

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 | keyword 关键字 |
| Response | { post_id : "01" 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 | 1. type 一开始是通过 NLP 自己识别的, 用户还可以修改 |

4. 编辑文章

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

5. 删除文章

| | |
|------------|---------------------------|
| URL | /api/deletePost |
| Parameters | [post_id] 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 | user_id 用户 ID |
| Response | [{ tag_id : "01" , } |

| | |
|-------------|--|
| | <pre> tag_name : "design" , description : "about design." , tag_img : "http://our_server/img2.png" }]</pre> |
| Memo | |

7. 新建标签

| | |
|-------------------|---|
| URL | /api/addTag |
| Parameters | user_id 用户 ID tag_name 标签名 description 标签简介 tag_img 图片 url |
| Response | |
| Memo | |

8. 删除标签

| | |
|-------------------|---------------------|
| URL | /api/deleteTag |
| Parameters | tag_id 标签 ID |
| Response | |
| Memo | 删除 tag 也会删除标签下的文章 |

9. 获取文章推荐内容

| | |
|-------------------|---|
| URL | /api/getPostRecommend |
| Parameters | post_id 文章 ID |
| Response | <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> |
| Memo | 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） |

| | |
|--|------------------------------|
| | 2. 推荐原因这里需要推敲 3. 一般 1~2 篇 |
|--|------------------------------|

10. 获取用户推荐内容

| | |
|-------------------|--|
| URL | /api/getUserRecommend |
| Parameters | user_id 用户 ID |
| Response | <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> |
| Memo | 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） 5. 推荐原因这里需要推敲 |

11. 过滤文章(等会儿做，我在做本地数据库)

| | |
|-------------------|--|
| URL | /api/filterPost |
| Parameters | user_id 用户 ID, keyword 关键字, type 类型(keyword tag) |
| Response | <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> |
| Memo | 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。 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 |

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