

消息中心（NotificationCenter）

（MxFramework4.2）

目录

目录	2
一、 介绍.....	3
二、 API.....	3
1. 注册消息监听事件	3
2. 注销消息监听事件	3
3. 注销所有消息监听事件	3
4. 发送消息.....	4
三、 如何使用	4
1. 发送消息.....	4
2. 接收消息.....	4
四、 源码结构	5
1. 核心代码路径.....	5
2. 示例工程路径.....	5
五、 源码下载	6

一、 介绍

`MessageCenter` 是基于委托和事件（观察者模式）而设计的，是 `MXFramework` 框架 的一个子模块。 主要是为了处理消息的收发。

二、 API

1. 注册消息监听事件

```
public static void AddMsgListener(string messageType,  
Action<string, object> handler)
```

2. 注销消息监听事件

```
public static void RemoveMsgListener(string messageType,  
Action<string, object> handler)
```

3. 注销所有消息监听事件

```
public static void ClearAllMsgListener()
```

4. 发送消息

```
public static void SendMessage(string messageType, string key,
object values)
```

三、 如何使用

1. 发送消息

```
MessageMgr.SendMessage("消息分组", "消息名称", 消息体);
```

2. 接收消息

```
private void Awake()
{
    //注册事件>>>>>>>>>>>>>>>>>>>>>>
    MessageMgr.AddMsgListener("消息分组", OnTestMessageCenterEvent);
}

private void OnDestroy()
{
    //记得有注册就会有注销>>>>>>>>>>>>>>>>>>>>>>
    MessageMgr.RemoveMsgListener("消息分组", OnTestMessageCenterEvent);
}
```

```

/// <summary>
/// 监听测试消息中心数据
/// </summary>
/// <param name="key">数据名称</param>
/// <param name="values">具体的数值</param>
private void OnTestMessageCenterEvent(string key, object values)
{
    switch (key)
    {
        case "消息名称":

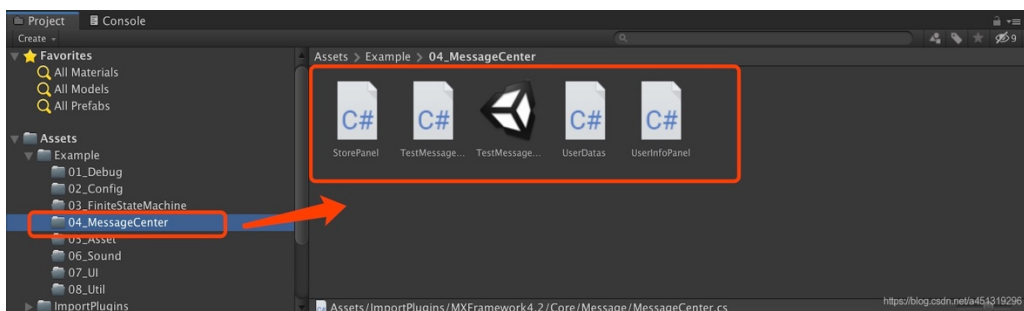
            数据类型 msg= (数据类型) values;
            break;
    }
}

```

四、 源码结构

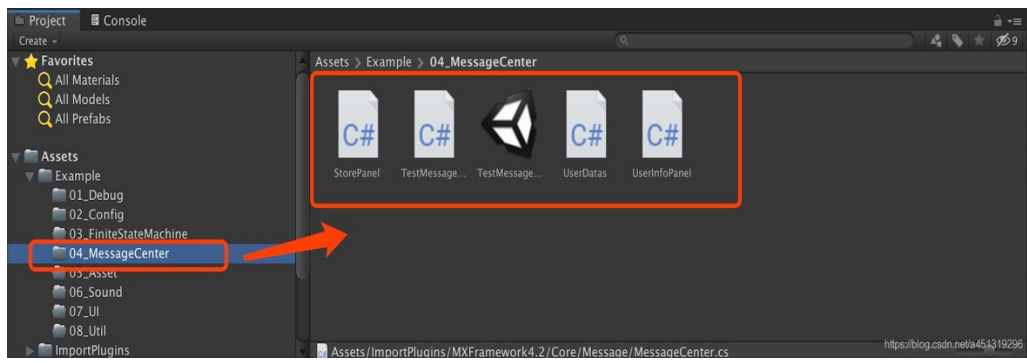
1. 核心代码路径

ImportPlugins/MXFramework**/Core/Message



2. 示例工程路径

Example/04_MessageCenter



五、 源码下载

MxFramework 源码: <https://github.com/yongliangchen/MXFramework>

博客文章: <https://blog.csdn.net/a451319296/article/details/109057994>

QQ 交流群: 1079467433