

Qomicex.Core 开发教程

概述

欢迎来到 Qomicex.Core 开发教程！本教程将帮助您了解 Qomicex.Core 的架构设计、核心功能和开发流程，为您快速上手开发基于 Qomicex.Core 的 Minecraft 启动器或相关工具提供指导。

项目简介

Qomicex.Core 是一个功能强大的 Minecraft 启动器核心库，提供了完整的启动流程管理、游戏资源管理、账户认证、Mod 管理等功能。它采用模块化设计，具有高度的可扩展性和可定制性，适用于各种规模的 Minecraft 相关应用开发。

主要特性

- **完整的启动流程管理：**从游戏安装到启动的完整流程
- **灵活的账户认证系统：**支持多种认证方式，包括 Microsoft、Yggdrasil 等
- **全面的资源管理：**游戏版本下载、Mod 管理、资源包管理等
- **强大的 Mod 支持：**支持 Forge、Fabric、Quilt 等多种 Mod 加载器
- **跨平台兼容：**支持 Windows、Linux 和 macOS 系统
- **模块化设计：**高度模块化的架构，易于扩展和定制
- **异步编程：**大量使用异步 API，提供良好的用户体验

目录

[[toc]]

在我们开始之前

在开始使用 Qomicex.Core 之前，您需要对开发环境和项目配置有一定的了解。请先阅读 [在我们开始之前](#) 章节，确保您的开发环境已正确配置。

安装并配置 Qomicex.Core

在配置好开发环境后，您需要安装并配置 Qomicex.Core 库。请阅读 [安装并配置 Qomicex.Core](#) 章节，了解如何安装和配置 Qomicex.Core。

验证模型

Qomicex.Core 提供了多种 Minecraft 验证模型，包括离线验证、Yggdrasil 验证和 Microsoft 验证。请阅读 [验证模型](#) 章节，了解如何使用这些验证模型。

安装器

Qomicex.Core 提供了多种 Mod 加载器的安装器，包括 Forge、Fabric、Quilt、LiteLoader 和 OptiFine。请阅读 [安装器](#) 章节，了解如何使用这些安装器。

资源补全器

资源补全器负责下载和管理 Minecraft 游戏所需的各种资源文件。请阅读 [资源补全器](#) 章节，了解如何使用资源补全器。

快速开始

环境要求

- .NET 8.0 或更高版本
- Visual Studio 2022 或其他兼容的 IDE
- Git 版本控制工具

创建第一个项目

```
# 创建项目
dotnet new console -n QomicexDemo
cd QomicexDemo

# 添加 NuGet 源（如果需要）
dotnet nuget add source https://api.nuget.org/v3/index.json --name nuget.org

# 添加 Qomicex.Core 依赖
dotnet add package Qomicex.Core
```

编写简单的启动器

```
using Qomicex.Core;
using Qomicex.Core.Modules.Helpers.Account.Microsoft;
using Qomicex.Core.Modules.Helpers.Resources;
using Qomicex.Core.Modules.Launcher;
using System;
using System.Threading.Tasks;

namespace QomicexDemo
{
    class Program
    {
        static async Task Main(string[] args)
        {
            Console.WriteLine("Qomicex.Core 启动器示例");
        }
    }
}
```

```

try
{
    // 1. 初始化核心组件
    var gameResourceHelper = new GameResourceHelper();
    var microsoftAuth = new Microsoft();

    // 2. 获取游戏版本列表
    Console.WriteLine("获取游戏版本列表...");
    var versions = await gameResourceHelper.GetMinecraftListAsync(1);
    Console.WriteLine($"可用版本数: {versions.Count}");

    // 3. 打印最新版本
    var latestVersion = versions.Find(v => v.Type == "release");
    Console.WriteLine($"最新正式版: {latestVersion?.Id}");

    // 4. 启动游戏 (简化示例)
    var launcher = new Qomicex.Core.Modules.Launcher.Launcher();
    var launchParams = new LauncherParam
    {
        GameDir = @"C:\Users\YourName\.minecraft",
        Version = latestVersion?.Id,
        JavaPath = "java",
        MaxMemory = 4096,
        Username = "Player"
    };

    Console.WriteLine("正在启动游戏...");
    var result = await launcher.StartGameAsync(launchParams);
    Console.WriteLine("游戏启动状态: " + result);

    catch (Exception ex)
    {
        Console.WriteLine($"发生错误: {ex.Message}");
    }

    Console.WriteLine("按任意键继续...");
    Console.ReadKey();
}
}

```

架构设计

核心架构

Qomicex.Core 采用三层架构设计：

1. **基础层**: 包含通用工具、数据模块、日志分析等基础功能
2. **核心服务层**: 提供游戏资源管理、账户认证、启动器核心功能
3. **业务逻辑层**: 处理游戏启动流程、Mod 管理等复杂业务逻辑

模块架构

Qomicex.Core 采用模块化设计，每个功能模块独立实现，提供清晰的接口：

- **数据模块**: 定义核心数据结构和配置
- **工具模块**: 提供通用的工具方法和功能
- **账户模块**: 处理用户认证和账户管理
- **资源模块**: 管理游戏资源、Mod、资源包等
- **启动器模块**: 负责游戏启动流程管理

核心功能开发

1. 游戏版本管理

```
using Qomicex.Core.Modules.Helpers.Resources;

var resourceHelper = new GameResourceHelper();

// 获取版本列表
var versions = await resourceHelper.GetMinecraftListAsync(1);

// 根据版本ID获取详细信息
var versionInfo = await resourceHelper.GetMinecraftVersionManifest(1);

// 下载游戏资源
var downloader = new ResourceDownloader();
await downloader.DownloadGameResourcesAsync("1.20.1", @"C:\Game\.minecraft");
```

2. 账户认证

Microsoft 账户认证

```
using Qomicex.Core.Modules.Helpers.Account.Microsoft;

var microsoftAuth = new Microsoft();

// 获取授权 URL
var authUrl = microsoftAuth.GetAuthorizationUrl();
Console.WriteLine($"请访问以下 URL 进行授权: {authUrl}");

// 获取用户输入的授权码
```

```

Console.WriteLine("请输入授权码: ");
var code = Console.ReadLine();

try
{
    var authResult = await microsoftAuth.AuthorizeAsync(code);
    Console.WriteLine($"访问令牌: {authResult.AccessToken}");
    Console.WriteLine($"刷新令牌: {authResult.RefreshToken}");
    Console.WriteLine($"过期时间: {authResult.ExpiresIn} 秒");
}
catch (Exception ex)
{
    Console.WriteLine($"授权失败: {ex.Message}");
}

```

Yggdrasil 认证

```

using Qomicex.Core.Modules.Helpers.Account.Yggdrasil;

var yggdrasilAuth = new Yggdrasil();

// 登录
var loginResult = await yggdrasilAuth.LoginAsync("username", "password");

// 验证访问令牌
var isValid = await yggdrasilAuth.ValidateAsync(loginResult.AccessToken);

// 刷新访问令牌
var refreshResult = await yggdrasilAuth.RefreshAsync(
    loginResult.AccessToken,
    loginResult.ClientToken
);

```

3. Mod 管理

```

using Qomicex.Core.Modules.Helpers.Resources.Expansion.CurseForge;

var curseForgeMods = new Mods("your-api-key");

// 搜索 Mod
var searchResults = await curseForgeMods.SearchAsync("sodium", "1.20.1");

// 获取 Mod 详情
var modDetails = await curseForgeMods.GetModDetailsAsync("sodium");

```

```
// 下载 Mod
```

```
await curseForgeMods.DownloadModAsync("sodium", "0.4.14",  
@"C:\Game\.minecraft\mods\sodium.jar");
```

4. 游戏启动

```
using Qomicex.Core.Modules.Launcher;
```

```
var launcher = new Launcher();
```

```
var launchParams = new LauncherParam  
{  
    GameDir = @"C:\Game\.minecraft",  
    Version = "1.20.1",  
    JavaPath = @"C:\Program Files\Java\jre-17\bin\javaw.exe",  
    MaxMemory = 4096,  
    MinMemory = 1024,  
    WindowWidth = 1280,  
    WindowHeight = 720,  
    Fullscreen = false,  
    Server = null,  
    Username = "Player",  
    AuthenticationToken = "your-auth-token"  
};
```

```
// 启动游戏
```

```
var result = await launcher.StartGameAsync(launchParams);
```

```
if (result)  
{  
    Console.WriteLine("游戏启动成功！");  
}  
else  
{  
    Console.WriteLine("游戏启动失败！");  
}
```

高级主题

自定义资源源

```
using Qomicex.Core.Modules.Helpers.Resources;
```

```

// 创建自定义资源源
var customSource = new ResourceSource
{
    Name = "My Custom Source",
    BaseUrl = "https://my-custom-mirror.com/",
    GameManifestUrl = "https://my-custom-mirror.com/game_manifest.json",
    LibraryUrl = "https://my-custom-mirror.com/libraries/",
    AssetUrl = "https://my-custom-mirror.com/assets/",
    VersionUrl = "https://my-custom-mirror.com/versions/"
};

// 使用自定义资源源获取版本列表
var resourceHelper = new GameResourceHelper(customSource);
var versions = await resourceHelper.GetMinecraftListAsync();

```

扩展 Mod 支持

```

using Qomicex.Core.Modules.Helpers.Installers;

public class MyCustomModLoaderInstaller : InstallerBase
{
    public override Task<bool> InstallAsync(
        string versionId,
        string inheritsFromJson,
        string? para1,
        string? para2,
        string? para3,
        string? para4
    )
    {
        // 实现自定义 Mod 加载器的安装逻辑
        return Task.FromResult(true);
    }
}

// 使用自定义安装器
var customInstaller = new MyCustomModLoaderInstaller();
await customInstaller.InstallAsync(
    "1.20.1-mycustomloader-1.0",
    null,
    "1.0",
    "1.20.1",
    null,
    null
);

```

自定义启动参数

```
using Qomicex.Core.Modules.Launcher;

var launchParams = new LauncherParam
{
    GameDir = @"C:\Game\.minecraft",
    Version = "1.20.1",
    JavaPath = @"C:\Program Files\Java\jre-17\bin\javaw.exe",
    MaxMemory = 4096,
    MinMemory = 1024,
    // 自定义 JVM 参数
    JvmArgs = new List<string>
    {
        "-XX:+UseG1GC",
        "-XX:G1HeapRegionSize=4M",
        "-XX:+UnlockExperimentalVMOptions",
        "-XX:+UseCompressedOops"
    },
    // 自定义游戏参数
    GameArgs = new List<string>
    {
        "--fullscreen",
        "--server", "mc.hypixel.net",
        "--port", "25565"
    }
};
```

常见问题

1. 游戏启动失败

问题：游戏无法正常启动，提示找不到 Java 路径。

解决方案：

```
using Qomicex.Core.Modules.Helpers;

var javaHelper = new JavaHelper();

// 搜索系统中的 Java
var javaLocations = await javaHelper.SearchForJavaAsync();

if (javaLocations.Any())
{
```

```

var firstJava = javaLocations.First();

var launchParams = new LauncherParam
{
    JavaPath = firstJava.Path,
    // 其他参数...
};

}
else
{
    Console.WriteLine("未找到 Java 运行时，请确保已安装 Java。");
}

```

2. 资源下载失败

问题：游戏资源或 Mod 无法下载，网络连接失败。

解决方案：

```

using Qomicex.Core.Modules.Helpers.Resources;

var downloader = new ResourceDownloader();
downloader.MaxRetries = 3; // 增加重试次数
downloader.Timeout = TimeSpan.FromMinutes(5); // 增加超时时间

try
{
    await downloader.DownloadGameResourcesAsync("1.20.1", @"C:\Game\.minecraft");
}
catch (Exception ex)
{
    Console.WriteLine($"下载失败: {ex.Message}");
    Console.WriteLine("请检查网络连接或尝试使用其他资源源。");
}

```

性能优化

1. 资源缓存策略

```

using Qomicex.Core.Modules.Helpers.Resources;

// 配置资源缓存
var cacheOptions = new ResourceCacheOptions
{
    MaxCacheSize = 10 * 1024 * 1024 * 1024, // 10GB
}

```

```

CacheExpiration = TimeSpan.FromDays(7), // 缓存过期时间
UseDiskCache = true, // 启用磁盘缓存
CacheLocation = @"C:\Game\.minecraft\cache" // 缓存位置
};

var resourceHelper = new GameResourceHelper(cacheOptions);

// 使用缓存获取版本信息
var versions = await resourceHelper.GetMinecraftListAsync(1);

```

2. 并发下载优化

```

using Qomicex.Core.Modules.HelpersDownloader;

var downloader = new Downloader
{
    MaxConcurrentDownloads = 8, // 并发下载数
    ChunkSize = 4 * 1024 * 1024, // 分块大小
    RetryCount = 3 // 重试次数
};

var downloadTasks = new List<Task>
{
    downloader.DownloadAsync("http://example.com/file1.jar", "file1.jar"),
    downloader.DownloadAsync("http://example.com/file2.jar", "file2.jar"),
    downloader.DownloadAsync("http://example.com/file3.jar", "file3.jar")
};

await Task.WhenAll(downloadTasks);

```

贡献指南

我们欢迎开发者为 Qomicex.Core 项目做出贡献！以下是贡献步骤：

- 1. Fork 项目**: 在 GitHub 上 Fork 项目到自己的仓库
- 2. 克隆项目**: 克隆自己的仓库到本地
- 3. 创建分支**: 创建一个新的分支用于开发
- 4. 开发功能**: 实现新功能或修复 bug
- 5. 编写测试**: 为新功能编写测试代码
- 6. 提交修改**: 提交修改到本地仓库
- 7. 创建 PR**: 在 GitHub 上创建 Pull Request
- 8. 等待审核**: 等待项目维护者审核您的 PR

许可证

Qomicex.Core 项目采用 MIT 许可证，允许您自由地使用、修改和分发项目代码。详细信息请查看项目根目录下的 LICENSE 文件。

联系方式

- **项目仓库:** <https://github.com/TheMyceliumOfAntan/Qomicex.Avalonia>
- **问题反馈:** <https://github.com/TheMyceliumOfAntan/Qomicex.Avalonia/issues>
- **讨论交流:** <https://github.com/TheMyceliumOfAntan/Qomicex.Avalonia/discussions>

其他资源

- **API 文档:** <http://localhost:8083/api/>
- **示例代码:** <https://github.com/TheMyceliumOfAntan/Qomicex.Avalonia/tree/main/Examples>
- **变更日志:**
<https://github.com/TheMyceliumOfAntan/Qomicex.Avalonia/blob/main/CHANGELOG.md>

Namespace Qomicex.Core

Classes

[Class1](#)

[DataModules](#)

[DataModules.DataDetails](#)

[DataModules.DataDetails.Account](#)

[DataModules.DataDetails.Java](#)

[DataModules.DataDetails.Launcher](#)

[DataModules.DataDetails.Launcher.WindowSize](#)

[DataModules.DataDetails.Version](#)

Structs

[DataModules.State](#)

[DataModules.SystemInfo](#)

Enums

[DataModules.State.StateCode](#)