AI+Node.js x-crawl 爬虫：为何传统爬虫已不再是数据抓取的首选?

AI 和 Node.js 爬虫结合

当 AI 搭配 Node.js 爬虫，这种组合将使得数据收集变得更加智能和高效。AI 可以帮助 Node.js 爬虫进行更加精准的目标定位。传统的爬虫往往依赖于固定的规则或模板进行数据的抓取，但这种方式在面对复杂多变的网页结构时往往力不从心。

为什么会需要 AI 辅助爬虫

随着网络技术的日新月异，网站更新变得愈发频繁，而类名或结构的改变往往给依赖这些元素的爬虫带来不小的挑战。在这样的背景下，结合 AI 技术的爬虫成为了应对这一挑战的有力武器。

首先，网站更新后类名或结构的改变可能导致传统的爬虫抓取策略失效。这是因为爬虫通常依赖于固定的类名或结构来定位并提取所需信息。一旦这些元素发生变化，爬虫就可能无法准确找到所需数据，从而影响数据抓取的效果和准确性。

然而，结合 AI 技术的爬虫则能够更好地应对这种变化。AI 还可以通过自然语言处理等技术，理解并解析网页的语义信息，从而更准确地提取所需数据。

什么是 x-crawl ?

x-crawl 是一个灵活的 Node.js AI 辅助爬虫库。灵活的使用方式和强大的 AI 辅助功能，使爬虫工作变得更加高效、智能和便捷。

它由两部分组成：

* 爬虫：由爬虫 API 以及各种功能组成，即使不依靠 AI 也能正常工作。
* AI：目前基于 OpenAI 提供的 AI 大模型，让 AI 简化很多繁琐的操作。

x-crawl GitHub: [github.com/coder-hxl/x...](http://cxyroad.com/”https://github.com/coder-hxl/x-crawl”)

x-crawl 文档: [coder-hxl.github.io/x-crawl/cn/](http://cxyroad.com/”https://coder-hxl.github.io/x-crawl/cn/”)

x-crawl 的特征

* ** AI 辅助** - 强大的 AI 辅助功能，使爬虫工作变得更加高效、智能和便捷。
* ** 写法灵活** - 单个爬取 API 都适配多种配置，每种配置方式都各有千秋。
* ** 多种用途** - 支持爬动态页面、静态页面、接口数据以及文件数据。
* ** 控制页面** - 爬取动态页面支持自动化操作、键盘输入、事件操作等。
* ** 设备指纹** - 零配置或自定义配置，避免指纹识别从不同位置识别并跟踪我们。
* ** 异步同步** - 无需切换爬取 API 即可进行异步或同步的爬取模式。
* ** 间隔爬取** - 无间隔、固定间隔以及随机间隔，决定是否高并发爬取。
* ** 失败重试** - 自定义重试次数，避免因短暂的问题而造成爬取失败。
* ** 轮换代理** - 搭配失败重试，自定义错误次数以及 HTTP 状态码自动轮换代理。
* ** 优先队列** - 根据单个爬取目标的优先级可以优先于其他目标提前爬取。
* ** 爬取信息** - 可控的爬取信息，会在终端输出彩色字符串信息。
* ** TypeScript** - 拥有类型，通过泛型实现完整的类型。

AI 和 x-crawl 爬虫结合示例

AI 和 x-crawl 结合，让爬虫和 AI 获取高评分度假屋的房屋图片：

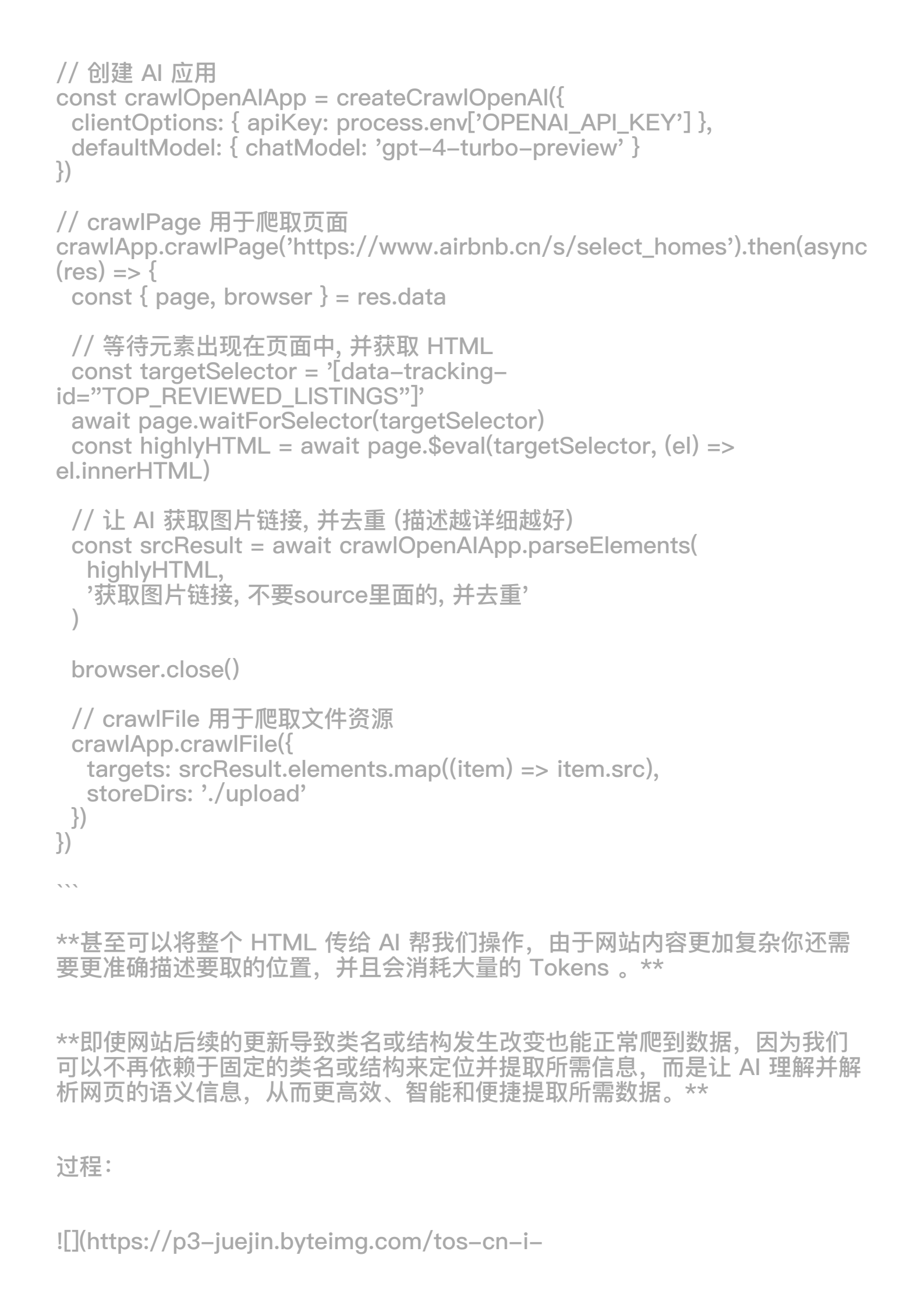
```
import { createCrawl, createCrawlOpenAI } from 'x-crawl'

// 创建爬虫应用
const crawlApp = createCrawl({
  maxRetry: 3,
  intervalTime: { max: 2000, min: 1000 }
})
```

```
// 创建 AI 应用
const crawlOpenAIApp = createCrawlOpenAI({
  clientOptions: { apiKey: process.env['OPENAI_API_KEY'] },
  defaultModel: { chatModel: 'gpt-4-turbo-preview' }
})

// crawlPage 用于爬取页面
crawlApp.crawlPage('https://www.airbnb.cn/s/select_homes').then(async
(res) => {
  const { page, browser } = res.data

  // 等待元素出现在页面中, 并获取 HTML
  const targetSelector = '[data-tracking-
id="TOP_REVIEWED_LISTINGS"]'
  await page.waitForSelector(targetSelector)
  const highlyHTML = await page.$eval(targetSelector, (el) =>
el.innerHTML)

  // 让 AI 获取图片链接, 并去重 (描述越详细越好)
  const srcResult = await crawlOpenAIApp.parseElements(
    highlyHTML,
    '获取图片链接, 不要source里面的, 并去重'
  )

  browser.close()

  // crawlFile 用于爬取文件资源
  crawlApp.crawlFile({
    targets: srcResult.elements.map((item) => item.src),
    storeDirs: './upload'
  })
})

```

甚至可以将整个 HTML 传给 AI 帮我们操作，由于网站内容更加复杂你还需要更准确描述要取的位置，并且会消耗大量的 Tokens 。

即使网站后续的更新导致类名或结构发生改变也能正常爬到数据，因为我们可以不再依赖于固定的类名或结构来定位并提取所需信息，而是让 AI 理解并解析网页的语义信息，从而更高效、智能和便捷提取所需数据。

过程：



k3u1fbpfcp/8f29cbd020bb4492a33afee1c4e39989~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1626&h=565&s=4869908&e=gif&f=127&b=23262e)

如果你想查看 `AI 需要处理的 HTML` 或 `AI 返回的 srcResult（房屋图片链接）`:
[由于内容太多此处放不下，就只能放在此链接 示例1 底部](http://cxyroad.com/ "https://coder-hxl.github.io/x-crawl/cn/guide/#example1")

AI 智能按需分析元素

无需手动分析 HTML 页面结构再提取所需的元素属性或值。现在只需将 HTML 代码输入到 AI 中，并告知 AI 您想获取哪些元素的信息，AI便会自动分析页面结构，提取出相应的元素属性或值。

```
import { createCrawlOpenAI } from 'x-crawl'

const crawlOpenAIApp = createCrawlOpenAI({
  clientOptions: { apiKey: '你的 API Key' }
})

const HTMLContent = `
  <div class="scroll-list">
    <div class="list-item">女装带帽卫衣</div>
    <div class="list-item">男装卫衣</div>
    <div class="list-item">女装卫衣</div>
    <div class="list-item">男装带帽卫衣</div>
  </div>
  <div class="scroll-list">
    <div class="list-item">男装纯棉短袖</div>
    <div class="list-item">男装纯棉短袖</div>
    <div class="list-item">女装纯棉短袖</div>
    <div class="list-item">男装冰丝短袖</div>
    <div class="list-item">男装圆领短袖</div>
  </div>
`

crawlOpenAIApp.parseElements(HTMLContent, '获取男装, 并去重').then((res) => {
  console.log(res)
  /*
```

```
    res:
    {
      elements: [
        { class: 'list-item', text: '男装卫衣' },
        { class: 'list-item', text: '男装带帽卫衣' },
        { class: 'list-item', text: '男装纯棉短袖' },
        { class: 'list-item', text: '男装冰丝短袖' },
        { class: 'list-item', text: '男装圆领短袖' }
      ],
      type: 'multiple'
    }
  */
})

```

甚至可以将整个 HTML 传给 AI 帮我们操作，由于网站内容更加复杂你还需要更准确描述要取的位置，并且会消耗大量的 Tokens 。

AI 智能生成元素选择器

能够帮助我们快速定位到页面中的特定元素。只需将 HTML 代码输入到 AI 中，并告知 AI 您想获取哪些元素的选择器，AI 便会根据页面结构自动为您生成合适的选择器，大大简化了确定选择器的繁琐过程。

示例：

```
import { createCrawlOpenAI } from 'x-crawl'

const crawlOpenAIApp = createCrawlOpenAI({
  clientOptions: { apiKey: '你的 API Key' }
})

const HTMLContent = `
  <div class="scroll-list">
    <div class="list-item">女装带帽卫衣</div>
    <div class="list-item">男装卫衣</div>
    <div class="list-item">女装卫衣</div>
    <div class="list-item">男装带帽卫衣</div>
  </div>
  <div class="scroll-list">
    <div class="list-item">男装纯棉短袖</div>
```

```
    <div class="list-item">男装纯棉短袖</div>
    <div class="list-item">女装纯棉短袖</div>
    <div class="list-item">男装冰丝短袖</div>
    <div class="list-item">男装圆领短袖</div>
  </div>
`

crawlOpenAIApp.getElementSelectors(HTMLContent, '获取所有女装
').then((res) => {
  console.log(res)
  /*
    res:
    {
      selectors: '.scroll-list:nth-child(1) .list-item:nth-of-type(1), .scroll-
list:nth-child(1) .list-item:nth-of-type(3), .scroll-list:nth-child(2) .list-
item:nth-of-type(3)',
      type: 'single'
    }
  */
})

```

甚至可以将整个 HTML 传给 AI 帮我们操作，由于网站内容更加复杂你还需要更准确描述要取的位置，并且会消耗大量的 Tokens 。

AI 智能回复爬虫问题

可以为您提供智能的解答和建议。无论是关于爬虫策略、反爬虫技巧还是数据处理等方面的问题，您都可以向AI提问，AI会根据其强大的学习和推理能力，为您提供专业的解答和建议，帮助您更好地完成爬虫任务。

```
import { createCrawlOpenAI } from 'x-crawl'

const crawlOpenAIApp = createCrawlOpenAI({
  clientOptions: { apiKey: '你的 API Key' }
})

crawlOpenAIApp.help('x-crawl 是什么').then((res) => {
  console.log(res)
  /*
    res:
    x-crawl 是一个灵活的 Node.js AI 辅助爬虫库，它提供了强大的人工智能辅
```

助功能，可以帮助开发者更高效、智能和便捷地进行网络爬虫工作。您可以在 GitHub 上找到更多关于 x-crawl 的详细信息和使用方式：https://github.com/coder-hxl/x-crawl。
*/
})

crawlOpenAIApp.help('爬虫的三大注意事项').then((res) => {
console.log(res)
/*
res:
在进行爬虫工作时，有三个重要的注意事项需要特别注意：

1. **遵守网站规则和法律法规**：在进行数据爬取时，一定要遵守网站的robots.txt文件中的规则，并且不要违反任何相关的法律法规。尊重网站所有者的意愿和数据的所有权是非常重要的。

2. **避免对网站造成过大负担**：爬虫在爬取数据时会占用网站的带宽和资源，过度频繁的访问会给网站带来压力甚至是瘫痪。因此，需要合理设置爬虫的访问频率，并且避免对网站造成过大的访问负担。

3. **数据处理和存储的合法性和隐私保护**：爬取到的数据可能涉及用户的隐私信息，因此在收集、存储和使用这些数据时，要符合相关的隐私保护法律法规，并且不要滥用这些数据。另外，在处理数据时也要保证数据的准确性和可靠性，避免因不当的数据处理而产生误解或造成不良影响。
*/
})

```

总结
--

**1. 智能按需分析元素**

传统的爬虫工作往往需要手动分析 HTML 页面结构，提取所需的元素属性或值。而现在，借助 x-crawl 的 AI 辅助，您可以轻松实现智能按需分析元素。只需告诉AI您想获取哪些元素的信息，AI 便会自动分析页面结构，提取出相应的元素属性或值。

**2. 智能生成元素选择器**

选择器是爬虫工作中不可或缺的一部分，它能够帮助我们快速定位到页面中的特定元素。现在，x-crawl 的 AI 辅助可以为您智能生成元素选择器。只需将 HTML 代码输入到 AI 中，AI 便会根据页面结构自动为您生成合适的选择器
```

，大大简化了确定选择器的繁琐过程。

3. 智能回复爬虫问题

在爬虫工作中，我们难免会遇到各种问题和挑战。而 x-crawl 的 AI 辅助可以为您提供智能的解答和建议。无论是关于爬虫策略、反爬虫技巧还是数据处理等方面的问题，您都可以向AI提问，AI会根据其强大的学习和推理能力，为您提供专业的解答和建议，帮助您更好地完成爬虫任务。

综上所述，结合 AI 技术的爬虫能够更好地应对网站更新后类名或结构改变的问题。

---

x-crawl GitHub: [github.com/coder-hxl/x...](http://cxyroad.com/ "https://github.com/coder-hxl/x-crawl")

x-crawl 文档: [coder-hxl.github.io/x-crawl/cn/](http://cxyroad.com/ "https://coder-hxl.github.io/x-crawl/cn/")

> 如果您觉得 x-crawl 对您有所帮助，或者您喜欢 x-crawl ，可以在 GitHub 上给 [x-crawl 存储库](http://cxyroad.com/ "https://github.com/coder-hxl/x-crawl") 点个 star 。您的支持是我们持续改进的动力！感谢您的支持！

原文链接: https://juejin.cn/post/7355798869110865939