**SnapRead接口文档**

Author: AlanDelip

1. **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, page 请求第几页 |
| 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**  **total**:5 **//总页数**  **current\_page**: 1 **//当前页**  }  ] |
| Memo | 文章内容概要如果用户没有定义的话就是前50词 |

1. **获取文章内容**

|  |  |
| --- | --- |
| URL | /api/getPost |
| Parameters | post\_id 文章ID |
| Response | {  **title**: “Material Design”,  **content**: “Material Design content…”**//文章内容**  **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. 这里考虑如果文章中需要图片作为主要内容怎么办，我们的文章图片一般是选择文章的第一个图片，而且不能记录位置 2. 一般先去本地数据库拿数据，不会去获取网络端文章内容，但是考虑到如果用户换设备或是查看的是推荐文章的话，本地数据库就没数据了 |

1. **搜索识别文章**

|  |  |
| --- | --- |
| URL | /api/searchPost |
| Parameters | keyword 关键字 |
| Response | {  **post\_id**: “01”  **title**: “Material Design”,  **content**: “Material Design content…”**//文章内容**  **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 2. 服务器默认是识别的时候就保存文章，如果用户点击取消就会发送删除文章的请求 |

1. **编辑文章**

|  |  |
| --- | --- |
| URL | /api/editPost |
| Parameters | **post\_id** 文章ID  **title** 文章标题，  **content** 文章内容，  **post\_url** 文章原始网页URL，  **post\_img** 图片（限制1张），  **type** 文章类型 |
| Response |  |
| Memo | POST请求，图片流和文本混合 |

1. **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 |

1. **获取推荐内容**

|  |  |
| --- | --- |
| URL | /api/getRecommend |
| Parameters | **user\_id**用户ID |
| Response | [  {  **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**  }  ] |
| 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 |

1. **·过滤文章**

|  |  |
| --- | --- |
| URL | /api/filterPost |
| Parameters | user\_id 用户ID, keyword 关键字, type 类型(keyword | tag) |
| 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 | 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

1. **Error\_code标准**