

# SnapRead 接口文档

Author: AlanDelip

## 0. 标准 Response 结构

```
{
  condition: "success | fail" //返回是否成功
  error_code: "00001" //失败编号
  msg: "" //系统级失败原因, 成功就不返回
  data:{} //数据
}
```

## 1. 获取用户收藏的网络端文章列表

URL	/api/getPostList
Parameters	user_id 用户 ID
Response	[ { post_id: "01" , title: "Material Design" , description: "Material design is popular..." //文章内容概要 post_url: "http://www.baidu.com" //文章原始网页 URL post_img: "http://our_server/img1.png" //文章图片(如果有) type: "设计" //文章类型 } ]
Memo	文章内容概要如果用户没有定义的话就是前 50 词

## 2. 获取文章内容

URL	/api/getPost
Parameters	post_id 文章 ID
Response	{ title: "Material Design" , content: "<div>Material Design content...</div>" //文章 Html 内容 post_url: "http://www.baidu.com" //文章原始网页 URL post_img: "http://our_server/img1.png" //文章图片(如果有) type: "设计" //文章类型 }
Memo	没有用户模块的话不能跨设备

### 3. 搜索识别文章

URL	/api/searchPost
Parameters	<b>keyword</b> 关键字
Response	{ <b>post_id</b> : "01" <b>title</b> : "Material Design" , <b>content</b> : "<div>Material Design content...</div>" //文章 Html 内容 <b>description</b> : "material design..." //文章简介 <b>post_url</b> : "http://www.baidu.com" //文章原始网页 URL <b>post_img</b> : "http://our_server/img1.png" //文章图片(如果有) <b>type</b> : "设计" //文章类型 }
Memo	1. type 一开始是通过 NLP 自己识别的, 用户还可以修改

### 4. 新增文章 POST

URL	/api/editPost
Parameters	<b>user_id</b> 用户 ID <b>post_id</b> 文章 ID <b>title</b> 文章标题, <b>content</b> 文章 Html 内容初期不能修改的, <b>description</b> 文章简介 <b>post_url</b> 文章原始网页 URL, <b>img_url</b> 图片 url, <b>type</b> 文章类型
Response	
Memo	

### 5. 删除文章

URL	/api/deletePost
Parameters	<b>[post_id]</b> 文章 ID 数组
Response	
Memo	一次可以删除多个文章

### 6. 获取用户的标签列表

URL	/api/getTag
Parameters	<b>user_id</b> 用户 ID
Response	[

	<pre>{   tag_id: "01" ,   tag_name: "design" ,   description: "about design." ,   tag_img: "http://our_server/img2.png" }</pre>
<b>Memo</b>	

## 7. 新建标签 POST

<b>URL</b>	/api/addTag
<b>Parameters</b>	<b>user_id</b> 用户 ID <b>tag_name</b> 标签名 <b>description</b> 标签简介 <b>tag_img</b> 图片 url
<b>Response</b>	
<b>Memo</b>	

## 8. 删除标签

<b>URL</b>	/api/deleteTag
<b>Parameters</b>	<b>tag_id</b> 标签 ID
<b>Response</b>	
<b>Memo</b>	删除 tag 也会删除标签下的文章

## 9. 获取文章推荐内容

<b>URL</b>	/api/getPostRecommend
<b>Parameters</b>	<b>post_id</b> 文章 ID
<b>Response</b>	<pre>[   {     post_id: "01" ,     title: "Java Lesson" ,     description: "Java is a language..." //文章内容概要     post_url: "http://www.baidu.com" //文章原始网页 URL     post_img: "http://our_server/img2.png" //文章图片(如果有)     type: "课程" //文章类型     reason: "经常看 IT 的文章" //推荐原因   } ]</pre>

<b>Memo</b>	<ol style="list-style-type: none"> <li>1. 这里推荐的可以是正常相关的文章，也可能是广告宣传，不过 id 要加以区分开（见附录）</li> <li>2. 推荐原因这里需要推敲</li> <li>3. 一般 1~2 篇</li> </ol>
-------------	---

## 10. 获取用户推荐内容

<b>URL</b>	/api/getUserRecommend
<b>Parameters</b>	<b>user_id</b> 用户 ID
<b>Response</b>	<pre>[   {     post_id: "01" ,     title: "Java Lesson" ,     description: "Java is a language..." //文章内容概要     post_url: "http://www.baidu.com" //文章原始网页 URL     post_img: "http://our_server/img2.png" //文章图片(如果有)     type: "课程" //文章类型     reason: "经常看 IT 的文章" //推荐原因   } ]</pre>
<b>Memo</b>	<ol style="list-style-type: none"> <li>4. 这里推荐的可以是正常相关的文章，也可能是广告宣传，不过 id 要加以区分开（见附录）</li> <li>5. 推荐原因这里需要推敲</li> </ol>

## 11. 过滤文章(等会儿做，我在做本地数据库)

<b>URL</b>	/api/filterPost
<b>Parameters</b>	<b>user_id</b> 用户 ID, <b>keyword</b> 关键字, <b>type</b> 类型(keyword   tag)
<b>Response</b>	<pre>[   {     post_id: "01" ,     title: "Material Design" ,     description: "Material design is popular..." //文章内容概要     post_url: "http://www.baidu.com" //文章原始网页 URL     post_img: "http://our_server/img1.png" //文章图片(如果有)     type: "设计" //文章类型   } ]</pre>
<b>Memo</b>	<ol style="list-style-type: none"> <li>1. 实际上是搜索，因为不建立本地数据库，用户可能一次未加载完所有的文章，所以搜索就不方便了。</li> <li>2. Type 参数是类型，可能是类别或是关键字，是关键字的时候需要对标题、标签和内容进行搜索</li> </ol>

## 附录

### 1. Post\_id 标准

普通类型文章：n00001, n00002

推广类文章: a00001, a00002

### 2. Error\_code 标准