EUResKit

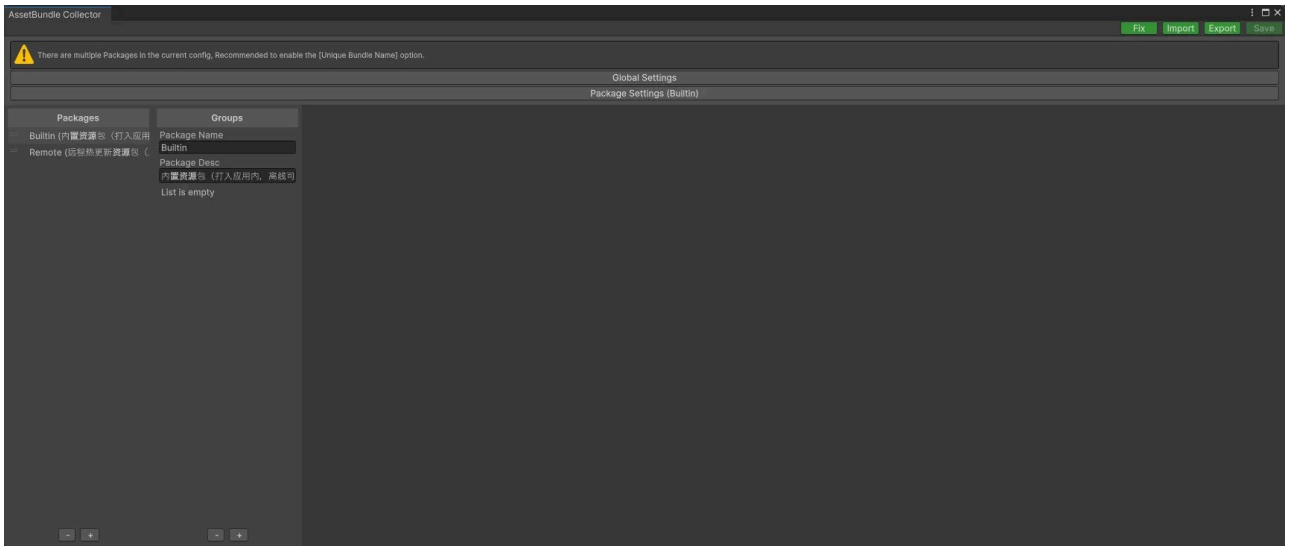
框架的入口面板

- 资源配置
- 代码生成
- 模块管理工具

资源配置

EURes的SO管理

AssetBundleCollectorSetting 为YooAssets自身的资源收集器



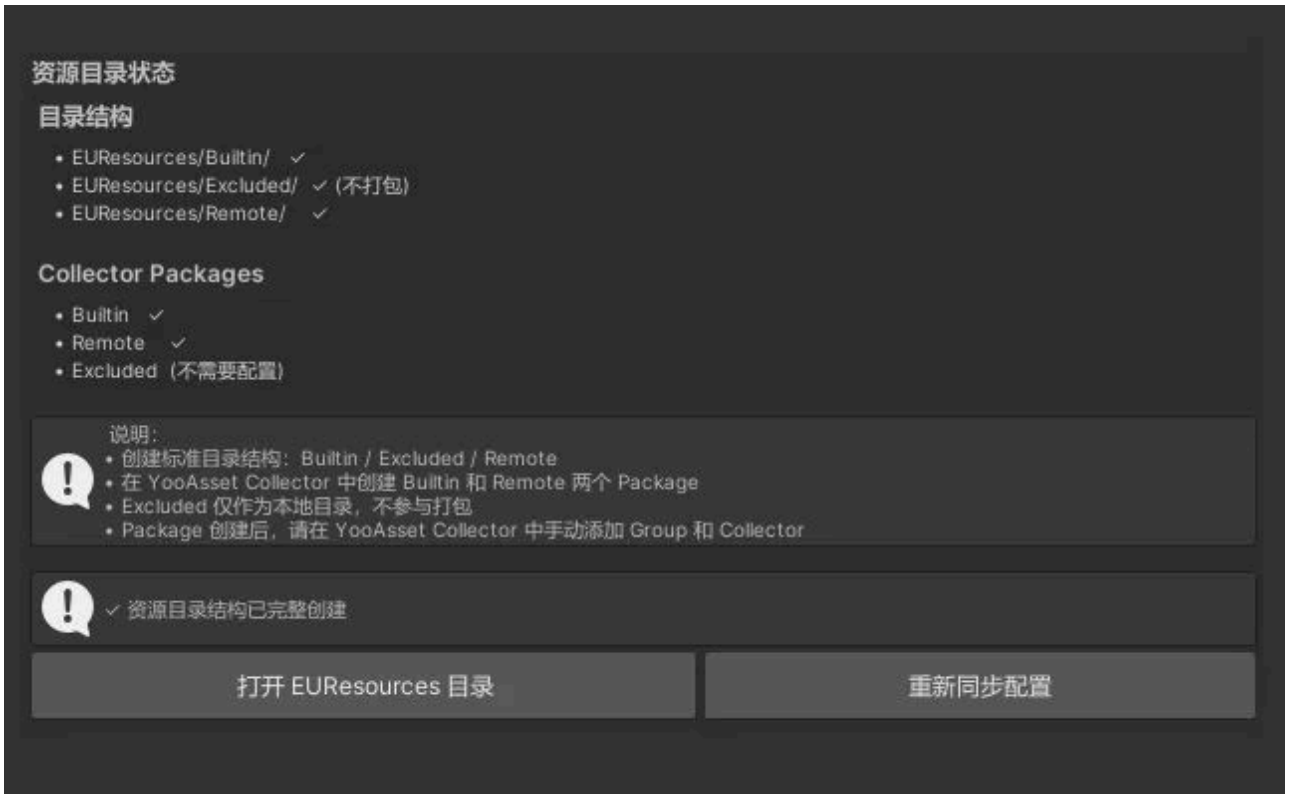
EUResServerConfig 为 代码读取的服务器配置 如果资源为远程加载的情况下进行配置

YooAssetSettings 为 YooAsset配置打包情况的处理

EUResKitPackageConfig 每个包初始化时的加载方式的配置 回字节从AssetBundleCollectorSetting中获取包数据

只需要从AssetBundleCollector同步即可

资源目录关联 创建默认目录结构


资源目录状态
目录结构
• EUResources/Builtin/ ✓
• EUResources/Excluded/ ✓ (不打包)
• EUResources/Remote/ ✓
Collector Packages
• Builtin ✓
• Remote ✓
• Excluded (不需要配置)
说明:
• 创建标准目录结构: Builtin / Excluded / Remote
• 在 YooAsset Collector 中创建 Builtin 和 Remote 两个 Package
• Excluded 仅作为本地目录，不参与打包
• Package 创建后，请在 YooAsset Collector 中手动添加 Group 和 Collector
✓ 资源目录结构已完整创建
打开 EUResources 目录
重新同步配置

代码生成

1. 用户操作界面的交互代码

生成EUResKitUserOpePopUp.cs.sbn 文件的代码 以及对应的prefab(修改为游戏内对应的游戏样式) 用于管理资源更新时，用户的操作和通知

2. EUResKit代码。分布类

固定生成的部分DefaultResKit.Generated，只提供InitPackageResAsync GetPackage SetDefaultPackage IsInitialized--初始化，获取包，设置默认包，判断包是否初始化

DefaultResKit.cs，包含默认的UI交互部分的逻辑和一个初始化所有包的入口。用户可以重写，处理新的资源部分如LoadSprite之类的接口也可以写在这里。EUResKit的一个处理思路就是只负责包的初始化和获取包，至于Handle让各自的Kit或者在胶水层处理，如EUUIKit的内部处理

模块管理工具

一次性生成/删除 所有的配置和代码部分，刷新程序集引用更改后的命名空间设置。

快速入门

引入框架之后，在模块管理中点击一键生成所有配置和代码

处理好YooAssetCollector，设置好包的加载方式

代码中，游戏初始化处，如下调用资源包的初始化

C#

```
using EUFramework.Extension.EURes; // 根据实际命名空间修改
using Cysharp.Threading.Tasks;
using UnityEngine;
```

```
4
5  public class GameLauncher : MonoBehaviour
6  {
7      private async void Start()
8      {
9          // 可选：设置下载进度回调（用于自定义 Loading 条）
10         EUResKit.SetDownloadProgressCallback((packageName, totalCount,
   currentCount, totalBytes, currentBytes) =>
11         {
12              float progress = (float)currentBytes / totalBytes;
13              Debug.Log($"[{packageName}] 下载进度: {progress * 100:F1}%");
14         });
15
16         // 初始化所有资源包
17         // 如果检测到更新，会自动弹出内置 UI 提示用户下载
18         bool success = await EUResKit.InitializeAllPackagesAsync();
19
20         if (success)
21         {
22             Debug.Log("资源初始化成功");
23             StartGame();
24         }
25         else
26         {
27             Debug.LogError("资源初始化失败");
28         }
29     }
30 }
```

如下调用资源的加载和句柄的释放

C#

```
1  using YooAsset;
2
3  // 异步加载预制体
4  var handle = EUResKit.GetPackage().LoadAssetAsync<GameObject>
   ("Assets/Prefabs/Player.prefab");
5  await handle.ToUniTask();
6
7  if (handle.Status == EOperationStatus.Succeed)
8  {
9      GameObject player = handle.AssetObject as GameObject;
10     Instantiate(player);
11 }
12
```

```
// 使用完毕释放资源
handle.Release();
```