###### 第一步：创建 jupyterlab 插件模板

# 创建 conda 虚拟环境

**conda create -n test --override-channels --strict-channel-priority -c conda-forge -c nodefaults jupyterlab=3 cookiecutter nodejs jupyter-packaging git**

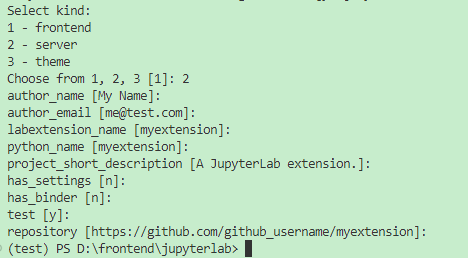
# 激活 test 环境

**conda activate test**

# 构建插件模板

**cookiecutter** [**https://github.com/jupyterlab/extension-cookiecutter-ts**](https://github.com/jupyterlab/extension-cookiecutter-ts)

选择插件类型，为了方便测试请求，这里选择 2 ，带有后端代码示例。



###### 第二步：进入 myextension 插件目录，安装依赖包和安装插件

# 激活 test 环境

**conda activate test**

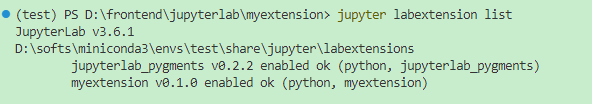
# 安装依赖包和示例插件

**pip install -ve .**

# 先验证前端插件是否安装成功

**jupyter labextension list**

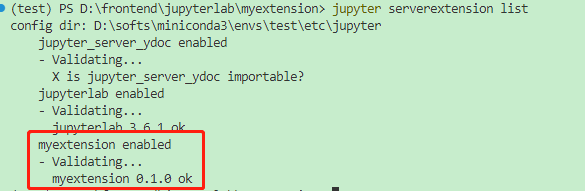
如果成功了会有下面的提示：



# 再验证后端服务插件是否安装成功

**jupyter serverextension list**

如果成功了会有下面的提示：



###### 第三步：启动 jupyter 服务验证插件

# 启动 jupyter 服务

**Jupyter lab**

在打开的网页编辑器中看到如下日志，打印了后端的响应数据，说明插件模板没有问题。

**第四步：修改前端插件请求基路径，请求本地另外的服务（先不提供该服务）**

修改 src/handler.ts 代码：



# 建立一个软链接，以便修改后不需要重新安装插件

**jupyter labextension develop --overwrite .**

# 重新构建插件

**Jlpm build**

刷新页面查看网页控制台日志：

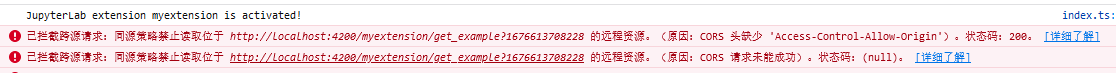


出现请求跨域问题，解决方案参见：

<https://developer.mozilla.org/zh-CN/docs/Web/HTTP/CORS>

**第五步：提供本地另外的服务，尝试解决跨域问题**

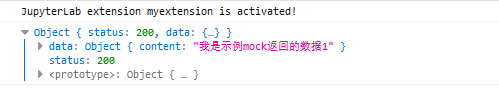
提供服务后，刷新页面：



本地服务响应头添加（与jupyterlab前端插件预检请求头保持对应关系）：



重启第三方服务，刷新jpyterlab页面：



表示跨域请求成功。