



链滴

# 【分享】html 内容块 + quicker 实现对嵌入块分页查询

作者: [kobi](#)

原文链接: <https://ld246.com/article/1652029173155>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

因为嵌入块一般特别长，想分页进行浏览就不用一次性看完了。

结合思源的html块，块属性实现对嵌入块的分页查询。代码很烂，不喜轻轻喷。

### 用法：

1. 复制html代码到html块中，并修改custom\_id为唯一标识。建议这个id使用html块id。（html代去文末复制）
2. 配置html块的属性。
  1. qrkid 表示 嵌入块id
  2. quickeruri指 这个动作的uri地址。 [quicker动作](#)
  3. sql：结尾不要用分号，不要用limit，不要用|| 拼接字符串（嵌入块好像不支持）。
3. 在quicker中配置siyuan token。

### 注意：目前还不完善。

1. 建议同时只打开一个使用html块对嵌入块进行分页查询的文档。（js写得很烂，打开多了可能dom获取）
2. 要html块里面的custom\_id每个html中要使用不一样的。（推荐使用html块id，这样就都不同了）
3. 查询的sql请注意，不要查询到自身或者同样使用html块进行分页查询的文档，以防套娃，导致程崩溃。



[演示.mp4](#)

## html代码如下:

```
<html>
<head>
  <style>
    #page_nav{width:100%;height:50px;display: flex; justify-content: center;}
    .page_btn{width:60px;display: inline-block; text-align: center;height: 50px;line-height: 50
x;}
    #page_nav a{color: #8ab4f8; cursor: pointer;}
    #pageNum{width:50px;}
  </style>
</head>
<body>
  <div id="page_nav">
    <form onload="formLoad()" readonly> </form>
    <a id="first" class="page_btn" href="javascript:first()">首页</a>
    <a id="pre" class="page_btn" href="javascript:pre()">上一页</a>
    <input id="pageNum" name="pageNum" value="0" readonly placeholder="第几页"/>
    <a id="next" class="page_btn" href="javascript:next()">下一页</a>
    <p>查询条数
      <select id="querycount" onchange="changeCount()">
        <option value="1">1</option>
        <option value="2">2</option>
        <option value="3">3</option>
        <option value="4">4</option>
        <option value="5" selected="selected">5</option>
        <option value="6">6</option>
        <option value="7">7</option>
        <option value="8">8</option>
        <option value="9">9</option>
        <option value="10">10</option>
        <option value="20">20</option>
        <option value="25">25</option>
        <option value="30">30</option>
        <option value="50">50</option>
      </select>
    </p>
  </div>
</body>
<script>
{
```

```
  const customID = '这个id为用户自己设置，用来唯一标识这个块的，建议使用html块id';//如: 20
20507204221-l9rthtz
  let Me = This(customID);
  console.log(Me);
```

```
// 初始化，自动加载 pageNum 和pageCount
init();
```

```
function init(){
  var embeddedBlockId = Me.block.getAttribute('custom-qrkid');
```

```

    if(!embeddedBlockId){
        alert("需要在HTML块的属性中配置：qrkid！ ")
        return;
    }

    var pageNum = document.querySelector(`div[data-node-id="${embeddedBlockId}"].getAttribute('custom-pagenum');
    var querycount = document.querySelector(`div[data-node-id="${embeddedBlockId}"].getAttribute('custom-querycount');
    if(!pageNum || !querycount){
        return;
    }
    Me.shadowRoot.querySelector("#pageNum").value = parseInt(pageNum);
    Me.shadowRoot.querySelector("#querycount").value = parseInt(querycount);
}

/**
 * HTML 块中的脚本获取当前块相关信息
 * @params {string} customID 内部定义的 ID
 * @returns {string} id 当前 HTML 块 ID
 * @returns {HTMLElement} block 当前 HTML 块
 * @returns {HTMLElement} shadowRoot 当前 HTML 块 shadowRoot
 */
function This(customID) {
    let protyle = document.querySelector(`protyle-html[data-content*="${customID}"]`);
    if (protyle) {
        let block = protyle.parentElement.parentElement;
        return {
            id: block.dataset.nodetid,
            block: block,
            shadowRoot: protyle.shadowRoot,
        };
    } else return null;
}

function changeCount(){
    first()
}
function first(){
    Me.shadowRoot.querySelector("#pageNum").value = 0;
    refreshEmbeddedBlock();
}
function pre(){
    var pageNum = parseInt(Me.shadowRoot.querySelector("#pageNum").value || 0);
    if(pageNum <= 0){
        return;
    }
    Me.shadowRoot.querySelector("#pageNum").value = pageNum - 1;
    refreshEmbeddedBlock();
}
function next(){
    var pageNum = parseInt(Me.shadowRoot.querySelector("#pageNum").value || 0);
    Me.shadowRoot.querySelector("#pageNum").value = pageNum + 1;
    refreshEmbeddedBlock();
}

```

```

}

function refreshEmbeddedBlock() {
  var embeddedBlockId = Me.block.getAttribute('custom-qrkid');
  var sql = Me.block.getAttribute('custom-sql');
  var quickeruri = Me.block.getAttribute('custom-quickeruri');

  if(!embeddedBlockId || !sql || !quickeruri){
    console.log("embeddedBlockId:" + embeddedBlockId)
    console.log("sql:" + sql)
    console.log("quickeruri:" + quickeruri)

    alert("需要在HTML块的属性中配置：qrkid , sql , quickeruri 这几个属性！！")
  }

  var pageNum = parseInt(Me.shadowRoot.querySelector("#pageNum").value || 0);
  var querycount = parseInt(Me.shadowRoot.querySelector("#querycount").value || 0);
  if(querycount <= 0){
    alert("请重新设置查询条数！")
  }

  sql = `${sql} limit ${pageNum*querycount} , ${querycount}`
  // alert(`embeddedBlockId:${embeddedBlockId}\n pageNum:${pageNum}\n querycount:
  {querycount}\n `)
  var jsonStr = JSON.stringify({qrkid:embeddedBlockId,sql:sql,pagenum:pageNum,querycount:querycount});
  // console.log(jsonStr)
  // var full_quickeruri = `${quickeruri}?qrkid=${embeddedBlockId}&sql=${sql}`;
  var full_quickeruri = `${quickeruri}?${jsonStr}`;
  console.log(full_quickeruri);
  location.href =full_quickeruri;
}
}

</script>
</html>

```