

CoolReader 前期开发进度及协议格式

- 开发方式：快速迭代
- 开发语言：
 - 服务器端：C/C++/golang
 - Web端：HTML/CSS/JS/PHP
 - Android端：Java
- 开发进度：

完成的内容	描述	时间
协议和规定的制定	指定远程请求的详细内容及使用的协议，方便后期各端开发	7.18-7.20

Version 0.1：

终端	需要完成的内容	时间	人员
安卓端	1.与服务器交互2.显示列表3.离线本地	7.21-7.26	2 人
Web端	1.主界面显示2.与服务器交互3.显示用户资源	7.21-7.26	1 人
服务端	1.协议解析2.返回数据给两个终端	7.21-7.23	4 人

Version 0.2

终端	需要完成的内容	时间	人员
安卓端	1.允许用户编辑标签2.优化显示3.加快显示	7.27-7.29	2 人
Web端	1.允许用户编辑标签，包括增删改2.本地查找	7.27-7.29	1 人
服务端	1.加速数据库I/O 2.日志异常功能3.仿照Libevent的Reactor模型	7.24-7.29	4 人

Version 1.0

对0.2版本进行整合，减少出错异常情况，减少耦合并记录可能会存在的问题及可优化的地方，在2.0版本中改进，在此基础上，添加新的功能，如推荐和分享功能。

- 远程请求格式：JSON
- 请求类型：
 - 1.登录验证
 - 2.注册申请
 - 3.同步数据请求（起始状态，传输，终止状态）
 - 4.删除请求
 - 5.添加请求
 - 6.查询请求
 - 7.出错情况

声明

- 这里只是处理请求的 json 数据格式
- edited by @luoyuping
- ok ,lets start ~

当前协议的制定均以方便简洁为首要考虑因素

正文

总体的格式

```
{
  "funType":Number,
  "funContent":
    {
      ....
    }
}
```

登陆模块

```
{
  "funType":10,
  "funContent":
    {
      "userInfo":
        {
          "mailAddr":String,
          "password":String
        }
    }
}
```

注册模块

```
{
  "funType":20 ,
  "funContent":
    {
      "userInfo":
        {
          "userName":String,
          "password":String,
          "mailAddr":String,
          "Sex":String
        }
    }
}
```

同步数据

- 同步服务器数据库 (发送更新请求)

web

```
{
  "funType":30,
  "funContent":
    {
      "mailAddr":string,
      "deviceType": "browser"           // PC 来的请求
    }
}
```

Android

```
{
  "funType":30,
```

```

"funContent":
{
    "mailAddr":string,
    "deviceType": "android"           // android 来的请求
    "url":String
}
}

```

____此处的url为Android已经标记为删除的URL，多个URL之间连接用&&&&

添加数据到

- Web或者 android 端发送 url到服务器

web

```

{
    "funType":50,
    "funContent":
    {
        "mailAddr":string,
        "deviceType": "brower",           // PC 来的请求
        "url":String,
        "tag":String           // tag 之间用#分割，手动提取
    }
}

```

添加数据分不同的平台是因为对添加的处理不同.对于web端仅需要将title返回给它就行,对于android 则需要将网页过滤后的全部信息给它.

注意这里的tag表示的是用户对当前链接添加的标签信息.这个字段在查找时会用到.

android

```

{
    "funType":50,
    "funContent":
    {
        "mailAddr":string,
        "deviceType": "android",           // android 来的请求
        "url":String,
        "tag":String           // tag 之间用#分割，手动提取
    }
}

```

删除数据

```

{
    "funType":40,
    "funContent":
    {
        "mailAddr":string,
        "deviceType": "android", // 标识是哪种设备提交的删除请求
        "url": "", // 可以同时删除多个URL
    }
}

```

注意这里如果对应的device值为android，那么可以直接删除数据库的内容.如果device的值为web,那么仅仅修改数据库中当前用户这条链接的状态为待同步状态

注意这里的device为不同的设备，由于考虑到PC端删除的处理操作与android处理不同，urlCount字段表示当前要批量删除的URL总数，用来给服务端解析Json时使用。

查询数据

```
{
  "funType":60,
  "funContent":
  {
    "mailAddr":string,
    "keywords":string           //获取查询的关键字
  }
}
```

注意这个请求是Web端使用的

关于请求返回数据格式

• 登录返回

OK 没有错误

```
{
  "funType":200,
  "describe":"OK"
}
```

USER_NOT_EXIST 不存在

```
{
  "funType":201,
  "describe":"USER_NOT_EXIST"
}
```

SERVER_ERROR 服务器错误

```
{
  "funType":202,
  "describe":"SERVER_ERROR"
}
```

WRONG_PASSWD 密码错误

```
{
  "funType":203,
  "describe":"WRONG_PASSWD"
}
```

• 注册返回

OK 注册成功

```
{
  "funType":200 ,
  "describe":"OK"
}
```

ERROR 服务器错误

```
{
  "funType":202,
  "describe":"SERVER_ERROR"
}
```

用户已经存在

```
{
  "funType":206,
  "describe":"USER_ALREADY_EXIST"
}
```

• 同步数据返回

OK_READY 准备开始同步数据

```
{
  "return1":
  {
    "funType":36 ,
    "content":{"url":""}
  }
  "return2":
  {
    "funType":200 ,
    "describe":"OK"
  }
  "return3":
  {
    "funType":37,
    "content":{"url":""," "title":""," "info":"","urlCount":5}
  }
}
```

对于Web端,info字段为空,Web端只需要URL和title

这里的url,title,info的内容均使用拼接的方式,内容使用&&&&来连接

这里的urlCount表示当前传送的内容的总数

OK_FINISH 结束传输

```
{
  "funType":38,
  "describe":"OK_FINISH"
}
```

• 删除数据返回

OK 没有错误

```
{
  "funType":200,
  "describe":"OK"
}
```

SERVER_ERROR 服务器错误

```
{
  "funType":202,
  "describe":"SERVER_ERROR"
}
```

• 添加数据返回

OK 没有错误

```
{
  "funType":200,
  "describe":"OK"
}
```

URL_ALREADY_EXIST 链接已经存在

```
{
  "funType":205,
  "describe":"URL_ALREADY_EXIST"
}
```

SERVER_ERROR 服务器错误

```
{
  "funType":202,
  "describe":"SERVER_ERROR"
}
```

• 查询数据返回(Web端查询时使用)

```
{
  "funType":61,
  "content":{"url":"","title":"","urlCount":5}
}
```

这里的urlCount表示返回内容总数

内容用&&&&进行拼接

URL_NOT_EXIST 查询的值不存在

```
{
  "funcType":204,
  "describe":"URL_NOT_EXIST"
}
```

SERVER_ERROR 服务器错误

```
{
  "funType":202,
  "describe":"SERVER_ERROR"
}
```

在此集中列举状态值代表的含义:

- 200 OK 没有错误
- 201 USER_NOT_EXIST 用户不存在
- 202 SERVER_ERROR 服务器错误
- 203 WRONG_PASSWD 密码错误
- 204 URL_NOT_EXIST 查询的值不存在
- 205 URL_ALREADY_EXIST 链接已经存在

####206 USER_ALREADY_EXIST 用户已存在