果。

1. 测试脚本及相关参数化文件
2. 测试报告PDF版本
3. 测试结果参见附件一《实验结果记录表》

# 附件一：《实验结果记录表》

实验结果记录表

|  |  |  |  |
| --- | --- | --- | --- |
| 测试人员 |  | | |
| 测试时间 |  | | |
| 测试结果 | | | |
| 系统登录接口平均响应时间（单位：秒） | | | |
| 并发用户数 | 平均响应时间 | 成功数 | 失败数 |
| 50 | 0.65 | 5000 | 100 |
| 100 | 0.95 | 15000 | 300 |
| 200 | 1.55 | 25000 | 1000 |