**如何进行运行前测试？**

在运行爬虫框架前，我们可能需要做很多准备工作  
比如：登录验证测试、内容提取规则测试  
这个时候我们就可以把PHPSpider当做类库来使用，获取单页面HTML并测试提取规则

**内容提取测试**

接下来我们以epooll这个站点的谋篇文章为例来演示内容提取方法

**获取HTML内容**

$url = "http://www.epooll.com/archives/806/";

$html = requests::get($url);

**提取文章标题**

// 选择器规则

$selector = "//div[contains(@class,'page-header')]//h1/a";

// 提取结果

$result = selector::select($html, $selector);

echo $result;

**提取文章作者**

$selector = "//div[contains(@class,'page-header')]//h6/span[1]";

$result = selector::select($html, $selector);

// 处理数据

$result = str\_replace("作者：", "", $result);

echo $result;

**提取文章入库完整示例**

$url = "http://www.epooll.com/archives/806/";

$html = requests::get($url);

// 抽取文章标题

$selector = "//div[contains(@class,'page-header')]//h1/a";

$title = selector::select($html, $selector);

// 检查是否抽取到标题

//echo $title;exit;

// 抽取文章作者

$selector = "//div[contains(@class,'page-header')]//h6/span[1]";

$author = selector::select($html, $selector);

// 检查是否抽取到作者

//echo $author;exit;

// 去掉 作者：

$author = str\_replace("作者：", "", $author);

// 抽取文章内容

$selector = "//div[contains(@class,'entry-content')]";

$content = selector::select($html, $selector);

// 检查是否抽取到内容

//echo $author;exit;

$data = array(

'title' => $title,

'author' => $author,

'content' => $content,

);

// 查看数据是否正常

//print\_r($data);

// 入库

db::insert("content", $data);

**运行PHPSpider**

通过上面的测试，我们就找出了文章内容页的field规则，配置到fields，然后调用PHPSpider

'fields' => array(

// 文章标题

array(

'name' => "article\_title",

'selector' => "//div[contains(@class,'page-header')]//h1/a",

'required' => true,

),

// 文章作者

array(

'name' => "article\_author",

'selector' => "//div[contains(@class,'page-header')]//h6/span[1]",

'required' => true,

),

// 文章内容

array(

'name' => "article\_content",

'selector' => "//div[contains(@class,'entry-content')]",

'required' => true,

),

)