**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 |
| 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 |

1. **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 |

1. **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**  **description :** “material design…” **//文章简介**  **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 |

1. **新增文章 POST**

|  |  |
| --- | --- |
| 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，  **description** 文章简介  **post\_url** 文章原始网页URL，  **img\_url** 图片url，  **type** 文章类型 |
| Response |  |
| Memo |  |

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. **获取用户的标签列表**

|  |  |
| --- | --- |
| URL | /api/getTag |
| Parameters | **user\_id** 用户ID |
| Response | [  {  tag\_id : “01”,  tag\_name : “design”,  description : “about design.”,  tag\_img : “http://our\_server/img2.png”  }  ] |
| Memo |  |

1. **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 |  |

1. **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 |

1. **获取文章推荐内容**

|  |  |
| --- | --- |
| URL | /api/getPostRecommend |
| Parameters | **post\_id** 文章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 3. 一般1~2篇 |

1. **获取用户推荐内容**

|  |  |
| --- | --- |
| URL | /api/getUserRecommend |
| 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(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 | [  {  **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**