

tabletool

配置表工具，包含代码生成，支持自定义类型

支持Resource或Addressable加载

Unity版本：2018及以上

安装

1. Editor->ProjectSettings->PackageManager->Scoped registries,添加条目：

```
1  Name:Typhoon
2  URL:https://registry.npmjs.org
3  Scope(s):com.typhoon
4
5  也可以手动编辑Package文件夹内的manifest.json，添加条目：
6  "scopedRegistries": [
7    {
8      "name": "Typhoon",
9      "url": "https://registry.npmjs.org",
10     "scopes": [
11       "com.typhoon"
12     ]
13   }
14 ]
15
```

2. 打开PackageManager窗口等待刷新，左上角选择My Registries一栏选中插件导入即可
3. 安装完毕后，顶部菜单会多出“配置表工具”
4. 菜单栏：配置表工具-->选项，完成安装后，按需求进行设置，保存并应用即可完成部署
5. 菜单栏：配置表工具-->打开文件夹，可方便跳转到配置表文件夹
6. 菜单栏：配置表工具-->一键导表，生成代码和对应的配置表资源

使用说明

sheet前三行空出，**第一行填标签，第二行注释或者说明，第三行填字段名**（主要用于代码导出，注意命名）

注意事项：

1. 仅支持xlsx文件，单个xlsx可以存在多个sheet

2. sheet命名规范: 中文名|英文名, 如果需要忽略不导出追加"|#ignore ",例如: 装备|equip|#ignore
3. 除了#link或者#ignore所在列外, 都需要有类型标签, 例如#int,#float...

标签说明:

标签名	说明
#key	唯一标识列 (ID列), 用来校验key或者被#link进行表间查找, 该列成员只能为int或者string,暂不支持其它类型做为key
#item>对象名[成员名]>子对象名[成员名]>孙对象名[成员名]	二次封装Object,[成员名]为可选项, 不填取对象名作为类型名称
#link>表格名>匹配列名	关联到其它表的某行数据, #link描述的列不需要填写类型标签, 例如#int,#float,它会以关联表对应的列项为准
#ignore	忽略项, 对应列无效, 不导出 (一般用于补充对应的说明)
#enumvalue	对应列为枚举值,需要和#key#int一起使用, 一个sheet只能存在一个#enumvalue, 搭配#enumname生成枚举
#enumname	对应列为枚举名, 需要和#string一起使用, 一个sheet只能存在一个#enumname, 搭配#enumvalue生成枚举
#int	int类型
#long	long类型
#float	float类型
#double	double类型
#bool	bool类型 (1为ture,0为false)
#string	string类型
#array<int>	int[]类型, 数据用 ' ' 分割,例如: 1 2 3 4 5
#array<long>	long[]类型, 数据用 ' ' 分割,例如: 1 2 3 4 5
#array<float>	float[]类型, 数据用' '分割,例如: 0.1 0.2 0.3
#array<double>	double[]类型, 数据用'分割',例如: 0.1 0.2 0.3
#array<bool>	bool[]类型, 数据用 ' ' 分割,例如: 1 0 1 1
#array<string>	string[]类型, 数据用 ' ' 分割,例如:

相关API

初始化

```
1 //初始化（同步模式，仅支持Resource模式）
2 TableHelper.Build();
3 //初始化（异步模式）
4 await TableHelper.BuildAsync();
```

查询

```
1 //查询指定表，对应列，匹配ID项（比较少用）
2 TableHelper.AllTables.GetTable("表名").Get("列名", "字符串ID");
3 TableHelper.AllTables.GetTable("表名").Get("列名", 数字ID);
4 //使用自动生成的API查询对应的表项,例如:
5 TableHelper.AllTables.FindTableValueExampleById(1);
6 TableHelper.AllTables.FindTableValueExampleByName("数据1");
```

检查

```
1 //是否存在表
2 TableHelper.AllTables.HasTable(填表名);
3 //是否存在表项
4 TableHelper.AllTables.HasTableValue(填表名,填列名,填ID);
5 //指定表检查
6 TableHelper.AllTables.valueExample.Has(填列名,填ID);
```

遍历

```
1 //例如：遍历valueExample表
2 var tableList = TableHelper.AllTables.valueExample.Data;
3 foreach (var element in tableList)
4 {
5     Debug.Log(element.Name);
6 }
```

