

I18nLib 使用教程

Version 1.0

1. 关于此教程

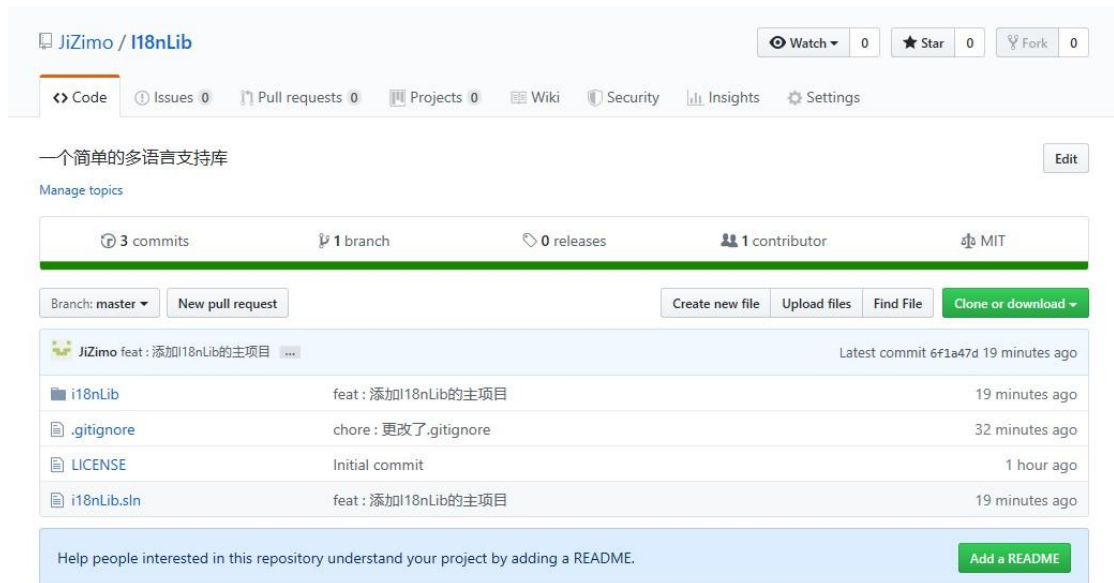
该教程是一个 I18nLib 库的入门教程，并非是该 I18nLib 的 API 手册。因此该教程仅仅是简单的介绍了 I18nLib 的用法，并未涉及其全部的 API。

2. 关于该库

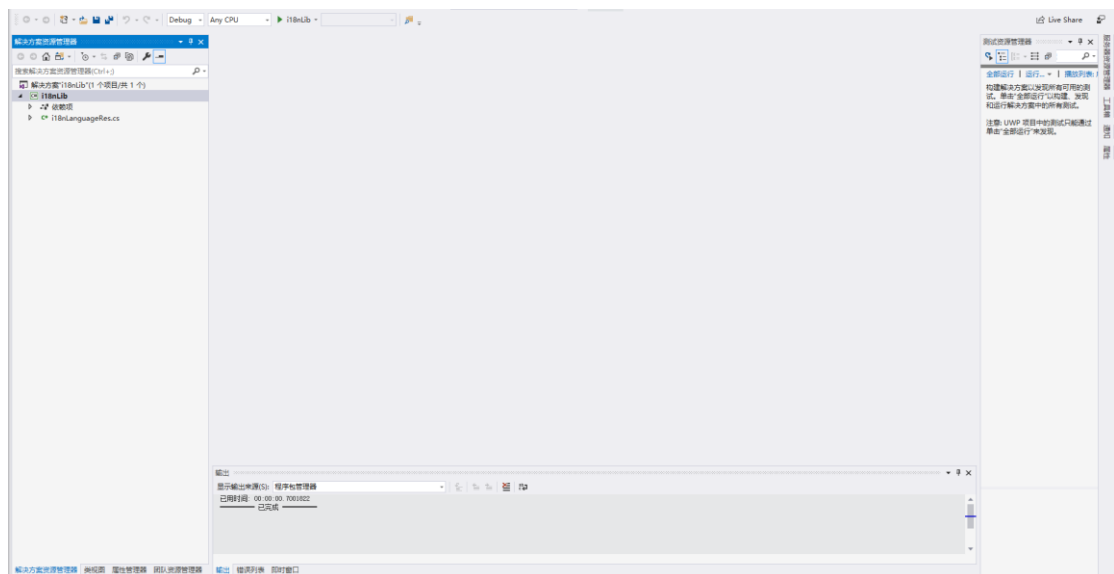
I18nLib 库使用 C#编写，运行于 CLI 环境下，是使用 MIT License 的开源程序库

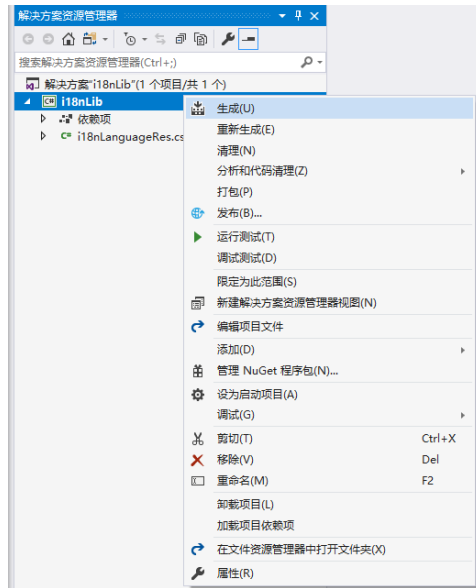
3. 开始

首先，我们需要从因特网上获取该库的一个副本，该项目的 Github 网址：<https://github.com/JiZimo/I18nLib>：



您可以通过该网址直接下载该项目的 zip 压缩包文件, 然后将其全部解压缩至您的工作目录, 打开 i18nLib.sln。配置管理器选择 Release





选择生成

然后就可以在 i18nLib-

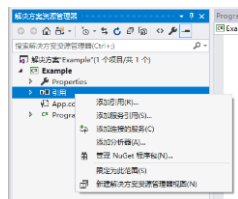
master\i18nLib\bin\Release\netstandard2.0 目录下找到 i18nLib.dll

即可使用该库

4. I18nLib 的使用

4.1 在项目中添加 I18nLib

首先，我们在 Visual Studio 中命名并创建一个 .net framework 项目或 .net core 项目，在这里我们将其命名为 Example，右击引用



选择添加引用-浏览，单击下面的浏览，在弹出的窗口

中选择我们刚刚生成的 i18nLib.dll 库。即可在您的项目中添加并使用 I18nLib 库。

4.2 创建 I18nLanguageRes 对象

I18nLanguageRes 对象是 I18nLib 库中的主要 API，通过创建这个对象便可使用 I18nLib 提供的多语言支持。首先，确定你要支持的语言，将语言名称放入一个 string[] 内，如下所示：

```
string[] languageList={"lang1","lang2","lang3"};
```

其中 lang1, lang2, lang3 为你想要支持的语言，这样的语言可以有无数个。然后我们添加如下一行来创建 I18nLanguageRes 对象：

```
I18nLanguageRes i18n=new I18nLanguageRes(languageList);
```

languageList 即为您创建的 string[] 名称

4.3 添加资源项

什么是资源项？

在 I18nLib 中，资源项指的是一组命名的不同语言的文本集合，是 I18nLib 进行硬编码多语言支持的重要部分。

如何添加资源项？

可以使用 AddItem() 成员函数来添加资源项，该函数有两个重载形式：

```
AddItem(string itemName,string[] text);  
AddItem(KeyValuePair<string,string[]> pair);
```

在这两个重载函数中，第一个参数为资源项的名称，第二个参数为资源项中的各个语言的内容。若未提供某一种语言的翻译文本，则将这一项留空（null）。

示例：

```
string[] greeting={"你好", "Hello"};
```

```
i18n.AddItem("Greeting",greeting);
```

该段代码向 i18n 这个 I18nLanguageRes 中添加了一个项名为"Greeting",内容为{"你好", "Hello"};的资源项。