链接提取器

链接提取器是对象，其唯一目的是从 `scrapy.http.Response` 最终将遵循的网页（对象）中提取链接。

目前 `scrapy.linkextractors.LinkExtractor` 在Scrapy，但您可以创建自己的自定义链接提取器通过一个简单的界面，以满足您的需求。

每个链接提取器唯一的公共方法是 `extract_links` 接收 `Response` 对象并返回 `scrapy.link.Link` 对象列表。链接提取器旨在实例化一次，并且 `extract_links` 使用不同的响应多次调用它们的方法以提取要遵循的链接。

链接提取器在 `CrawlSpider` 类中使用（在Scrapy中可用），通过一组规则，但您也可以在您的蜘蛛中使用它，即使您没有从中进行子类化 `CrawlSpider` ，因为它的目的非常简单：提取链接。

内置链接提取器参考

`scrapy.linkextractors` 模块中提供了与Scrapy捆绑在一起的链接提取器类 。

默认链接提取器是 `LinkExtractor` ，它与以下内容相同 `LxmlLinkExtractor` ：

```
from scrapy.linkextractors import LinkExtractor
```

以前的Scrapy版本中曾经有过其他链接提取器类，但现在已弃用。

LxmlLinkExtractor

class **scrapy.linkextractors.lxmlhtml.LxmlLinkExtractor**(*allow = ( )，deny = ( )，allow_domains = ( )，deny_domains = ( )，deny_extensions = None，restrict_xpaths = ( )，restrict_css = ( )，tags = ('a'，'area')，attrs = ('href'，)，canonicalize = False，unique = True，process_value = None，strip = True*)

LxmlLinkExtractor是推荐的链接提取器，具有方便的过滤选项。它是使用lxml强大的HTMLParser实现的。

参数：

- **allow**（*正则表达式（或列表）*） - 一个正则表达式（或正则表达式列表），（绝对）URL必须匹配才能被提取。如果没有给出（或为空），它将匹配所有链接。

- **deny**（*正则表达式（或列表）*） - 单个正则表达式（或正则表达式列表），（绝对）URL必须匹配才能被排除（即未提取）。它优先于`allow`参数。如果没有给出（或为空），它将不排除任何链接。
- **allow_domains**（*str 或list*） - 单个值或包含域的字符串列表，用于提取链接
- **deny_domains**（*str 或list*） - 单个值或包含域的字符串列表，不考虑用于提取链接
- **deny_extensions**（*list*） - 包含扩展名的单个值或字符串列表，在提取链接时应忽略这些扩展名。如果没有给出，它将默认为scrapy.linkextractors包`IGNORED_EXTENSIONS`中定义的 列表 。
- **restrict_xpaths**（*str 或list*） - 是一个XPath（或XPath列表），它定义响应中应从中提取链接的区域。如果给定，则仅扫描由这些XPath选择的文本以获取链接。见下面的例子。
- **restrict_css**（*str 或list*） - 一个CSS选择器（或选择器列表），用于定义响应中应从中提取链接的区域。有与...相同的行为`restrict_xpaths`。
- **tags**（*str 或list*） - 提取链接时要考虑的标记或标记列表。默认为。`('a', 'area')`
- **attrs**（*list*） - 查找要提取的链接时应考虑的属性或属性列表（仅适用于`tags`参数中指定的那些标记）。默认为`('href',)`
- **canonicalize**（*boolean*） - 规范化每个提取的url（使用w3lib.url.canonicalize_url）。默认为`False`。请注意，canonicalize_url用于重复检查; 它可以更改服务器端可见的URL，因此对于具有规范化和原始URL的请求，响应可能不同。如果您使用LinkExtractor跟踪链接，则保持默认值更加健壮`canonicalize=False`。
- **unique**（*boolean*） - 是否应对提取的链接应用重复过滤。
- **process_value**（*可调用*） - 接收从标签中提取的每个值和扫描的属性的函数，可以修改该值并返回一个新值，或者返回`None`完全忽略该链接。如果没有给出，`process_value`默认为。`lambda x: x`

  例如，要从此代码中提取链接：

  ```
  <a href="javascript:goToPage('../other/page.html'); return false">Link
  text</a>
  ```

  您可以使用以下功能`process_value`：

  ```
  def process_value(value):
      m = re.search("javascript:goToPage\('(.*?)'", value)
      if m:
          return m.group(1)
  ```

- **strip**（*boolean*） - 是否从提取的属性中去除空格。根据HTML5标准，前导和尾部空格必须从被剥离`href`的属性`<a>`，`<area>`以及许多其他的元素，`src`属性`<img>`，`<iframe>`元件等，所以LinkExtractor默认条空间字符。设置`strip=False`为关闭它（例如，如果您从允许前导/尾随空格的元素或属性中提取URL）。