

ScutGame 网络库 Flash 版本 帮助文档

- 版本：V1.0
- 作者：ScutGame
- 日期：2013.12.27

- 使用方法：

在 as3 工程 或 Air 工程里导入 libs 文件里的 **ScutDataLogic.swc** 和 **ScutNetwork.swc**

- 网络库 API：

As3 版本的网络库完全与 ScutGame SDK 的接口一致。只要转换成相应的 as3 写法即可。

- Sample 工程解析
 - 该 sample 工程是用 FlashAir 搭建。
 - 里面通讯代码适用于其他任何 as3 工程。
 - 源代码可参考 sample/ ScutGameNetDemo

工程解析：

1. Lib 库的初始化

```
// 游戏初始化 // 固定添加即可。  
// 1. 注册事件派发机制  
// 2. 初始化FrameManager  
GameInit();  
  
private function GameInit():void  
{  
    //注册派发事件  
    ScutDataLogic.CDataRequest.Instance().RegisterLUACallBack(ScutDataLogic.DataDispatch.netDataDispatch);  
    //强制调用框架管理器，以执行初始化操作  
    FrameManager.Instance().init(this.stage);  
  
    // 设置服务器URL。 只需设置一次。如有更改，重新设置即可。  
    ScutDataLogic.CNetWriter.setUrl("http://ph.scutgame.com/service.aspx");  
}
```

在游戏启动时，统一加上这个初始化函数。主要用来注册事件的派发。还有设置所有协议请求的服务器地址。

2. 注册回调函数

```
// 注册侦听事件 ( 即通讯成功回调函数 )  
registerCallback(EVENT_GET_RANK,onGetRankHandler);
```



```

//请求排行榜列表事件
public static var EVENT_GET_RANK:String = "Event_Get_Rank";
//提交分数事件
public static var EVENT_SUBMIT_RANK:String = "Event_Submit_Rank";

```

所有的游戏协议事件都可以先声明好一个唯一的名字。用来请求和回调侦听。

3. 发送请求

```

private function sendRequest():void
{
    //写入协议字段
    ZyWriter.writeString("ActionId","1001");
    ZyWriter.writeString("PageIndex","1");
    ZyWriter.writeString("PageSize","30");

    // 游戏发起请求
    //parm: String 该协议名称 ( 可自定义 )
    ScutDataLogic.ZyExecRequest.request(EVENT_GET_RANK);
}

```

4. 回调函数处理

```

protected function onGetRankHandler(event:Event):void
{
    var ZyReader:CNetReader = CNetReader.getInstance();
    if(ZyReader.getResult() == 0){
        var pageCount:int = ZyReader.getInt();
        var RecordNums_1:int = ZyReader.getInt();
        var dataAry:Array = [];
        if(RecordNums_1 != 0){
            for(var k:int=0;k<RecordNums_1;k++){
                var mRecordTabel_1:Object = {};
                ZyReader.recordBegin();
                mRecordTabel_1.UserName= ZyReader.readString();
                mRecordTabel_1.Score= ZyReader.getInt();
                ZyReader.recordEnd();
                dataAry.push(mRecordTabel_1);
            }
        }
    }
}

```
