

# Launch

Launch 类为 Qomicex Core的核心类之一，负责处理Minecraft启动相关的工作。

你可以点击[这里](#)查看详细方法列表

## 调用示例

### 启动游戏

```
//Example
```

# Launch

Launch 类为 Qomicex Core的核心类之一，负责处理Minecraft启动相关的工作。

## 方法列表

### 启动游戏

方法	Launch	返回值类型	bool
参数	参数类型	描述	
param	<a href="#">LauncherParam</a>	游戏启动的必要参数	

### 生成启动参数

方法	SelectParam	返回值类型	string
参数	参数类型	描述	
param	<a href="#">LauncherParam</a>	游戏启动的必要参数	

## 解压Natives

方法	UnzipNatives	返回值类型	bool
参数	参数类型	描述	
jsonPath	string	游戏启动的必要参数	
minecraftPath	string	游戏启动的必要参数	
versionPath	string	游戏启动的必要参数	

### 解压Natives(异步)

方法	UnzipNativesAsync	返回值类型	Task<bool>
参数	参数类型	描述	
jsonPath	string	游戏启动的必要参数	
minecraftPath	string	游戏启动的必要参数	

versionPath	string	游戏启动的必要参数
-------------	--------	-----------

# Account

Account包含多个类,为 Qomicex Core的核心功能之一,负责处理Minecraft账户登录相关的工作。

## Microsoft类

Microsoft 类负责微软账户登录相关的工作。

你可以点击[这里](#)查看详细方法列表

## 使用示例

### ⚠ 警告

该类需要在使用前构造时传入你的Azure门户ID。

```
var ms = new Qomicex_Core.Modules.Account.Microsoft("你的应用ID");
```

## 使用微软OAuth登录Minecraft正版账户

```
using Qomicex_Core.Modules.Account.Microsoft;
//注册类
var ms = new Microsoft("你的门户ID");

//开始OAuth登录
OAuthResponse oauth = await ms.OAuthLogin();

//创建一个计时器,用于判断登录是否超时
var stopwatch = Stopwatch.StartNew();

//将用户代码复制到剪切板以使用户登录时粘贴
Clipboard.SetText(oauth.UserCode);

MessageBox.Show($"您的用户代码 {oauth.UserCode} 已复制到剪切板");

//打开用户验证页面
Process.Start(new ProcessStartInfo
{
    FileName = oauth.VerificationUri,
    UseShellExecute = true
});

//初始化登录状态字典
```

```

Dictionary<string, string> state = new Dictionary<string, string>();

//启用循环轮询用户登录状态,当超时或获取到access_token时停止
while (stopwatch.Elapsed.TotalSeconds < oauth.ExpiresIn)
{
    //获取用户登录状态
    state = await ms.GetUserAuthorizationState(oauth);
    //检查返回的状态字典
    if (state.ContainsKey("error"))
    {
        //字典存在错误信息
        string err = state["error"];

        ///如果错误是等待授权或轮询过快,则继续等待
        if (!(err == "authorization_pending" || err == "slow_down"))
        {
            MessageBox.Show($"登录失败: {err}", "错误", MessageBoxButton.OK,
            MessageBoxImage.Error);
            break;
        }

        //等待指定的时间后继续轮询
        System.Threading.Thread.Sleep(oauth.Interval * 1000);
    }
    else if (state.ContainsKey("access_token"))
    {
        //如果存在access_token键则登录成功,跳出循环
        break;
    }
    else
    {
        //处理其它错误
        MessageBox.Show($"登录失败:\n{state["error"]}", "提示",
        MessageBoxButton.OK, MessageBoxImage.Error);
        break;
    }
}
//登录成功后停止计时器
stopwatch.Stop();

//获取用户信息
//当登录成功后,状态字典中会包含access_token和refresh_token
var userInfo = await ms.GetUserInfo(state["access_token"], state["refresh_token"]);

MessageBox.Show($"登录成功! \nName: {userInfo.Name}\nUuid: {userInfo.Uuid}\nToken:

```

```
{userInfo.Token}", "成功", MessageBoxButton.OK, MessageBoxImage.Information);
```

## 刷新微软登录令牌

```
using Qomicex_Core.Modules.Account.Microsoft;
```

```
//注册类
```

```
var ms = new Microsoft("你的门户ID");
```

```
//刷新令牌
```

```
var userInfo = await ms.RefreshUserInfo(refresh_token);
```

# Microsoft 类

Microsoft 类负责微软账户登录相关的工作。

## ⚠ 警告

该类需要在使用前构造时传入你的Azure门户。

```
var ms = new Qomicex_Core.Modules.Account.Microsoft("你的应用ID");
```

## 方法列表

### Oauth登录

方法	OAuthLogin	返回值类型	Task<OAuthResponse>
参数	参数类型	描述	

### 获取用户验证状态

方法	GetUserAuthorizationState	返回值类型	Task<Dictionary<string, string>>
参数	参数类型	描述	
oAuthResponse	OAuthResponse	微软登录结果信息	

这里的返回值是一个字典，包含了用户的验证状态信息。当登录成功时，字典中会包含以下键值对：

- access\_token: 用户的访问令牌
- refresh\_token: 用户的刷新令牌

这两个令牌用于后续的API调用和用户信息获取。

当登录没有成功时，字典中会包含以下键值对：

- error: 错误信息

### 获取用户账户信息

方法	GetUserInfo	返回值类型	Account
----	-------------	-------	---------

参数	参数类型	描述
accessToken	string	微软AccessToken
refresh_token	string	微软RefreshToken

## 刷新用户账户信息

方法	RefreshUserInfo	返回值类型	<a href="#">Account</a>
参数	参数类型	描述	
refresh_token	string	微软RefreshToken	



# Tools

Tools 为QomiceX Core 包含多个类，负责提供多种实用功能。

## 类列表

你可以下方超链接查看详细方法列表

- [MainTools类](#)
- [MFTTools类](#)

## 调用示范

### 生成离线UUID

```
MainTools tools = new MainTools();  
string uuid = tools.NameToUuid({用户名});
```

# MainTools

MainTools 类为Tools 类中的主要工具类

## 方法列表

### 生成离线UUID

方法	NameToUuid	返回值类型	string
参数	参数类型	描述	
name	string	玩家名称，用于生成离线模式UUID	

### 获取最优内存

方法	GetOptimalMemory	返回值类型	long
参数	参数类型	描述	
percentage	double	内存占比（默认0.6，即60%），计算可用内存的指定百分比（单位：MB）	

### 是否为64位系统

方法	Is64	返回值类型	bool
参数	参数类型	描述	
无参数			

### 快速获取有效 Minecraft 版本列表

方法	SearchVersionsFast	返回值类型	List<string>
参数	参数类型	描述	
GameDir	string	Minecraft游戏目录路径，仅返回存在对应.json文件的版本名称	

### 获取有效的 Minecraft 版本列表

此方法会在获取版本列表后获取ModLoader信息，验证Json是否正常等后返回DataDetails.Version类型

方法	SearchVersions	返回值类型	<a href="#">List&lt;DataDetails.Version&gt;</a>
参数	参数类型	描述	
GameDir	<code>string</code>	Minecraft游戏目录路径	

## 解压文件

方法	Unzip	返回值类型	<code>bool</code>
参数	参数类型	描述	
zipFilePath	<code>string</code>	压缩包文件路径	
targetDir	<code>string</code>	解压目标目录	

## 删除指定后缀外的文件

方法	DeleteExcept	返回值类型	<code>bool</code>
参数	参数类型	描述	
folderPath	<code>string</code>	目标文件夹路径	
keepSuffix	<code>string</code>	需要保留的文件后缀（如.txt）	

## 获取Mod加载器类型

方法	GetModLoaderType	返回值类型	<code>string[]</code>
参数	参数类型	描述	
Version	<code>string</code>	Minecraft版本名称	
GameDir	<code>string</code>	Minecraft游戏目录路径，用于读取版本.json文件	

## 从Zip压缩包提取指定文件夹

方法	ExtractFolderFromZip	返回值类型	<code>List&lt;string&gt;</code>
参数	参数类型	描述	
zipPath	<code>string</code>	压缩包路径	

targetFolderInZip	string	压缩包内需要提取的文件夹名称
outputDirectory	string	提取到的目标目录

## 检查版本可用性

方法	CheckVersionAvailablity	返回值类型	<a href="#">State</a>
参数	参数类型	描述	
GameDir	string	Minecraft游戏目录路径	
Version	string	需要检查的版本名称	

## 检查是否为管理员权限

方法	IsAdministrator	返回值类型	bool
参数	参数类型	描述	
无参数			

## 验证文件 SHA1 哈希

方法	VerifyFileSha1	返回值类型	bool
参数	参数类型	描述	
filePath	string	需要验证的文件路径	
expectedHash	string	预期的SHA1哈希值（小写字母串）	

## 搜索 Minecraft 目录

方法	SearchMinecraftDir	返回值类型	List<string>
参数	参数类型	描述	
无参数			

## 从Zip压缩包读取指定文件

方法	ReadSpecifyFileFromZip	返回值类型	byte[]
参数	参数类型	描述	
path	string	压缩包文件路径	
fileName	string	需要读取的压缩包内文件名称	



































# 类型列表

这里是Qomicex Core中所有的的类型列表。

## LauncherParam

该类型派生自[Launch](#)类，用于储存游戏启动的必要参数。

名称	类型	描述
GameDir	string	游戏目录
LauncherName	string	启动器名称
Java	<a href="#">Java</a>	Java信息
FullScreen	bool	是否全屏
Width	string	游戏窗口宽度，默认854
Height	string	游戏窗口高度，默认480
...	...	同 <a href="#">Launch</a>

## Account

名称	类型	描述
Name	string	用户名称
Uuid	string	用户唯一标识符（UUID）
Token	string	登录令牌
AccessToken	string	访问令牌，默认值为 "0"
RefreshToken	string	用于刷新访问令牌
LoginMethod	string	登录方式，如 "Legacy"

## Java

类型用于储存Java信息

名称	类型	描述
Name	string	名称
Path	string	路径
Version	string	版本号
VersionID	int	版本标识符
Type	string	类型
Arch	string	架构（如 x64）

## Launch

用于储存游戏启动的必要参数。

名称	类型	描述
Account	<a href="#">Account</a>	用户账户信息
Java	<a href="#">Java</a>	Java 环境信息
WindowSize.Width	string	游戏窗口宽度，默认 854
WindowSize.height	string	游戏窗口高度，默认 480
DevideVersion	bool	版本隔离
AdditionalParam	string	启动器附加参数
Version	string	启动器指定版本号
MaxMemory	string	最大分配内存，如 "4096"

## Version

储存Minecraft版本信息的类型。

名称	类型	描述
GameDir	string	游戏根目录路径
Name	string	版本名称

名称	类型	描述
Type	<code>string[]</code>	ModLoader类型标签, 如 ["OptiFine-{version}", "Forge-{version}"]
GameVersion	<code>string</code>	游戏版本号, 如 "1.20.1"
VersionDir	<code>string</code>	版本文件夹路径
State	<code>string</code>	当前状态, 如 "Available"
StateDescribe	<code>string</code>	状态描述, 如 "Everything is OK"

## OAuthResponse

名称	类型	描述
DeviceCode	<code>string</code>	设备授权码, 用于设备端轮询授权状态
UserCode	<code>string</code>	用户输入的验证码, 用于在验证页面完成授权
VerificationUri	<code>string</code>	用户访问的验证地址, 用于输入验证码完成授权
ExpiresIn	<code>int</code>	授权码的有效时间 (秒)
Interval	<code>int</code>	轮询授权状态的时间间隔 (秒)

## State

MainTools 中的用于存储状态的类型。

名称	类型	描述
Name	<code>string</code>	当前状态, 如 "Available","Error","Warn"
Describe	<code>string</code>	状态描述, 如 "Everything is OK"