



链滴

[思源笔记经验分享] Jupyter 模式

作者: [shuoying](#)

原文链接: <https://ld246.com/article/1654279855533>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

简介

主题 [Dark+](https://ld246.com/forward?goto=https%3A%2F%2Fgithub.com%2FZuoqiuYingyi%2Fsiyuan-theme-dark-plus) 在 `v1.1.0` 大版本更新中来了一个大功能: 在思源中以 Jupyter 的形式执行代码

-

实现方案

- [siyuan-theme-dark-plus/app/jupyter/](https://ld246.com/forward?goto=https%3A%2F%2Fgithub.com%2FZuoqiu-Yingyi%2Fsiyuan-theme-dark-plus%2Ftree%2Fmain%2Fapp%2Fjupyter)

-

功能介绍

-

管理内核与会话

-

使用 RESTful 风格 API 与 Jupyter 服务进行交互

- [The REST API — Jupyter Server documentation](https://ld246.com/forward?goto=https%3A%2F%2Fjupyter-server.readthedocs.io%2Fen%2Flatest%2Fdevelopers%2Frest-api.html)
- [Swagger UI - Jupyter Server API](https://ld246.com/forward?goto=https%3A%2F%2Fpetstore.swagger.io%2F%3Dhttps%3A%2F%2Fraw.githubusercontent.com%2Fjupyter%2Fjupyter_server%2Fmaster%2Fjupyter_server%2Fservices%2Fapi%2Fapi.yaml%23%2Fcontents%2Fget_api_contents__pat_)

-

-
-

发送待执行的代码, 接受内核的返回结果并将其插入到笔记中

-

使用 WebSocket 与 Jupyter 内核进行交互

- [WebSocket kernel wire protocols](https://ld246.com/forward?goto=https%3A%2F%2Fjupyter-server.readthedocs.io%2Fen%2Flatest%2Fdevelopers%2Fwebsocket-protocols.html%23)

-

-

-

如果喜欢该功能, 欢迎为 [Dark+](https://ld246.com/forward?goto=https%3A%2F%2Fgithub.com%2FZuoqiu-Yingyi%2Fsiyuan-theme-dark-plus) 点一个 🌟 (*^▽^*)

使用教程

1--添加-Jupyter-服务对思源的信任

1. 添加 Jupyter 服务对思源的信任

<p>打开配置文件 <code>jupyter_lab_config.py</code> 或 <code>jupyter_notebook_config.p</code> </p>

使用命令 <code>jupyter lab --generate-config</code> 或 <code>jupyter notebook --ge</code> <code>erate-config</code> 生成配置文件
Windows 系统生成的配置文件一般在 <code>C:\Users\<用户名>\jupyter</code> 目

<p>找到 <code>c.ServerApp.allow_origin</code> 或 <code>c.NotebookApp.allow_origin</</code> 字段</p>

该字段用于设置被信赖的请求源
也可以通过 <code>c.ServerApp.allow_origin_pat</code> 或 <code>c.NotebookApp.allow</code> <code>origin_pat</code> 字段使用正则表达式进行设置可信源

<p>将该字段设置为思源 <code>'location.origin'</code> 或 <code>'*</code> </p>

<code><code>Ctrl + Shift + I</code></code> 打开思源的开发者工具, 在控制台中输入 <code>location.origi</code> </code> 后回车即可获得思源的源地址, 桌面端一般为 <code>http://127.0.0.1:6806</code>
设置为 <code>'*</code> 将信任所有请求源
示例: <code>c.ServerApp.allow_origin = 'http://127.0.0.1:6806'</code>

<p>保存文件并启动 jupyter 服务</p>

<h3 id="2--登录-Jupyter-服务">2. 登录 Jupyter 服务</h3>
<p><code><code>https://ld246.com/images/img-loading.svg" alt="思源笔记功能示范 28.gif" dat</code></code>-src="https://b3logfile.com/file/2022/06/%E6%80%9D%E6%BA%90%E7%AC%94%E8%AE%B</code></code>-</code></code>%E5%8A%9F%E8%83%BD%E7%A4%BA%E8%8C%83-28-38ef47d3.gif"> </p>

<p><code><code>文档块菜单</code></code> > <code><code>Jupyter</code></code> > <code><code>全局设置</code></code> </p>

<p>打开全局设置窗口</p>

<p>输入服务器 URL 并点击 <code><code>确定</code></code> </p>

<code><code>http(s)://hostname(:port)</code></code>

<p>(可选, Cookie 认证方案) 点击 <code><code>测试</code></code> 链接跳转到登录页面并登录</p>

(可选, Cookie 认证方案) `Ctrl + Shift + I` 打开开发者工具, 获得 `Cookie`

- 方案 1: 在控制台输入 `document.cookie` 并从输出结果复制 `_xsrf` 字段
- 方案 2: 选择一个网络链接并从 HTTP 请求头的 `Cookies` 字段复制 `_xsrf` 字段
- `_xsrf` 字段完整格式为 `_xsrf=d|xxxxxxxx|xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx|ddddd`
- `d`: 一个十进制数字
- `x`: 一个十六进制数字

关闭 Jupyter 窗口并重新打开全局设置窗口

输入 `Cookies` 或 `Token` 并点击 `确定`

`Cookies` 认证方案仅适用于 Jupyter 服务与思源服务 URL host 相同的情况

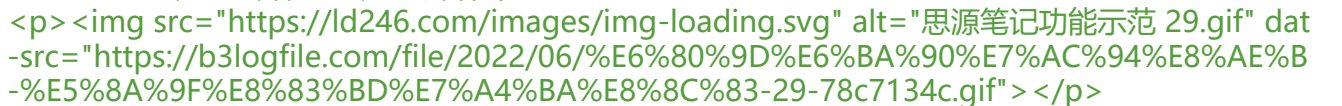
示例:

- 思源服务 URL: `http://127.0.0.1:6806`
- Jupyter 服务 URL: `http://127.0.0.1:8888`

关闭全局设置窗口

单击 `重新加载窗口` 按钮或 `Ctrl + F5` 刷新页面

3. 建立会话



- 单击 `文档块菜单` > `Jupyter` > `文档设置` 单项打开文档设置窗口
- 选择想要启动的内核, 输入 `会话名称` 与 `会话目录`, 单击 `新建` 按钮启动该内核并建立会话
- `会话名称` 是在 Jupyter 中查看活动内核时显示的名称

-

<p> <code>会话目录</code> 所指定的目录是使用相对路径读写文件时的基点</p>

- 示例: 假设 Jupyter 服务设置的根目录为 <code>D:\foo</code>, 这里会话目录设置为 <code>bar\测试</code>, 那么使用该内核访问相对路径 <code>./</code> 实际访问的是 <code>D:\foo\ar</code> 目录

-

<p>单击<kbd>中断</kbd>或<kbd>重启</kbd>按钮, 若没有弹窗则表示成功与服务器建立连接<p>

- <kbd>中断</kbd>按钮会中断正在运行(忙碌)的内核所执行的任务
- <kbd>重启</kbd>按钮会重启内核, 内核之前运行时产生的所有变量与数据均会丢失

-

<p>单击<kbd>连接</kbd>按钮</p>

- 该按钮会将会话与内核信息写入文档块的块属性中
- 在该文档中想要执行代码时可以使用这些信息建立与内核的连接

-

<p><kbd>F5</kbd> 刷新文档, 若文档右上角内核状态显示 <code>启动中</code> 或 <code>闲</code> 则表示连接内核成功</p>

<p></p> - <p>单击<kbd>代码块菜单</kbd> > <kbd>Jupyter</kbd> > <kbd>运行代码</kbd> 菜 <p>项</p> - 单击该项后会以文档关联的内核执行代码块中的代码 - 之后将代码的执行结果插入到该代码下的超级块中(如果没有则新建) - 输出结果中的内容会以 Markdown 源码的形式进行解析 - <p><kbd>代码块菜单</kbd> > <kbd>Jupyter</kbd> > <kbd>运行代码 (转义输出结果)</p> <p>项</p> - 输出结果中的所有半角符号都使用 <code>\</code> 进行转义, 从而避免其在插入时解析为 Ma <p>kdown</p>

>5. 关闭连接</h3>

<p>单击<kbd>文档块菜单</kbd> > <kbd>Jupyter</kbd> > <kbd>关闭连接</kbd> 菜单项</p>

<p>该操作不会关闭内核与会话, 而是关闭与它的连接</p>

不会清空内核执行代码时产生的变量与数据

下一次运行代码时会重新建立一个 WebSocket 连接

<p>该操作会重置运行序号</p>

>6. 关闭内核</h3>

<p></p>

单击<kbd>文档块菜单</kbd> > <kbd>Jupyter</kbd> > <kbd>文档设置</kbd> 打开文档设置窗口

单击 <kbd>关闭</kbd> 按钮以关闭会话与内核

<kbd>F5</kbd> 刷新文档, 若文档右上角内核状态显示 <code>No Kernel</code> 则表示核关闭成功

>注意事项</h2>

<p>目前尚未对 <code>input()</code> 输入函数进行适配</p>

<p>目前尚未对除 <code>文本</code>、<code>图片</code> 与 <code>错误输出</code> 外的返回消息进行适配, 如果需要解析其他消息, 请到 Issues 反馈</p>

反馈时请附代码与样式截图

暂未有对存在鼠标交互操作的 <code>ipywidgets</code> 支持计划

<p>目前尚不支持对 <code>*.ipynb</code> 文件的导入与导出</p>

<p>部分魔术指令可能不支持</p>

>使用</h2>

若不使用 Dark+ 主题, 可以下载该主题 <code>v1.1.0+</code> 版本, 将 Dark+ 主题目录下的 <code>theme.js</code> 文件 复制 到所使用的其他主题的目录下
