

```
#展示模板
scrapy genspider -l
```

```
PS C:\Users\23303\Desktop\scrapy相关\fang> scrapy genspider -l
Available templates:
  basic
  crawl
  csvfeed
  xmlfeed
PS C:\Users\23303\Desktop\scrapy相关\fang> |
```

scrapy genspider -t crawl 爬虫名 首个请求的网址

这里还是以

```
https://www.xiaoshuopu.com
```

为例子

```
from scrapy.linkextractors import LinkExtractor
from scrapy.spiders import CrawlSpider, Rule

class Baiduspider(CrawlSpider):
    name = "baidu"
    allowed_domains = ["xiaoshuopu.com"]
    start_urls = [f"https://www.xiaoshuopu.com/class_{type}/" for type in
range(1,4)]
    Rule(
        LinkExtractor(
            allow=r'https://www.xiaoshuopu.com/xiaoshuo/\d+/\d+/',
            deny=r'https://www.xiaoshuopu.com/xiaoshuo/\d+/\d+/\d+\.html',
            # restrict_xpaths='//*[@id="at"]'
        ),
        callback='parse_books',
        follow=True
    ),
    Rule(
        LinkExtractor(
            allow=r'https://www.xiaoshuopu.com/xiaoshuo/\d+/\d+/\d+\.html',
            restrict_xpaths='//*[@id="at"]'
        ),
        callback='parse_item',
        follow=False
    ),
)

def parse_books(self, response):
    book_name = response.xpath('//h1/text()').extract()
    print('book_name:', book_name, response.url)

def parse_item(self, response):
```

```
title = response.xpath('//*[@id="amain"]/dl/dd[1]/h1/text()').extract()
content = response.xpath('//*[@id="htmlContent"]//text()').extract()
print('title:',title,response.url)
# item = {}
#item["domain_id"] = response.xpath('//input[@id="sid"]/@value').get()
#item["name"] = response.xpath('//div[@id="name"]').get()
#item["description"] = response.xpath('//div[@id="description"]').get()
# return item
```

这里

allow=r'<https://www.xiaoshuopu.com/xiaoshuo/>\d+\d+/',

是打算匹配所有书对应的页面，比如这种



follow=True 代表可以继续跟进，也就是后面的规则允许在对应的页面进行规则匹配，这里我们还想获取这个页面上章节的链接然后请求，所以肯定是要跟进的

deny=r'<https://www.xiaoshuopu.com/xiaoshuo/>\d+\d+\d+.html',这个是不允许访问什么，可以看到文章的详情页,这个链接符合了 allow=r'<https://www.xiaoshuopu.com/xiaoshuo/>\d+\d+/',这个表达式的要求，但是实际我们不想要，那可以用deny排除，deny的优先级要比allow高。

```
https://www.xiaoshuopu.com/xiaoshuo/17/17512/7678526.html
```



之后我们可以单起一个rule来匹配详情页内的东西

```
Rule(  
    LinkExtractor(  
        allow=r'https://www.xiaoshuopu.com/xiaoshuo/\d+/\d+/\d+\.html',  
        restrict_xpaths='//*[@id="at"]'  
    ),  
    callback='parse_item',  
    follow=False  
)
```

由于小说的详情页里没有具体的链接要继续抓取，所以这里的follow就改为false即可