Gerapy 分布式爬虫管理框架

简介

基于Scrapy，Scrapyd，Scrapyd-Client，Scrapyd-API，Django和Vue.js的国人开发分布式爬虫管理框架。
Gerapy 可以帮助我们：

1. 方便地控制爬虫运行
2. 直观地查看爬虫状态
3. 实时地查看爬取结果
4. 简单地实现项目部署
5. 统一地实现主机管理
6. 轻松地编写爬虫代码(笔者感觉比较鸡肋)

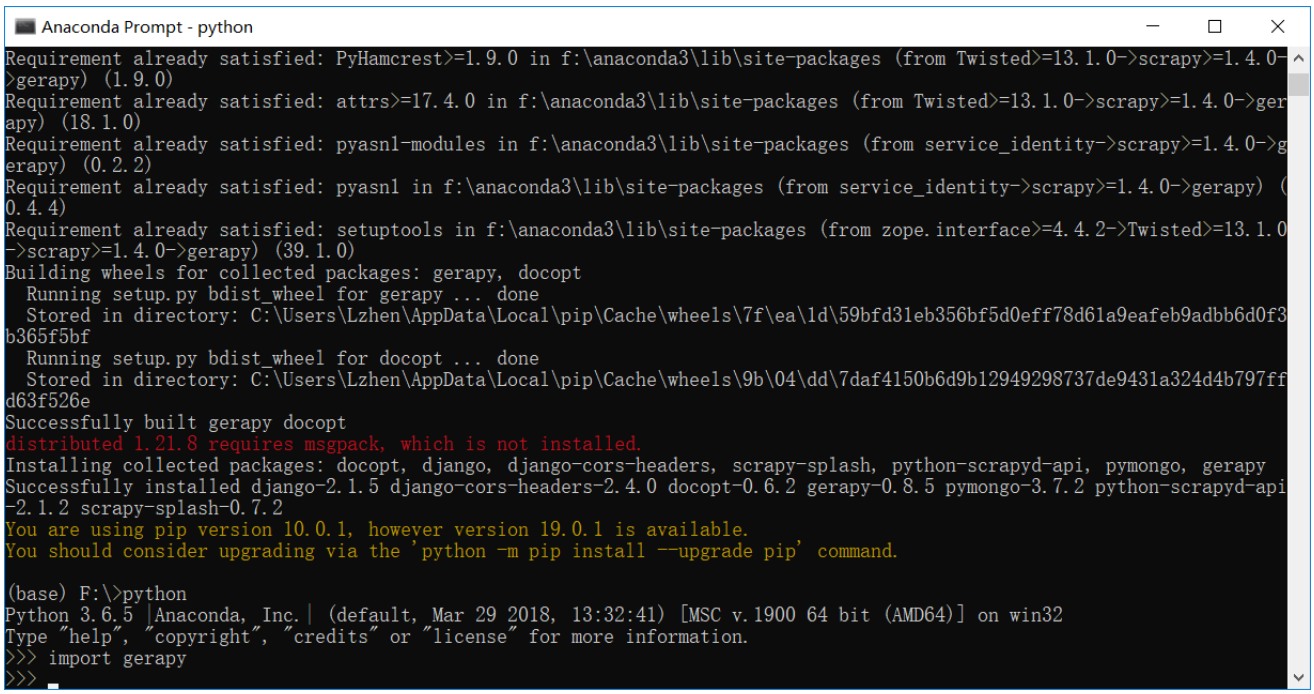
   Gerapy是在Python 3上开发的。可能会支持Python 2

安装

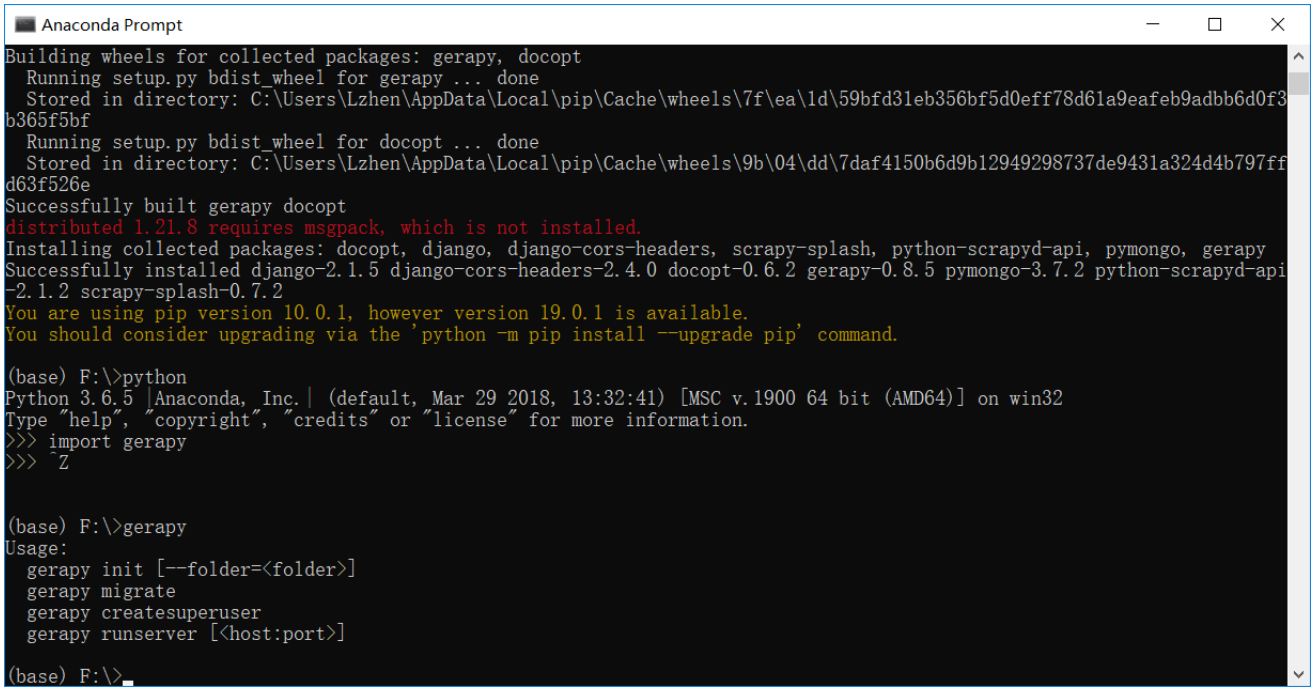
使用pip安装，命令如下：

```
pip install gerapy
```

pip 安装完成后，用Python shell测试下：

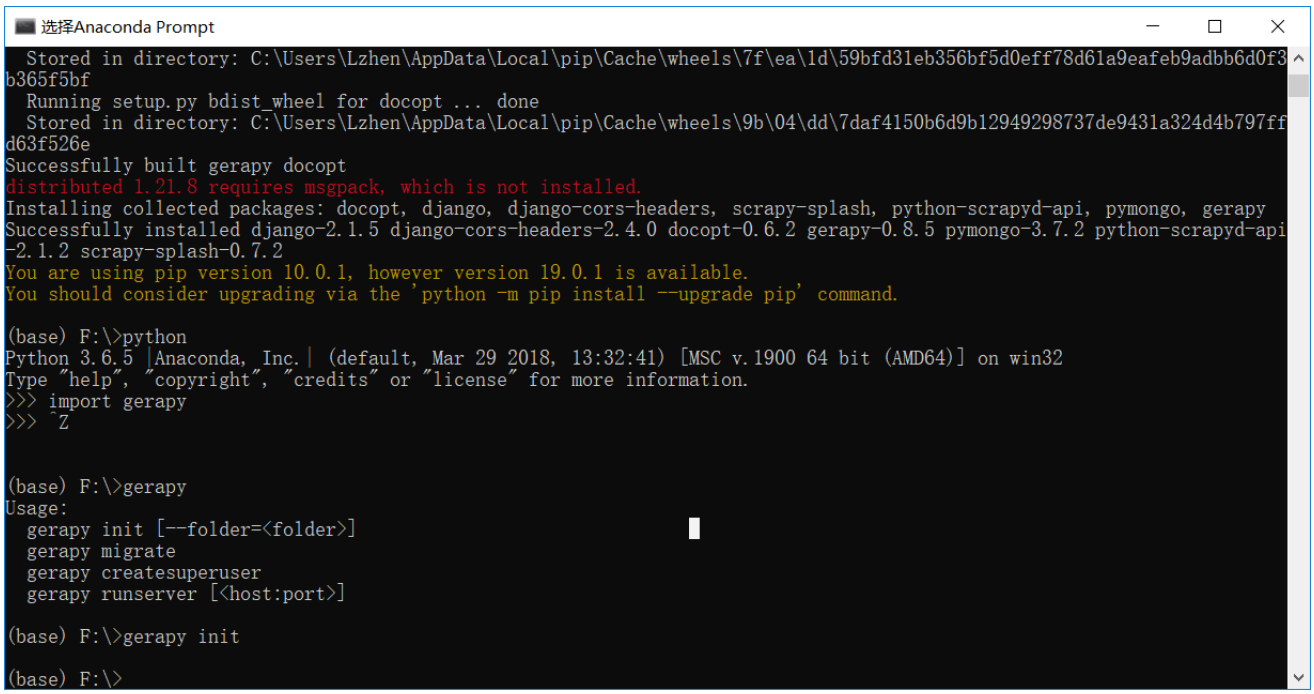


安装完成之后我们就可以使用 gerapy 命令了，输入 gerapy 便可以获取它的基本使用方法：

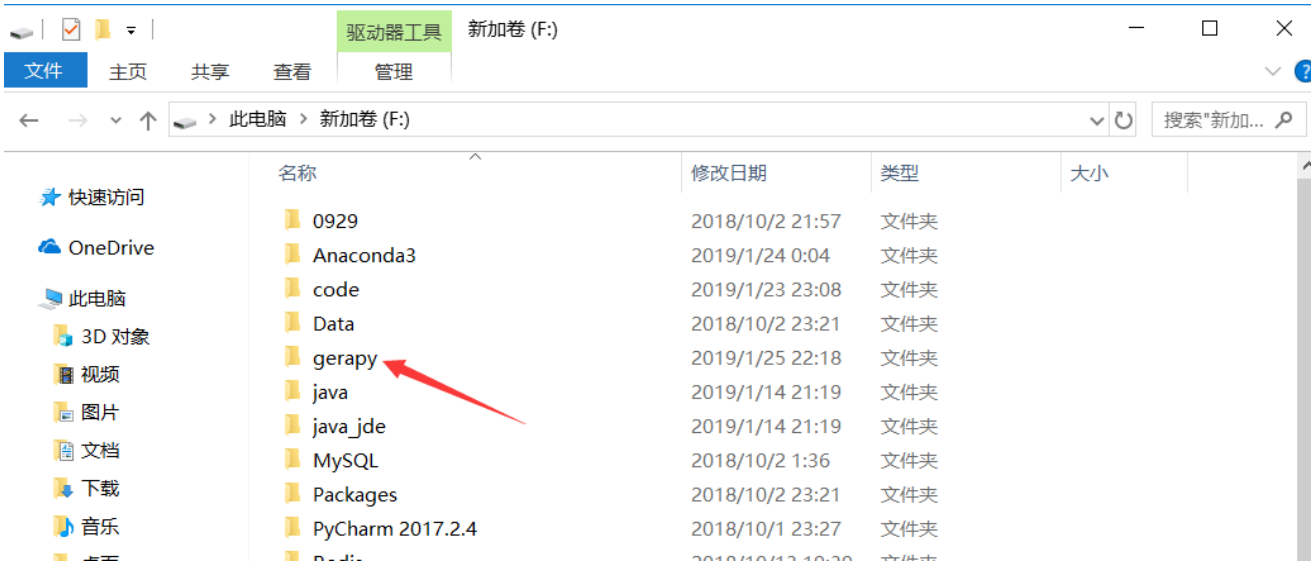


Gerapy 初始化

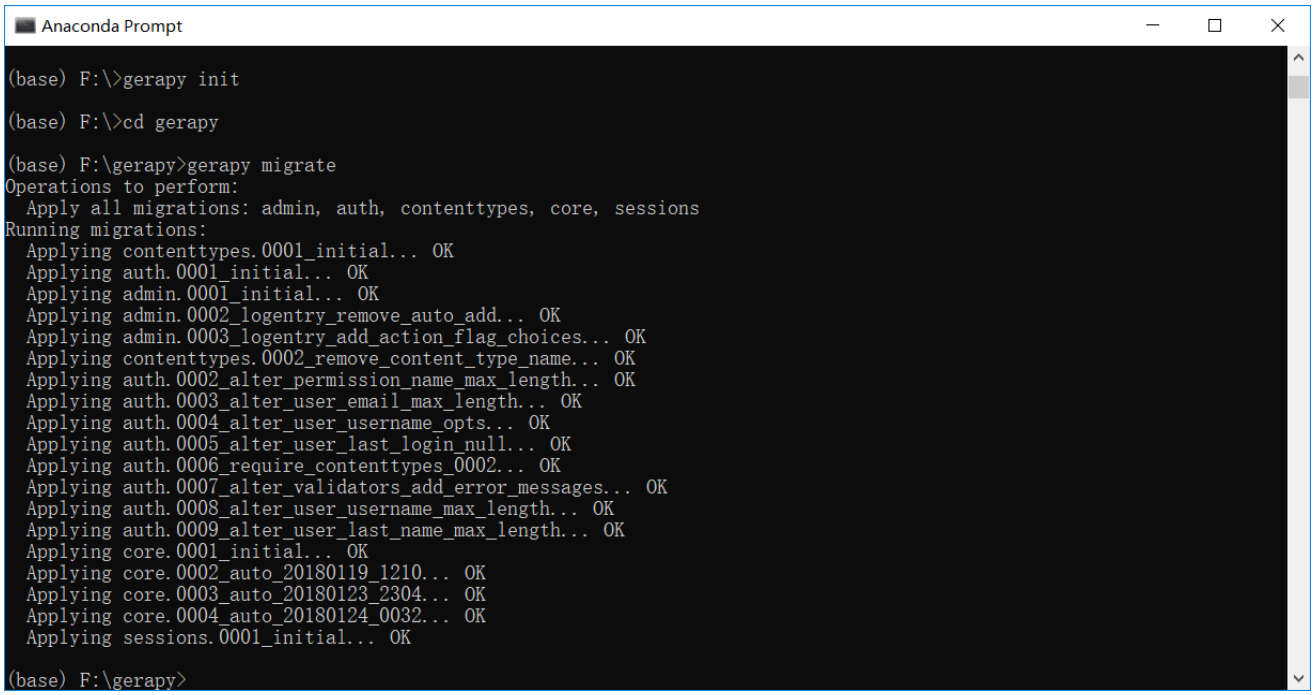
接下来我们来开始使用 Gerapy，首先利用如下命令进行一下初始化，在任意路径下均可执行如下命令：



执行完毕之后，本地便会生成一个名字为 gerapy 的文件夹，接着进入该文件夹，可以看到有一个 projects 文件夹，我们后面会用到。

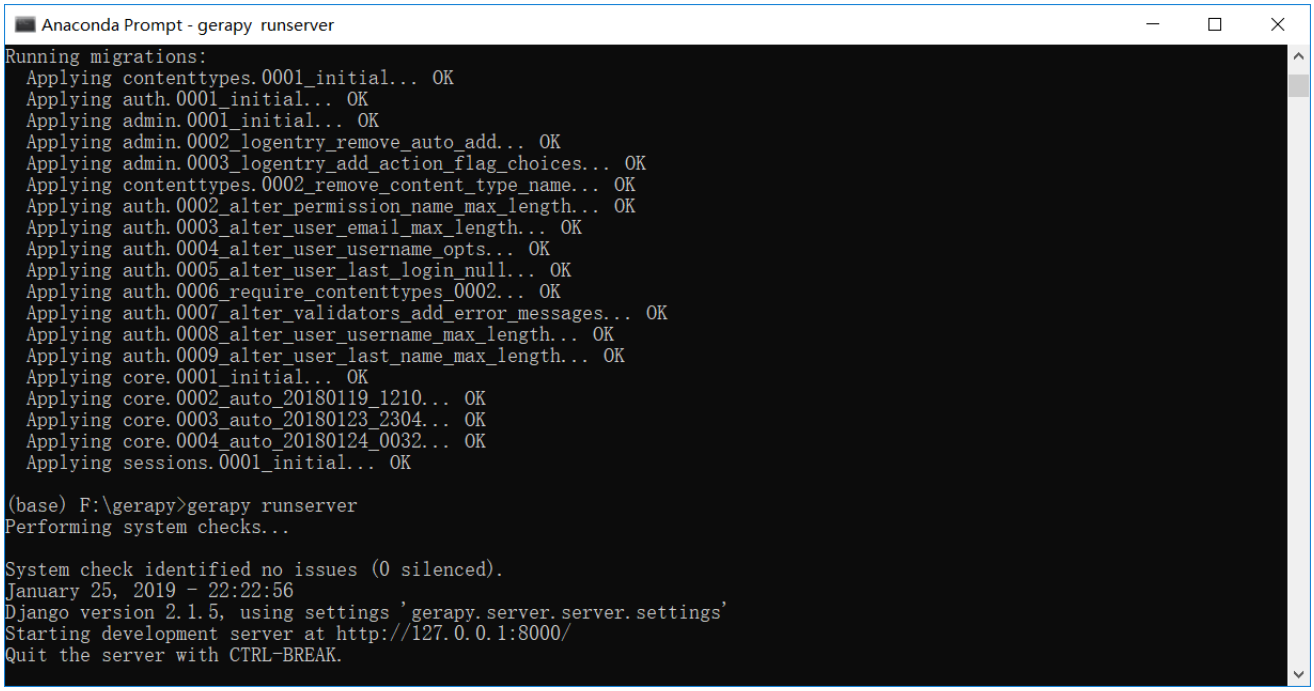


紧接着执行数据库初始化命令：

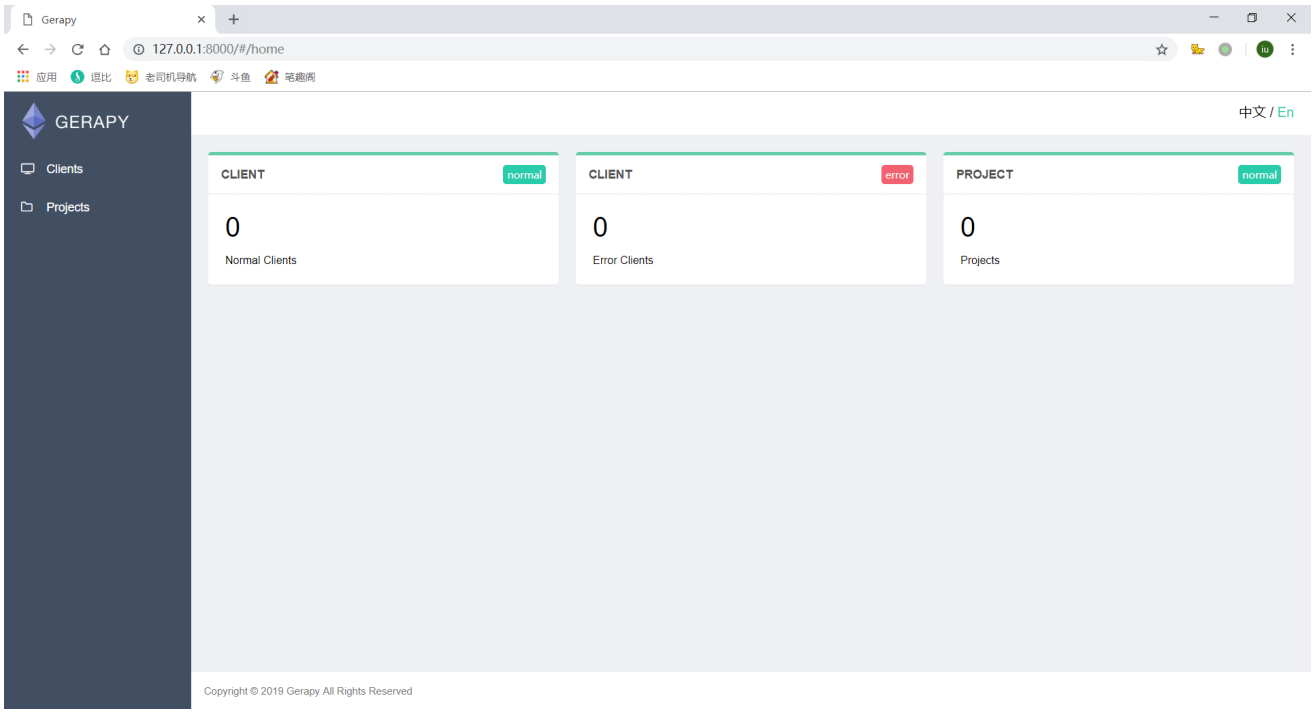


这样它就会在 gerapy 目录下生成一个 SQLite 数据库，同时建立数据库表。

接着我们只需要再运行命令启动 `gerapy` 服务就好了：



这样我们就可以看到 Gerapy 已经在 8000 端口上运行了。 我们使用浏览器访问 `127.0.0.1:8000`，看看gerapy界面了。完整 `gerapy` 界面如下：

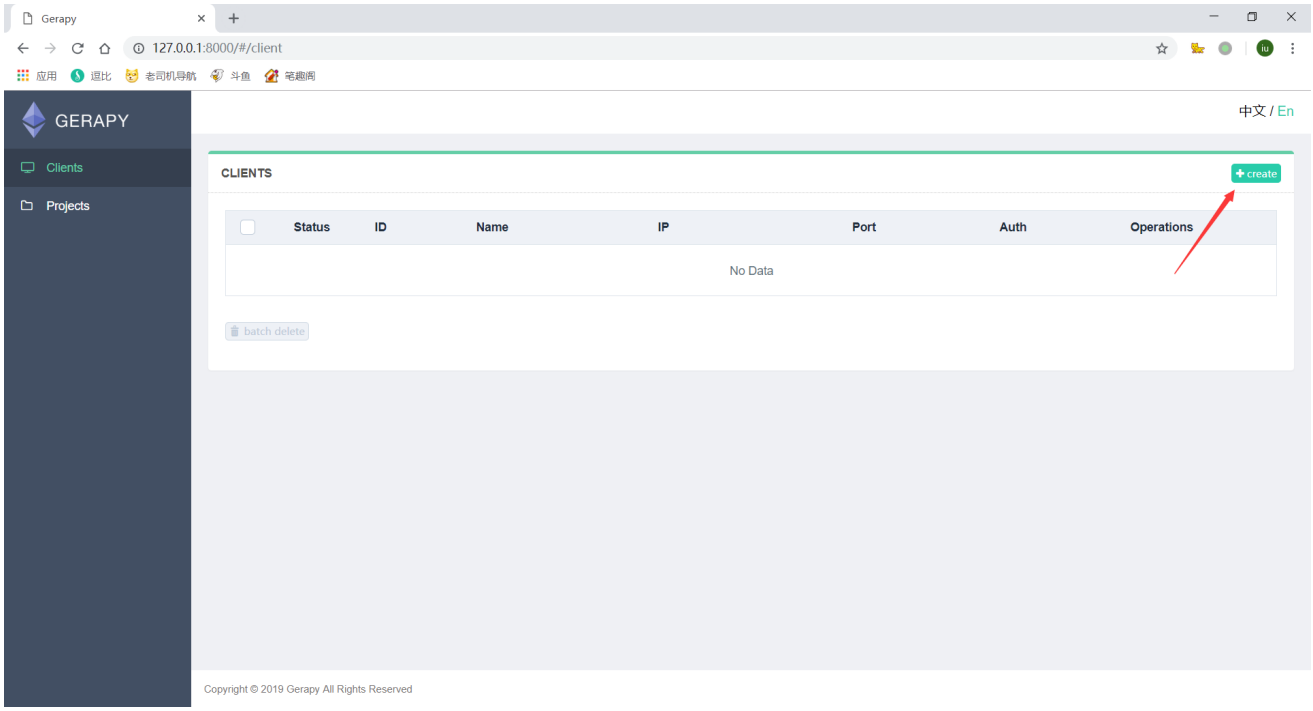


这里显示了主机、项目的状态，当然由于我们没有添加主机，所以所有的数目都是 0。

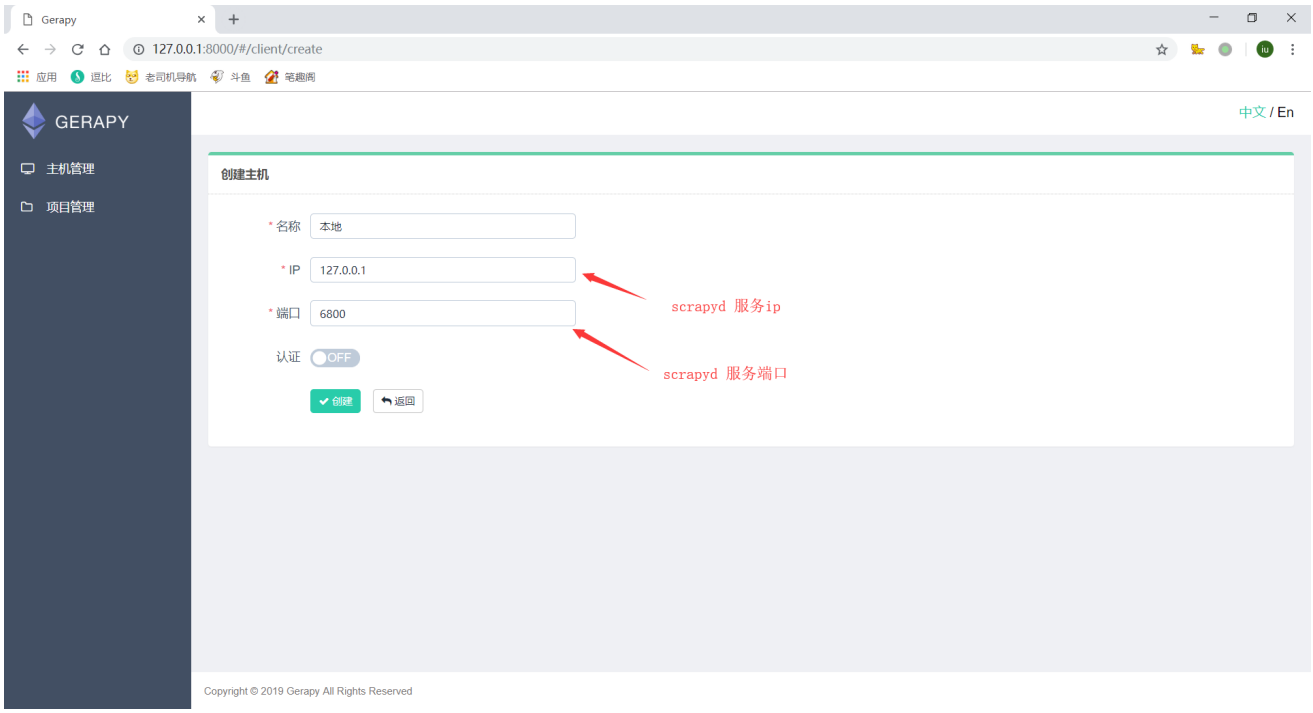
如果我们可以正常访问这个页面，那就证明 Gerapy 初始化都成功了。

主机管理

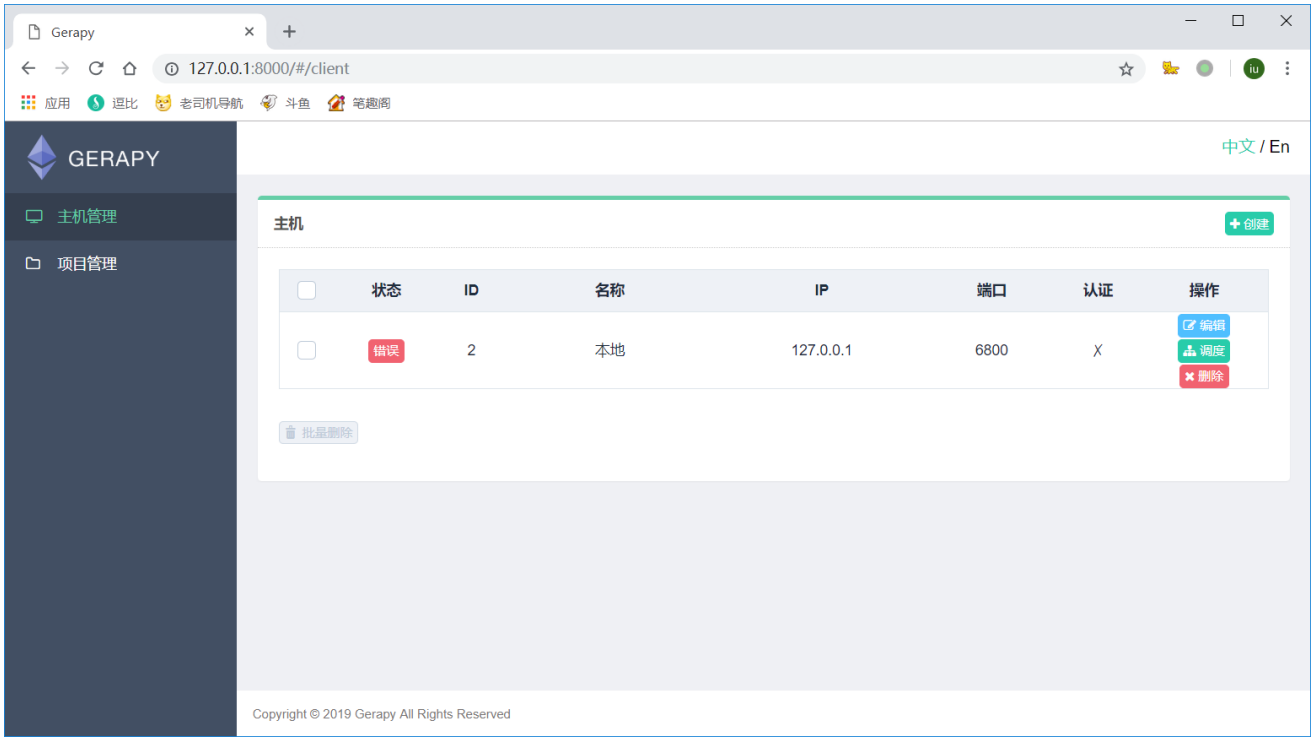
接下来我们可以点击左侧 Clients 选项卡，即主机管理页面，添加我们的 Scrapyd 远程服务，点击右上角的创建按钮即可添加我们需要管理的 Scrapyd 服务：



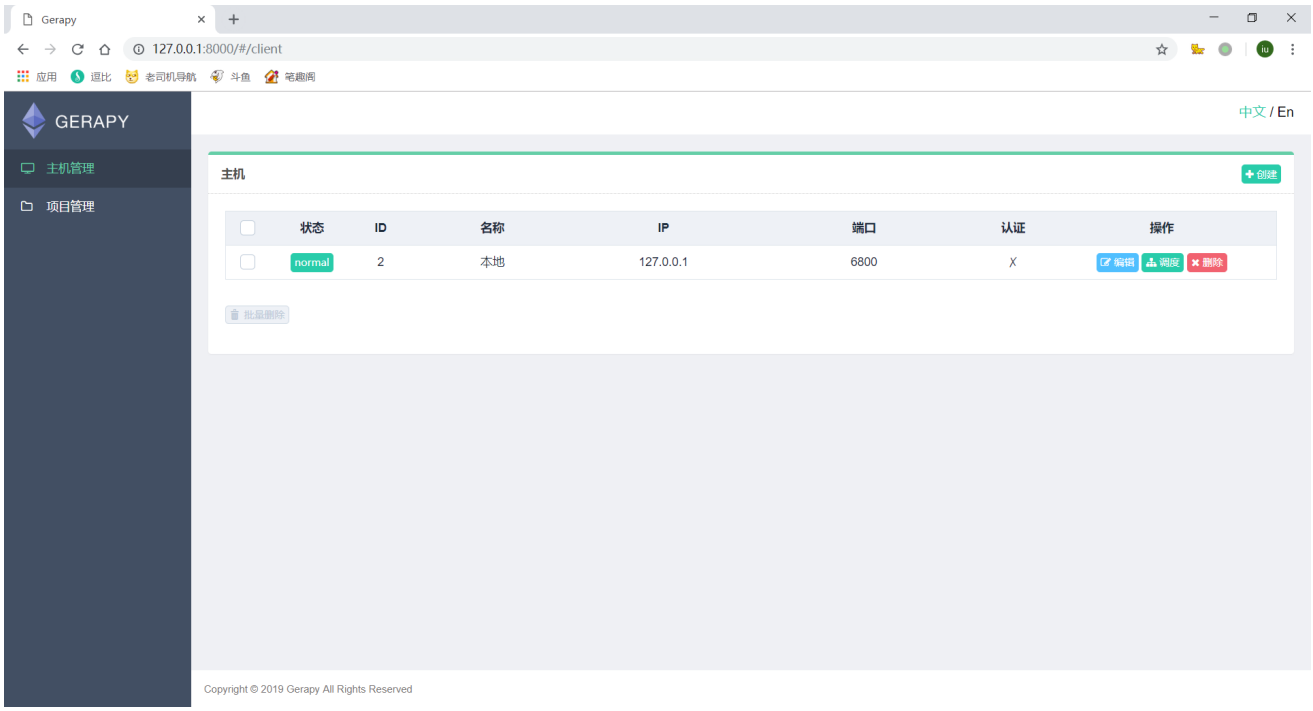
创建主机：



输入完成后点击创建，然后会自动报存，然后就可以在我们的主机管理看到我们刚刚创建的主机了：



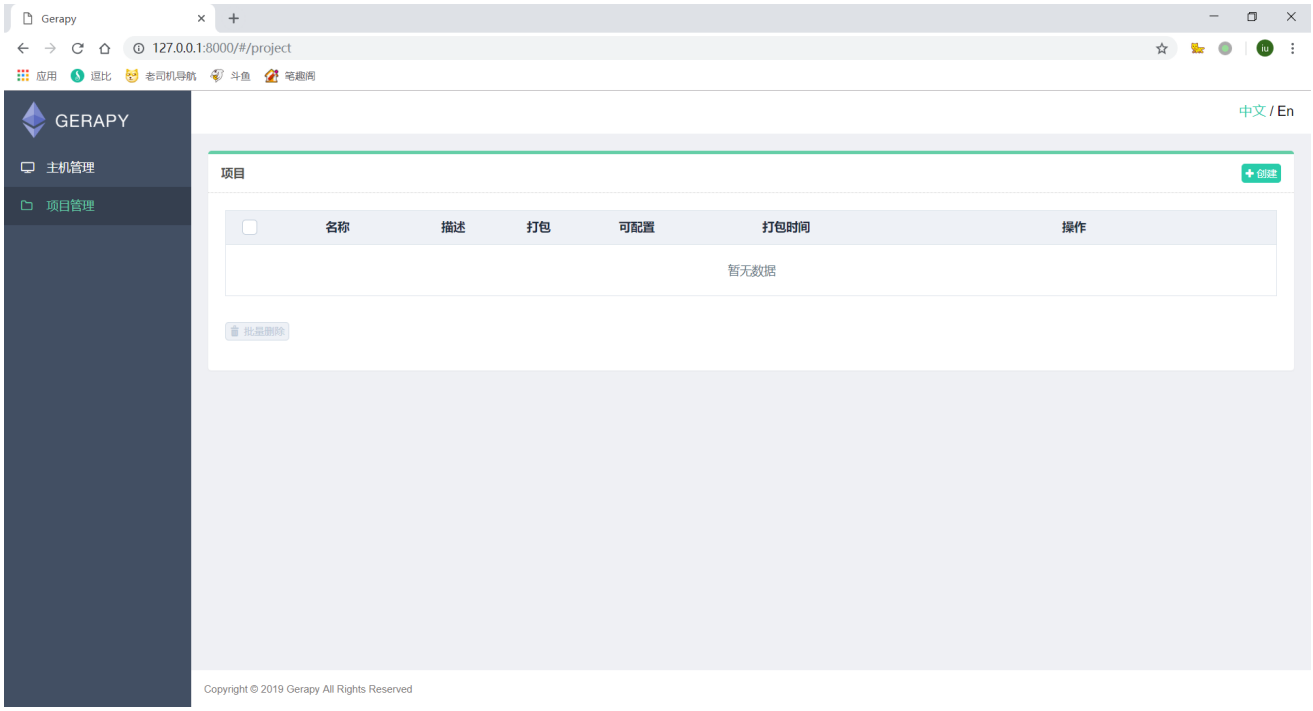
有没发现刚刚创建的主机的状态是错误的，说明 `gerapy` 服务没有访问 `scrapyd` 服务，我们需要开启或者重启 `scrapyd` 服务，开启或者重启后，刷新我们主机管理界面：



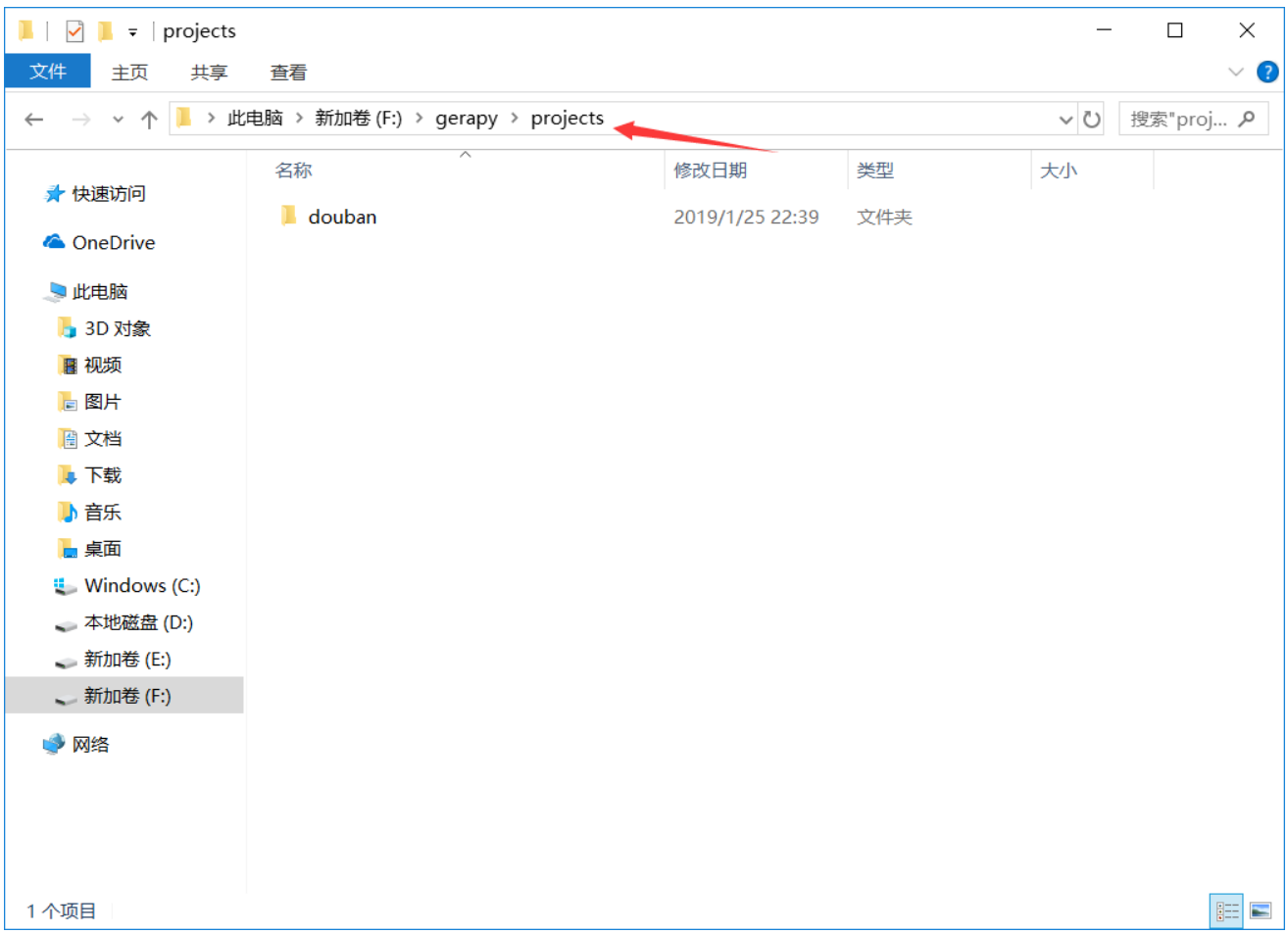
这样我们可以在状态一栏看到各个 Scrapyd 服务是否可用，同时可以一目了然当前所有 Scrapyd 服务列表，另外我们还可以自由地进行编辑和删除。

项目管理

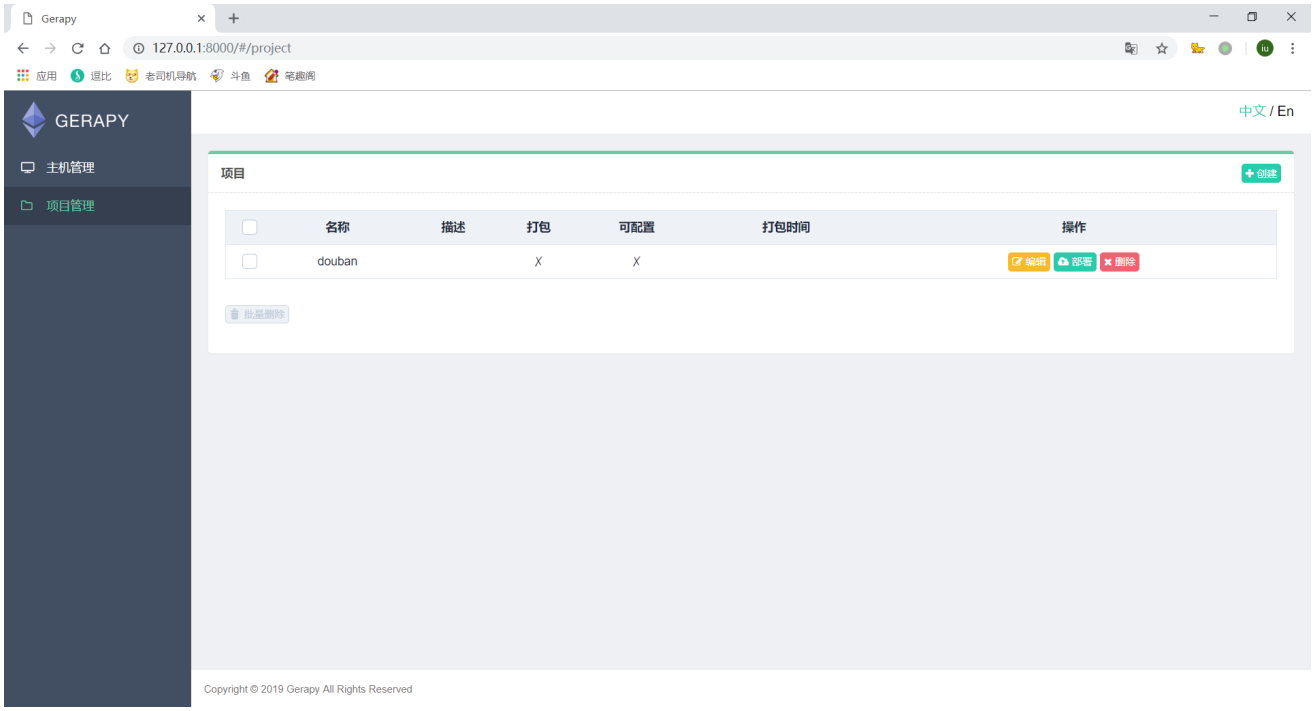
Gerapy 的核心功能当然是项目管理，在这里我们可以自由地配置、编辑、部署我们的 Scrapy 项目，点击左侧的 Projects ，即项目管理选项，我们可以看到如下空白的页面：



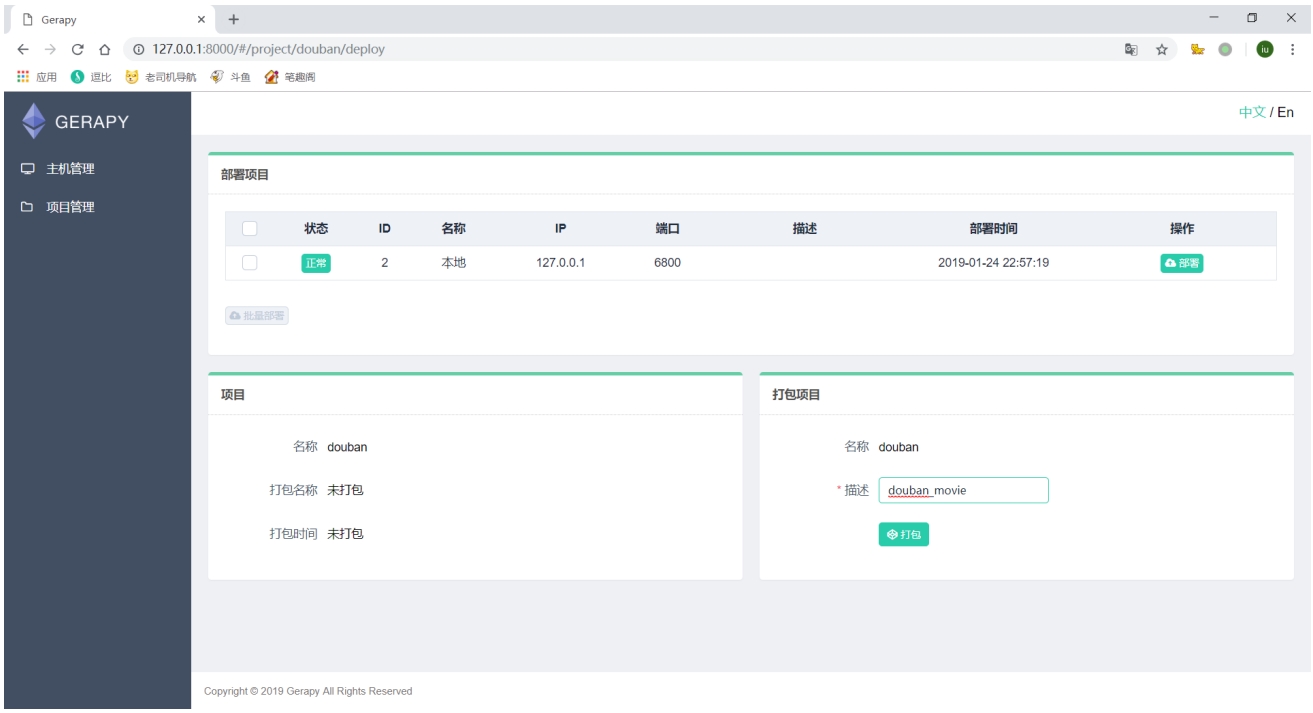
假设现在我们有一个 Scrapy 项目，如果我们想要进行管理和部署，还记得初始化过程中提到的 projects 文件夹吗？这时我们只需要将项目拖动到刚才 gerapy 运行目录的 projects 文件夹下，例如我这里还是拿上文的豆瓣电影top250项目爬虫，把它拖动到 projects 文件夹下：



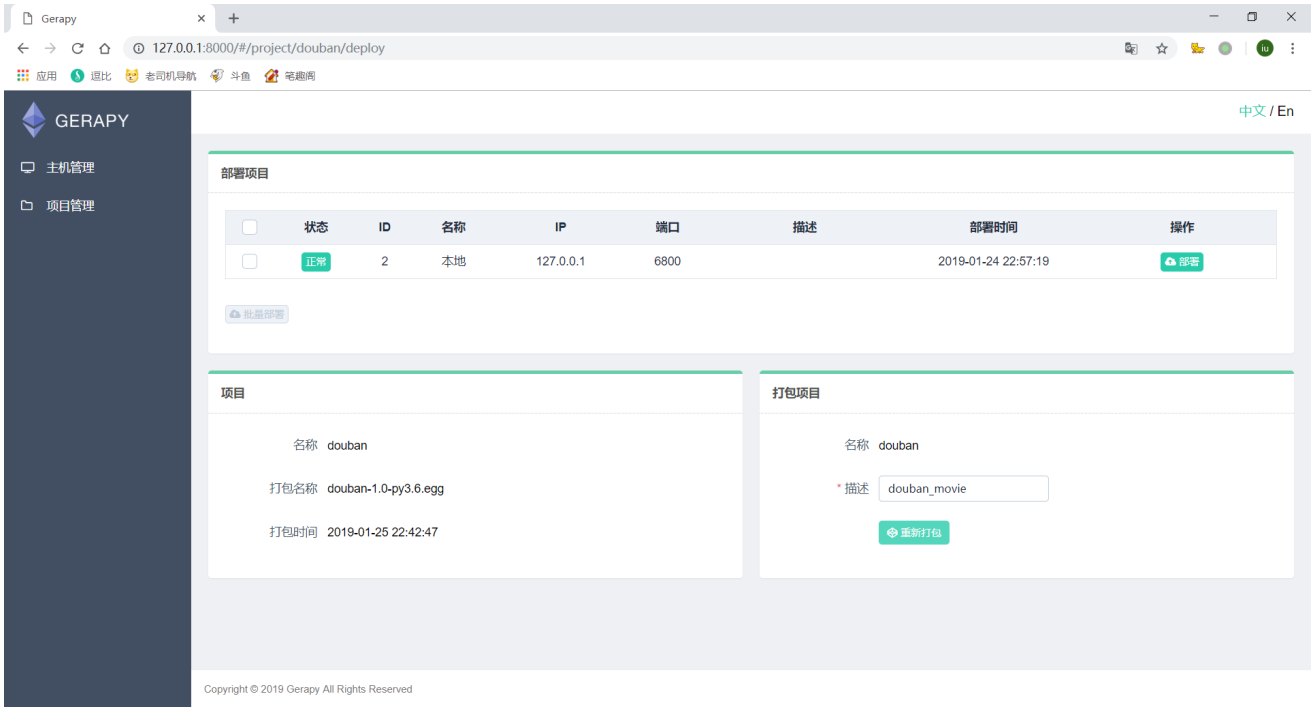
这时刷新项目管理页面，我们便可以看到 Gerapy 检测到了这个项目，同时它是不可配置、没有打包的：



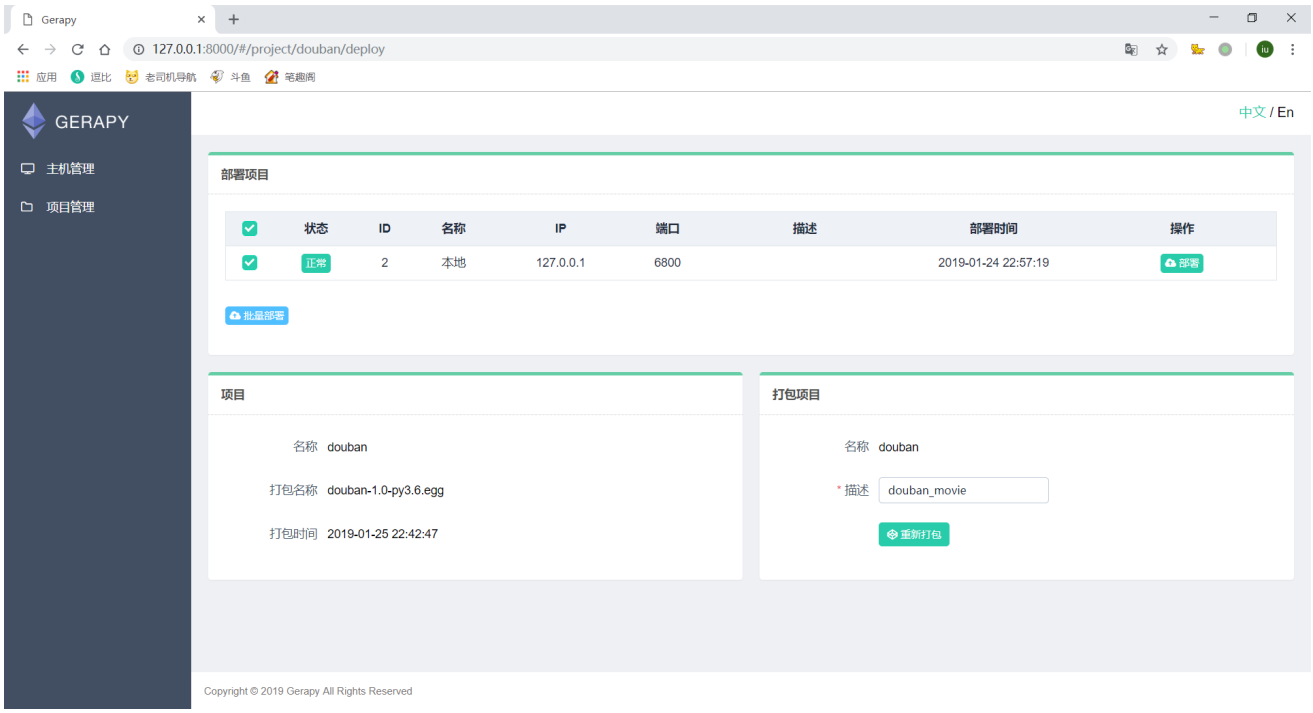
这时我们可以点击部署按钮进行打包和部署，在右下角我们可以输入打包时的描述信息：



然后点击打包按钮，即可发现 Gerapy 会提示打包成功，同时在左侧显示打包的结果和打包名称：

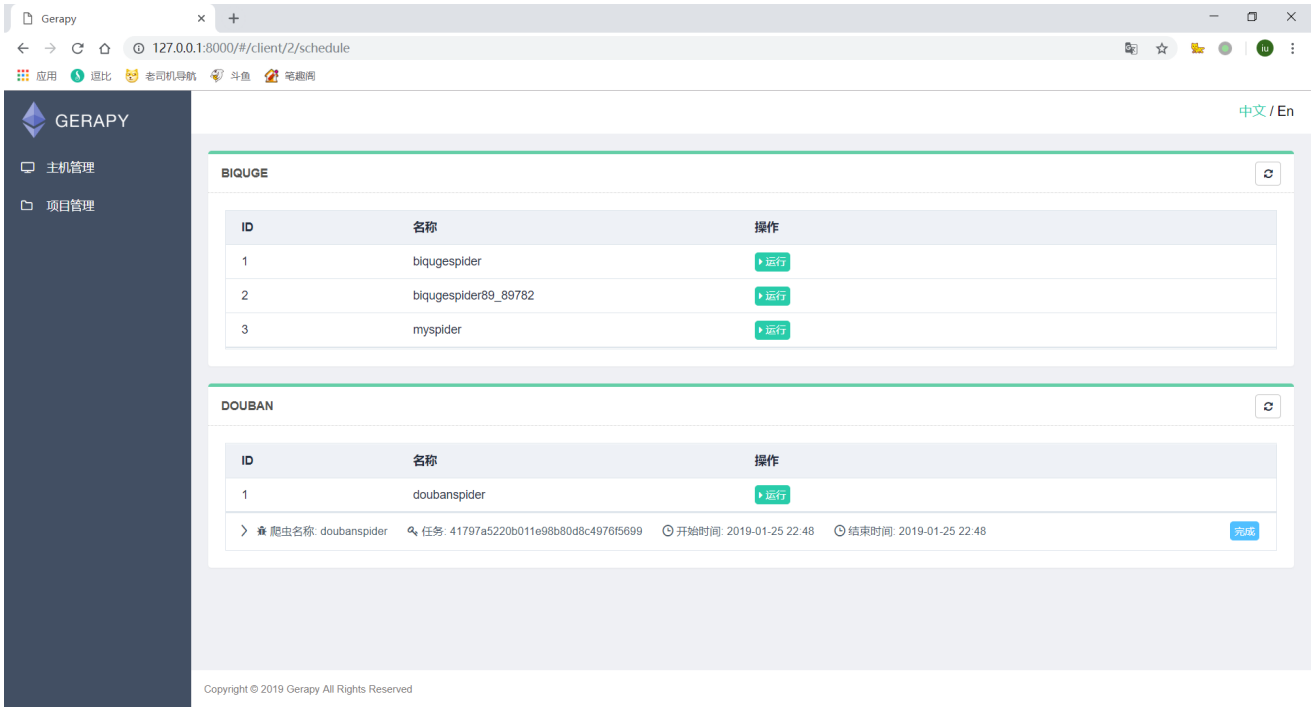


打包成功之后，我们便可以进行部署了，我们可以选择需要部署的主机，点击后方的部署按钮进行部署，同时也可以批量选择主机进行部署，示例如下：

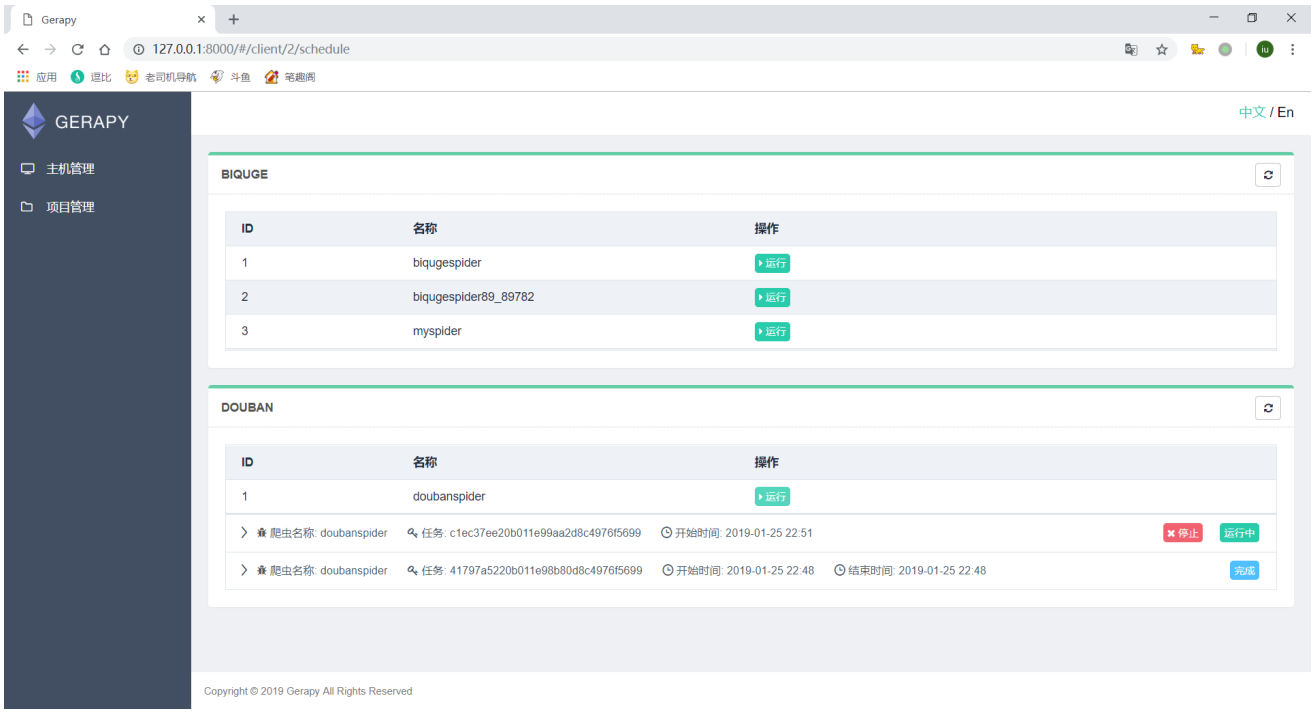


项目监控任务

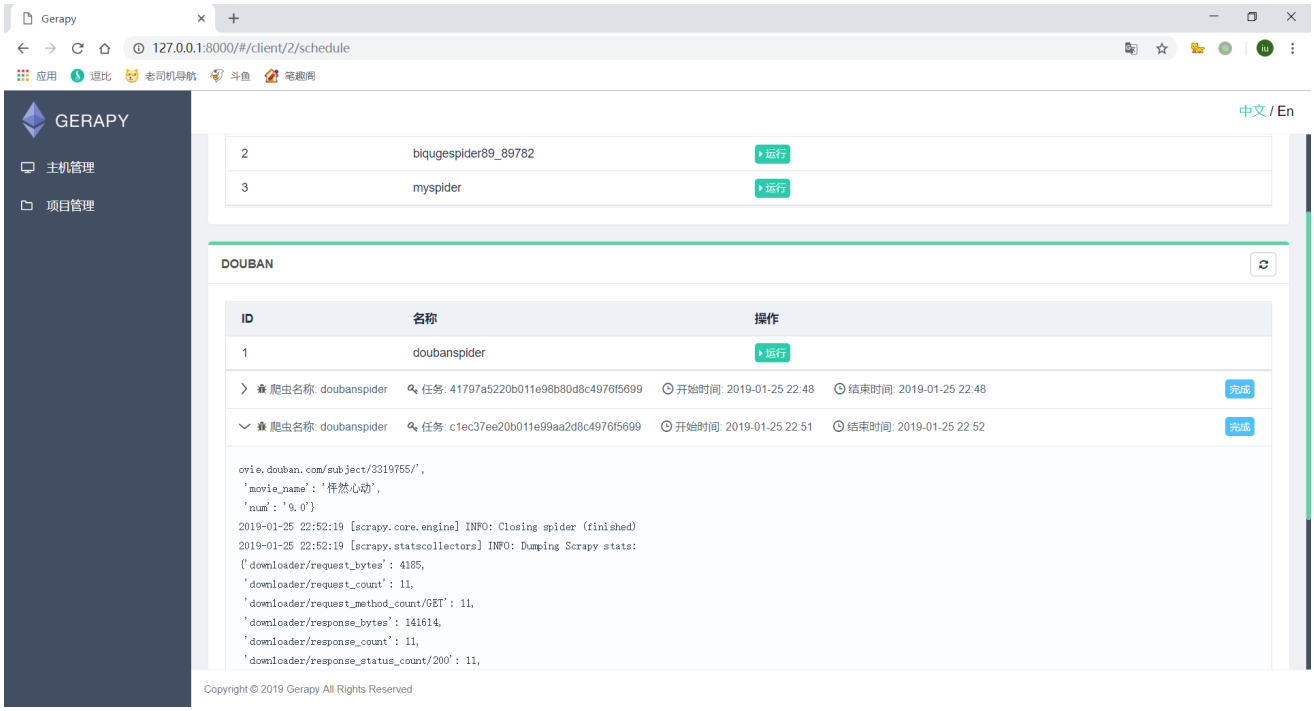
部署完毕之后就可以回到主机管理页面进行任务调度了，任选一台主机，点击调度按钮即可进入任务管理页面，此页面可以查看当前 Scrapyd 服务的所有项目、所有爬虫及运行状态：



我们可以通过点击新任务、停止等按钮来实现任务的启动和停止等操作，同时也可以通过展开任务条目查看日志详情：



我们可以通过点击新任务、停止等按钮来实现任务的启动和停止等操作，同时也可以通过展开任务条目查看日志详情：

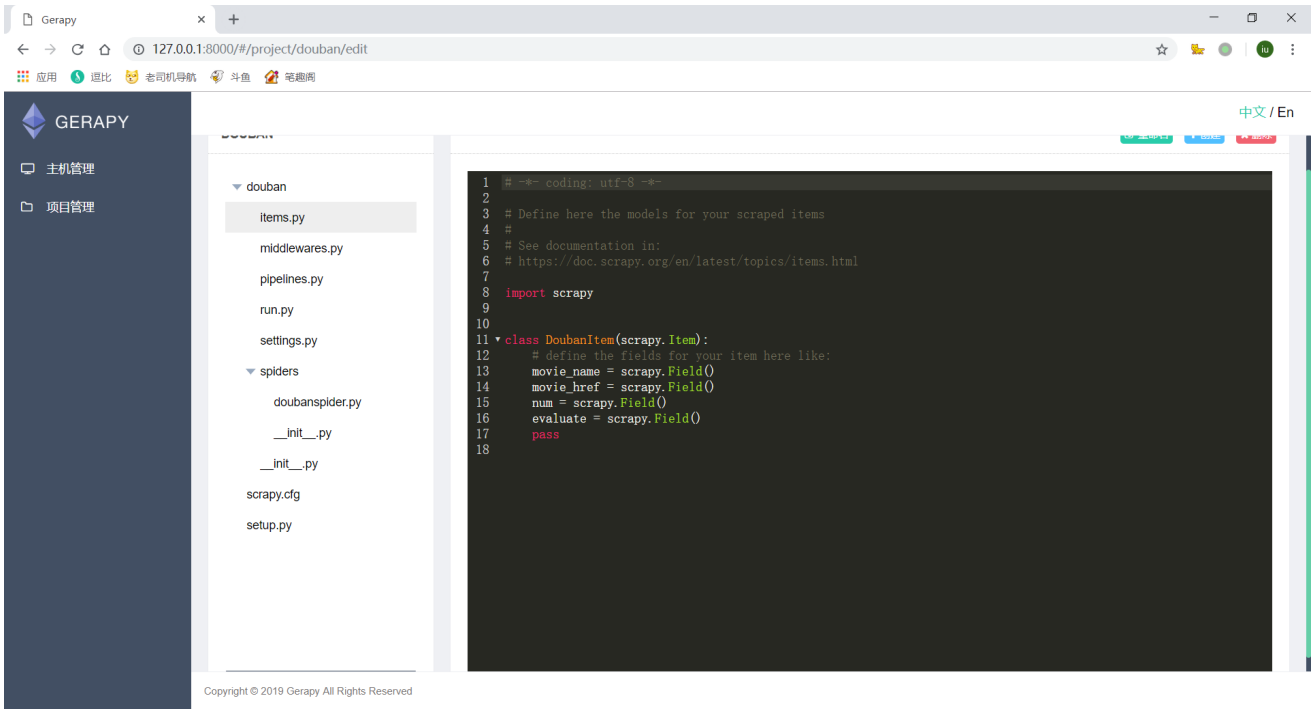


另外我们还可以随时点击停止按钮来取消 Scrapy 任务的运行。

这样我们就可以在此页面方便地管理每个 Scrapyd 服务上的 每个 Scrapy 项目的运行了。

项目编辑

同时 Gerapy 还支持项目编辑功能，有了它我们不再需要 IDE 即可完成项目的编写，我们点击项目的编辑按钮即可进入到编辑页面，如图所示：



这样即使 Gerapy 部署在远程的服务器上，我们不方便用 IDE 打开，也不喜欢用 Vim 等编辑软件，我们可以借助于本功能方便地完成代码的编写。

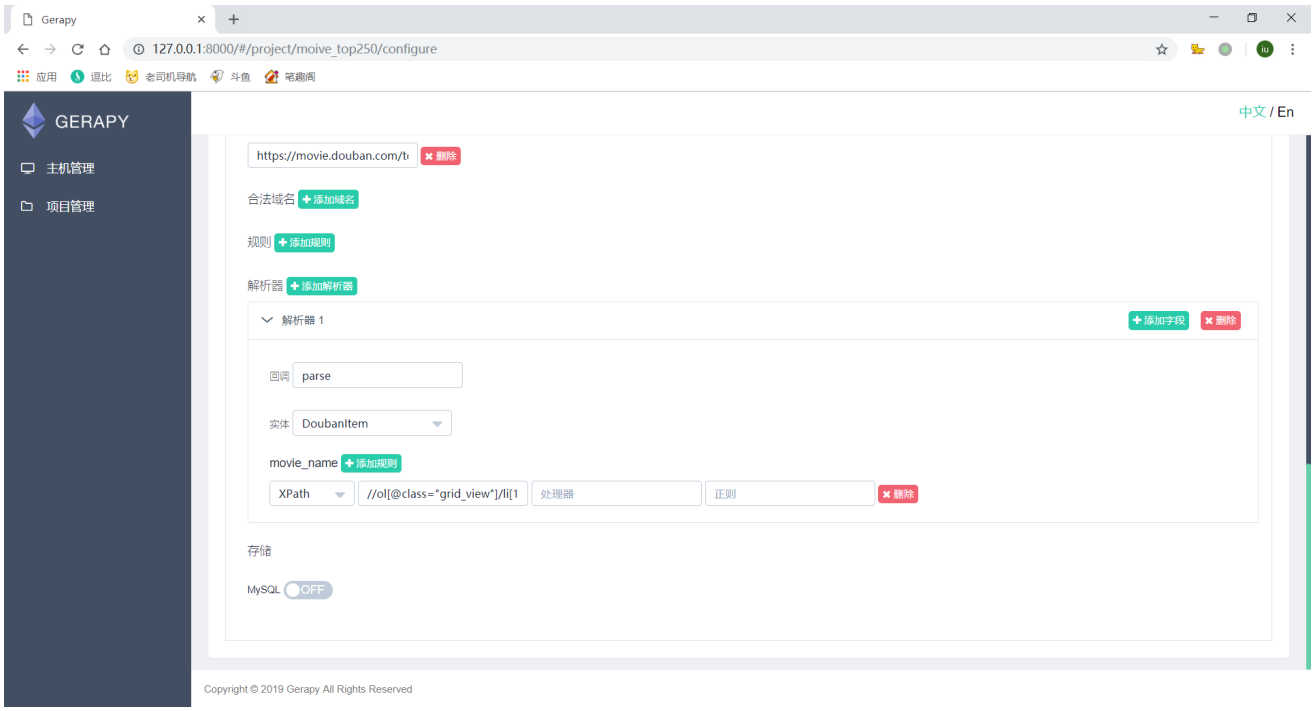
代码生成

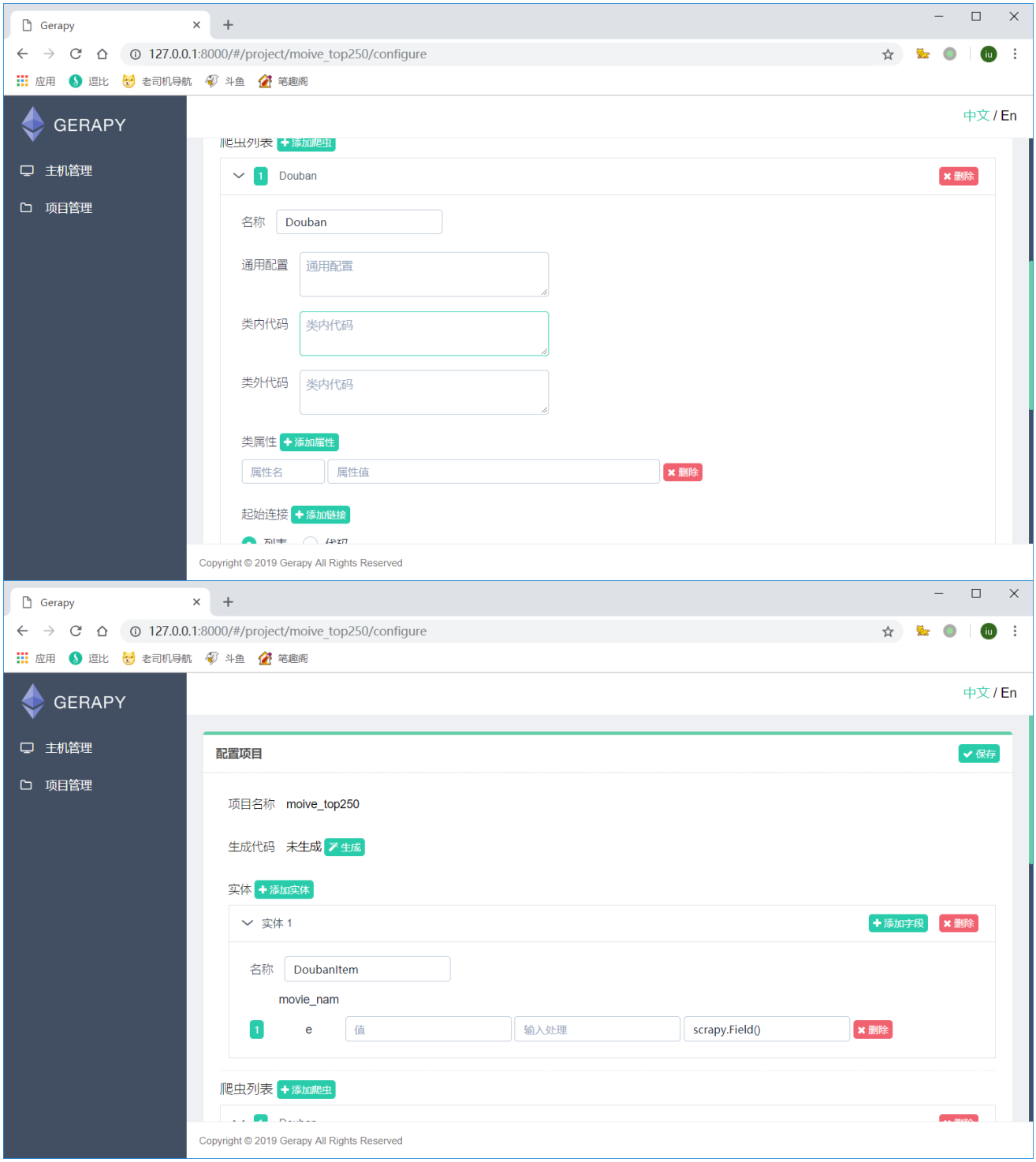
上述的项目主要针对的是我们已经写好的 Scrapy 项目，我们可以借助于 Gerapy 方便地完成编辑、部署、控制、监测等功能，而且这些项目的一些逻辑、配置都是已经写死在代码里面的，如果要修改的话，需要直接修改代码，即这些项目都是不可配置的。

在 Scrapy 中，其实提供了一个可配置化的爬虫 CrawlSpider，它可以利用一些规则来完成爬取规则和解析规则的配置，这样可配置化程度就非常高，这样我们只需要维护爬取规则、提取逻辑就可以了。如果要新增一个爬虫，我们只需要写好对应的规则即可，这类爬虫就叫做可配置化爬虫。

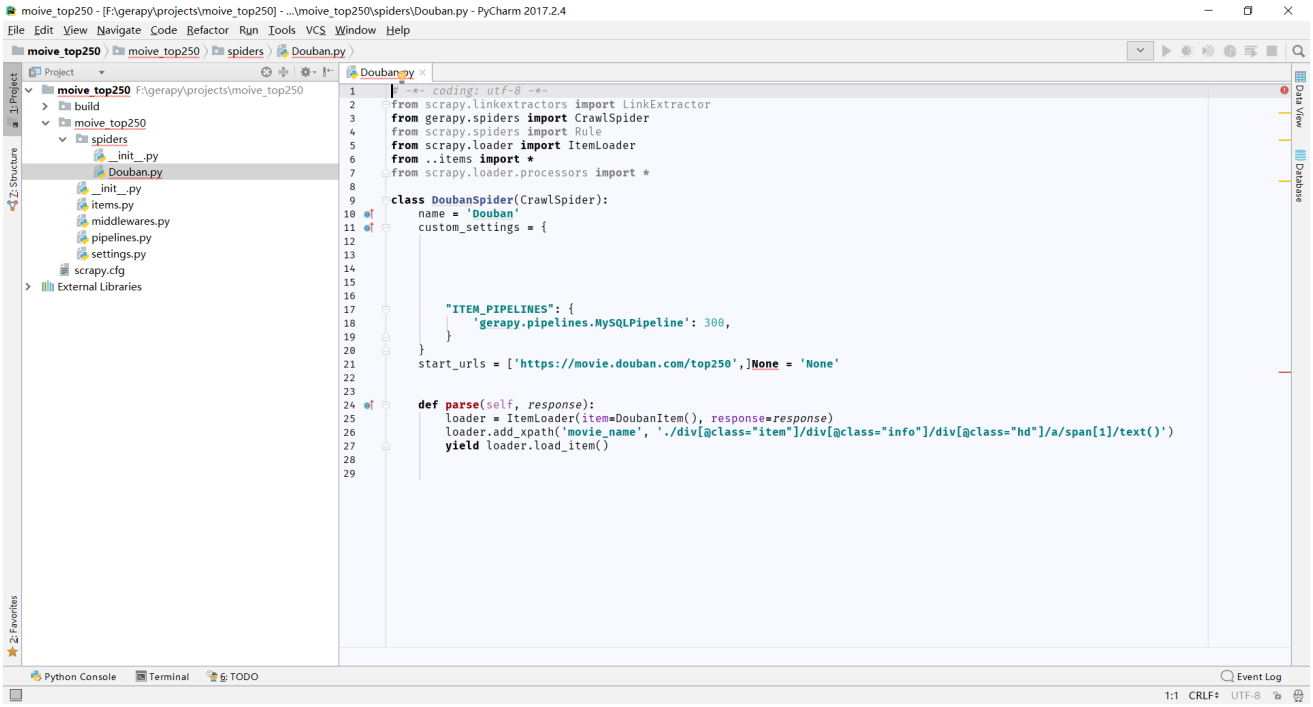
Gerapy 可以做到：我们写好爬虫规则，它帮我们自动生成 Scrapy 项目代码。

我们可以点击项目页面的右上角的创建按钮，增加一个可配置化爬虫，接着我们便可以在此处添加提取实体、爬取规则、抽取规则了，例如这里的解析器，我们可以配置解析成为哪个实体，每个字段使用怎样的解析方式，如 XPath 或 CSS 解析器、直接获取属性、直接添加值等多重方式，另外还可以指定处理器进行数据清洗，或直接指定正则表达式进行解析等等，通过这些流程我们可以做到任何字段的解析。



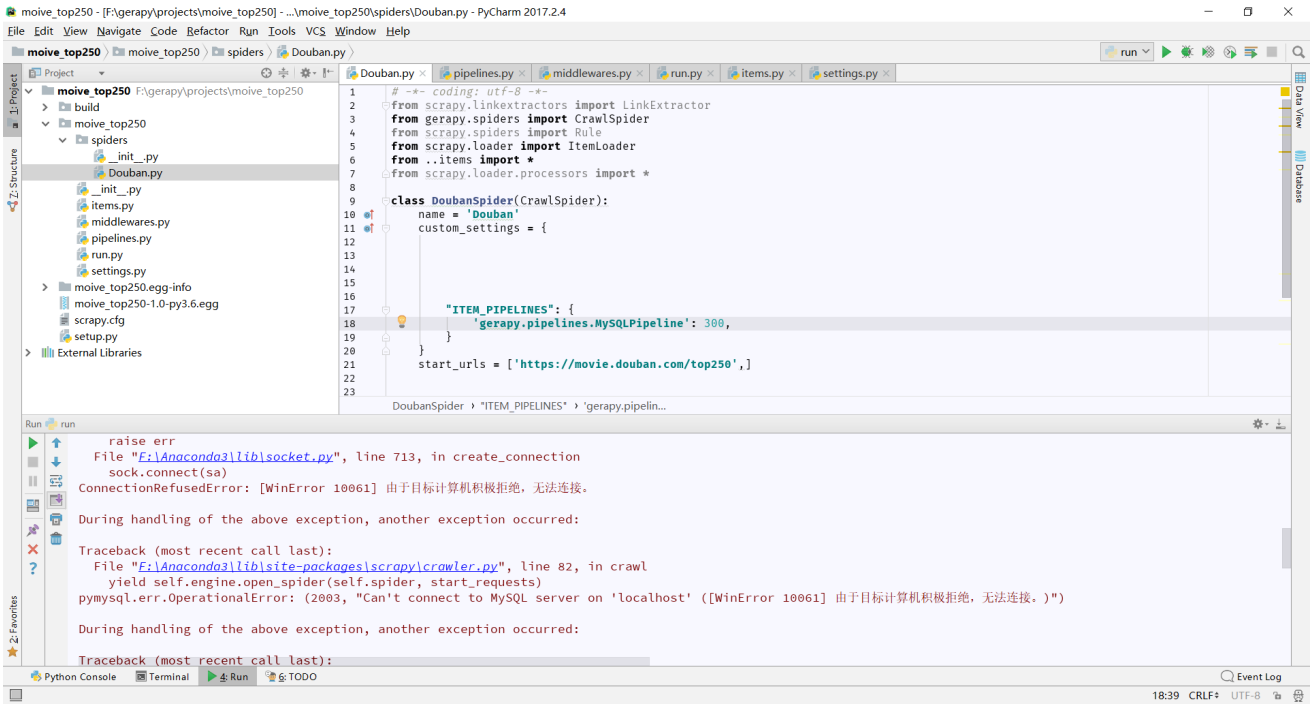


然后点击代码生成，生成的代码和 `scrapy` 项目结构是一样的：



笔者不用 Gerapy 生成代码功能，生产的代码还有错误，我们修改下，然后我们像之前项目看看能不能部署运行。

笔者打包成功了，尝试部署了很多次没有成功部署上，可能还是我们在界面上编写的代码不规范，没有让 Gerapy 读取我们的代码。然后尝试了直接运行，刚开始不能正常运行，然后修改了部分代码提示了我们如下图错误：



错误提示我们说连接 mysql 数据库失败，我们在界面上配置代码时，没有选择使用 mysql 存储 ， 说明 这个 Gerapy 生产代码部分开发的不是非常好用。