核心代码只有三个，前端main.html，后端main.py，爬虫代码config\_policy\_spider\spiders\gov\_policy\_spider.py。

后端main.py介绍：总共有8个接口，下面一一介绍。

@app.post("/submit\_form")接口，作用是将前端传递过来的配置信息保存到项目中的config.json中。

@app.post("/start\_scrapy")接口，异步执行一个爬虫代码，将爬虫代码放入到线程中执行（只能放入到线程中，如果放入到子进程，会因为暂停操作，导致整个后端一起跟着暂停）。使用scrapy\_instances列表来管理全部实例。并且使用管道来流式读取爬虫代码的输出。

@app.get("/stream\_scrapy")接口，作用是提供一个接口，方便前端读取管道里的输出。

@app.post("/pause\_scrapy")接口，作用是暂停爬虫进程。

@app.post("/resume\_scrapy")接口，作用是恢复被暂停的爬虫进程。

@app.post("/stop\_scrapy")接口，作用是终止爬虫进程。由于有io操作，暂停之后需要等一会才能终止完全。

@app.get("/list\_instances")接口，展示目前在scrapy\_instances列表里管理的全部实例。

@app.post("/import\_csv\_to\_postgres")接口，执行数据库写入操作。加上了表级锁，防止写错误。同时也因为表级锁，不能同时写同一张表。

爬虫代码介绍：执行逻辑是先执行start\_requests函数，然后执行parse\_list函数，再然后执行parse\_detail函数。

start\_requests函数作用：解析配置，启动爬虫。

parse\_list函数作用：找到网页url下的政策列表位置，一一点击进去执行parse\_detail函数来爬取内容。

parse\_detail函数作用：对点击进入的网页爬取config.json定义要爬的字段。

Main.html前端介绍：太长了，但基本上就是按钮点击绑定，点击按钮触发事件，然后调用后端的接口。