

C#

跨设备/前后端
开发探索



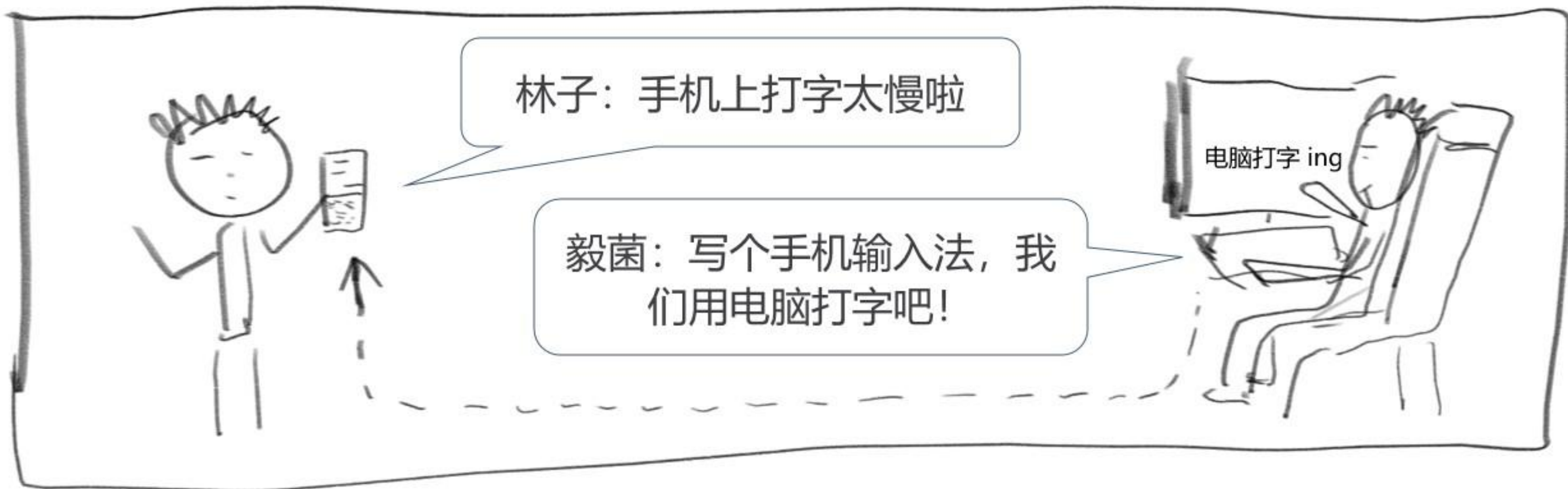
吕毅





CVTE
Dream Future

好奇心



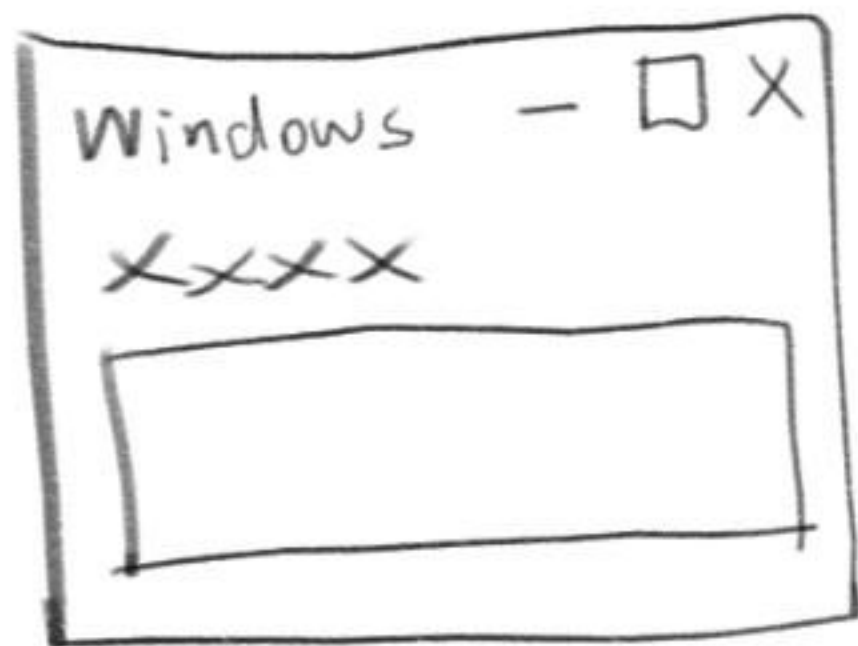


尝试

我们需要



Web 后端



桌面端



移动端



先掌握所有理论知识再实践

❄ 乏味的学习

无论什么技术，先搞起来

⚡ 成就感鼓励





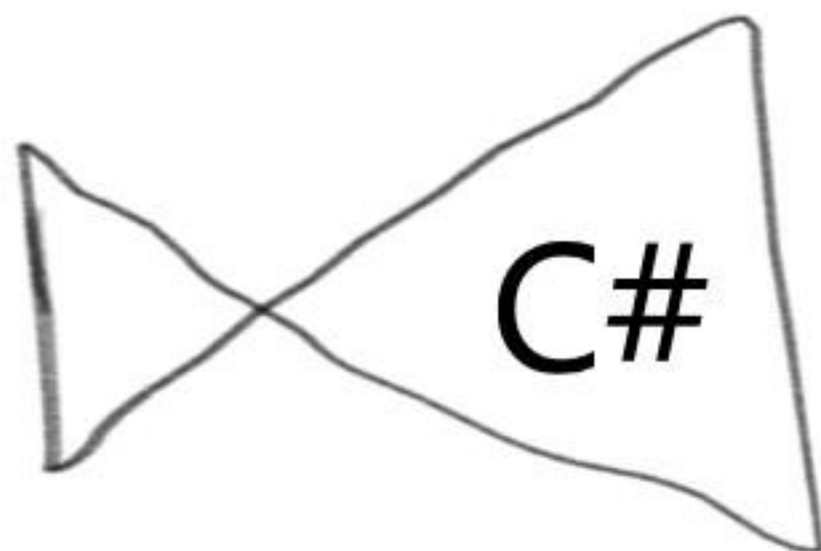
从~~入门~~到放弃





不是学，而是玩

降低学习成本，甚至入门不学习



毅菌：我们有 C#，还有什么不能马上搞起来！

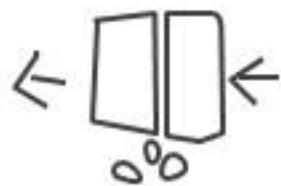
电 脑

搞起来

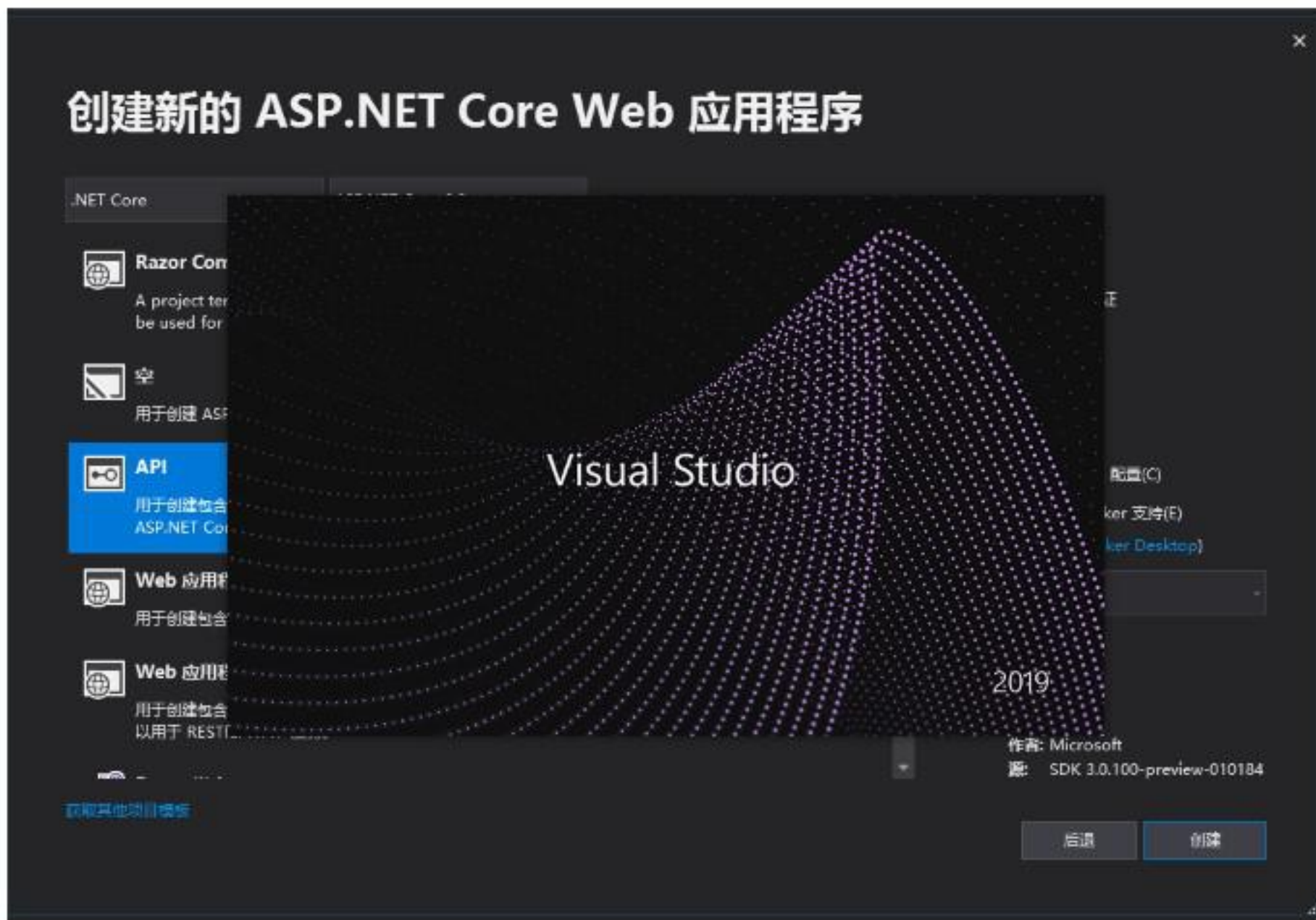
Visual Studio

2019





CVTE
Dream-Future





<https://localhost:5000/api/keyboard/dotnetclub>



```
// GET api/keyboard/5
/// <summary>
/// 获取指定 <paramref name="token"/> 下正在输入的文本。
/// 为了保持幂等性，即使输入操作结束并开始输入下一条文本，此操作也不会得到下一条输入的文本。
/// 这样，即使不断在浏览器中访问网址，也不会导致 App 中的获取失效。
/// 在 App 中请使用 POST 方法以便在消息上屏后可以清除上屏的消息并获取到下一条新消息。
/// </summary>
```

```
[HttpGet("{token}")]
```

1 个引用 | walterlv, 70 天前 | 1 名作者, 6 项更改 | 0 个请求 | 0 异常

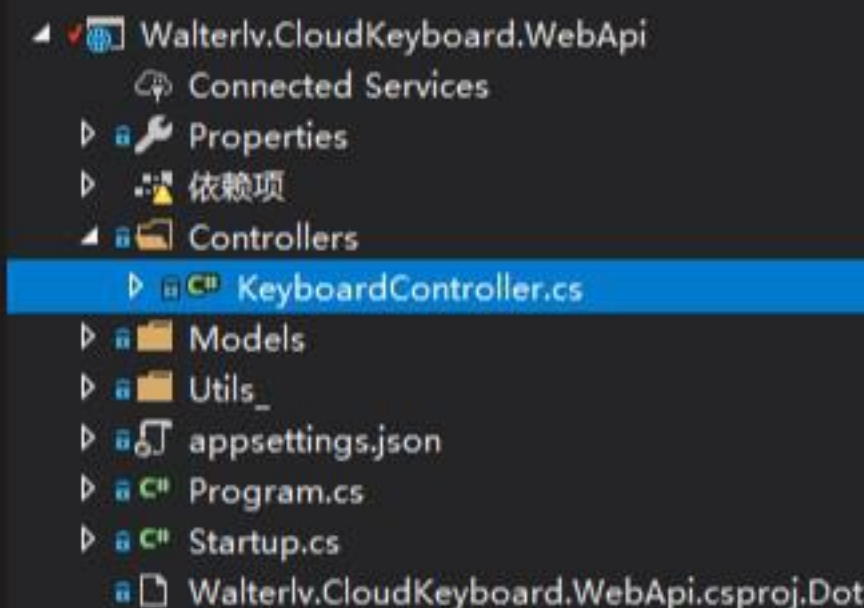
```
public ActionResult<TypingText> Get(string token)
```

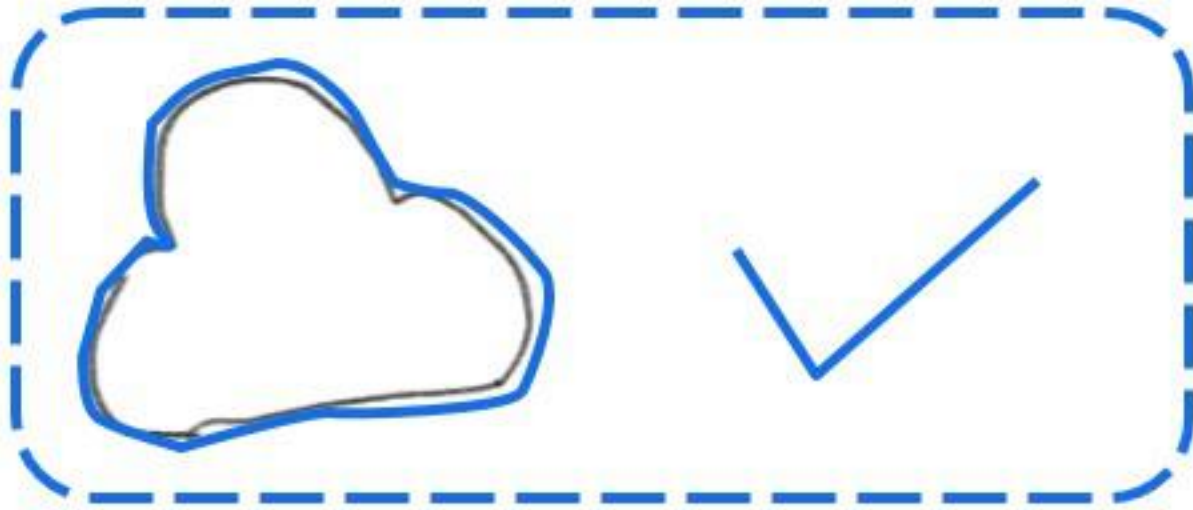
```
{
    var keyboard = _context.Keyboards.Find(token);
    if (keyboard == null)
    {
        return NotFound(new TypingResponse(false, $"Token {token} not found."));
    }

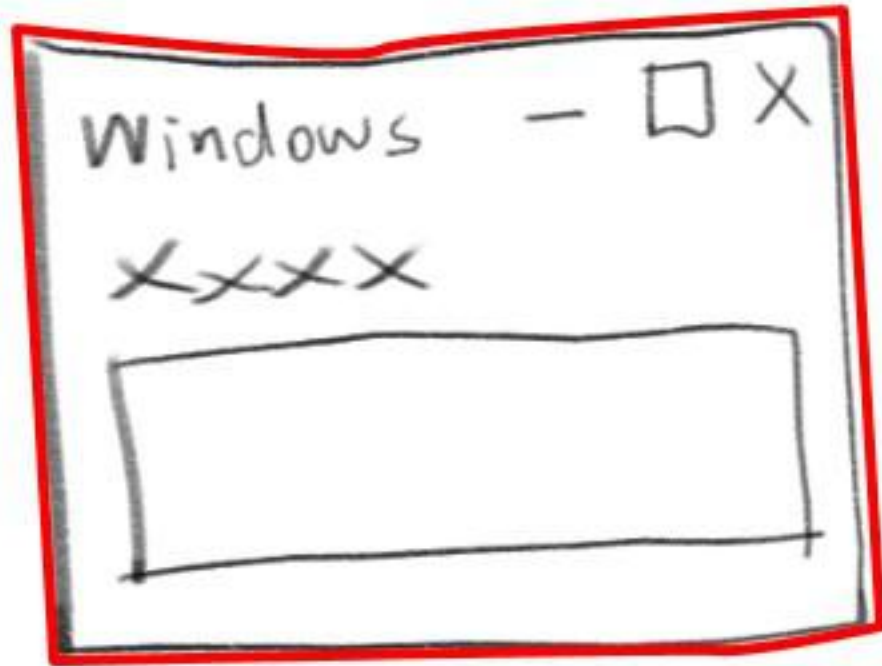
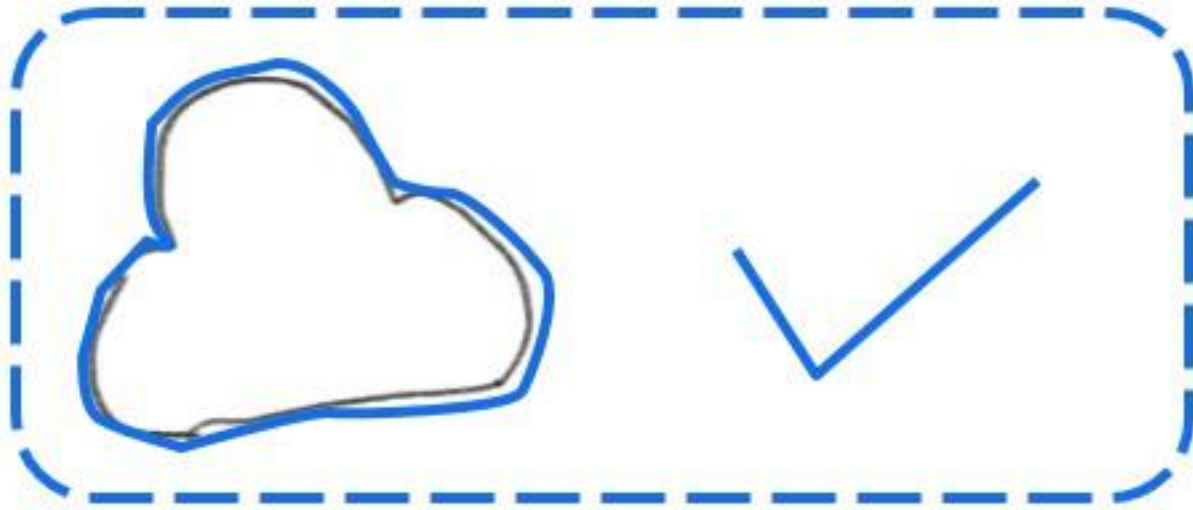
    var value = _context.Typings.FirstOrDefault(x => x.KeyboardToken == token);

    if (value == null)
    {
        return NotFound(new TypingResponse(false, $"Token {token} has no texts."));
    }

    return value.AsClient();
}
```





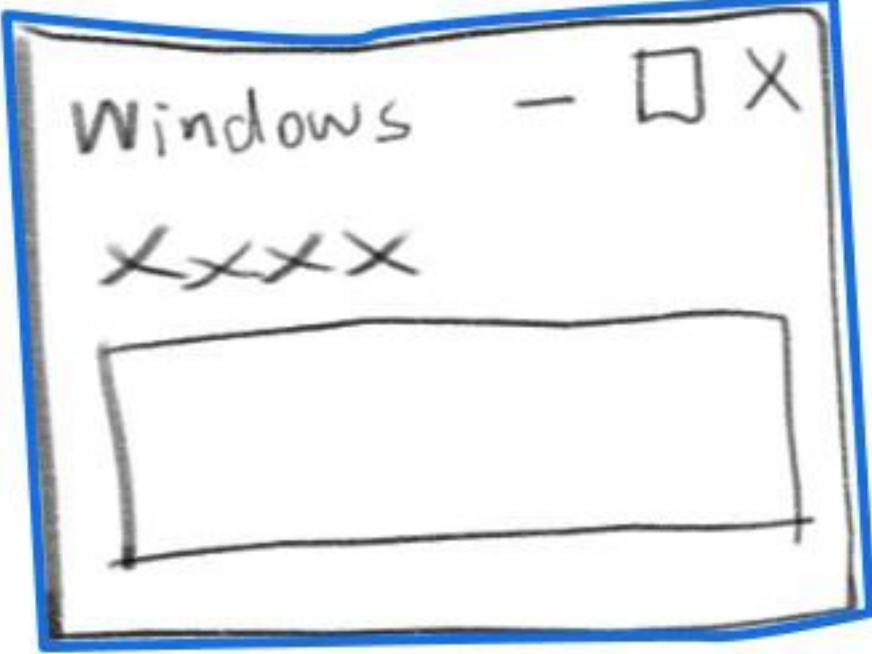
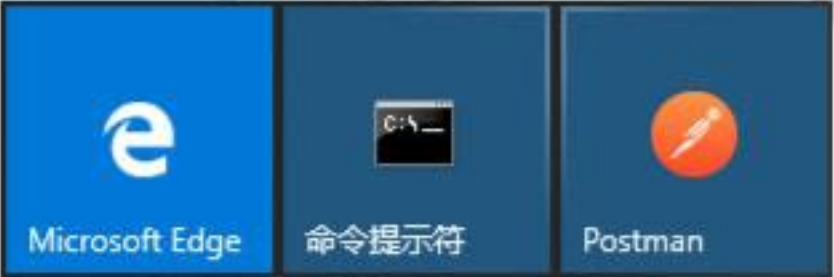


公共的类型/模型定义

.NET Standard 类库



辅助调试



添加新项目



语言

平台

项目类型

最近使用的项目模板



ASP.NET Core Web 应用程序

C#



控制台应用(.NET Core)

用于创建可在 Windows、Linux 和 MacOS 上的 .NET Core 上运行的命令行应用程序的项目。

C#

Linux

macOS

Windows

控制台



ASP.NET Core Web 应用程序

用于通过 .NET Core 或 .NET Framework 创建 Windows、Linux 和 macOS 版 ASP.NET Core 应用程序的项目模板。创建 Razor 页面、MVC、Web API 和单页(SPA)应用程序。

C#

Windows

Linux

macOS

Web



WPF 应用(.NET Framework)

Windows Presentation Foundation 客户端应用程序

C#

Windows

桌面



WPF App (.NET Core)

Windows Presentation Foundation 客户端应用程序

C#

Windows

桌面



类库(.NET Standard)

用于创建目标为 .NET Standard 的类库的项目。

C#

Android

iOS

Linux

macOS

Windows

库



Azure Functions

用于创建 Azure Function 项目的模板。

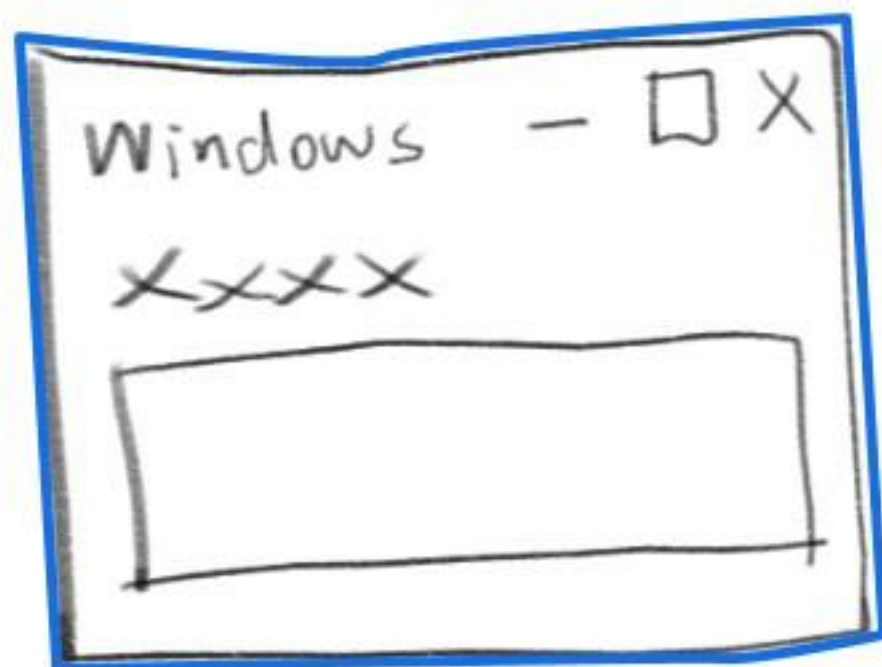
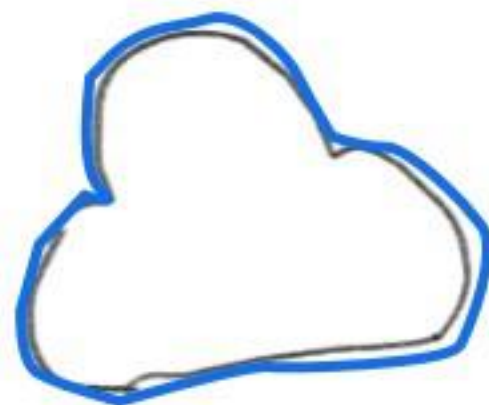
C#

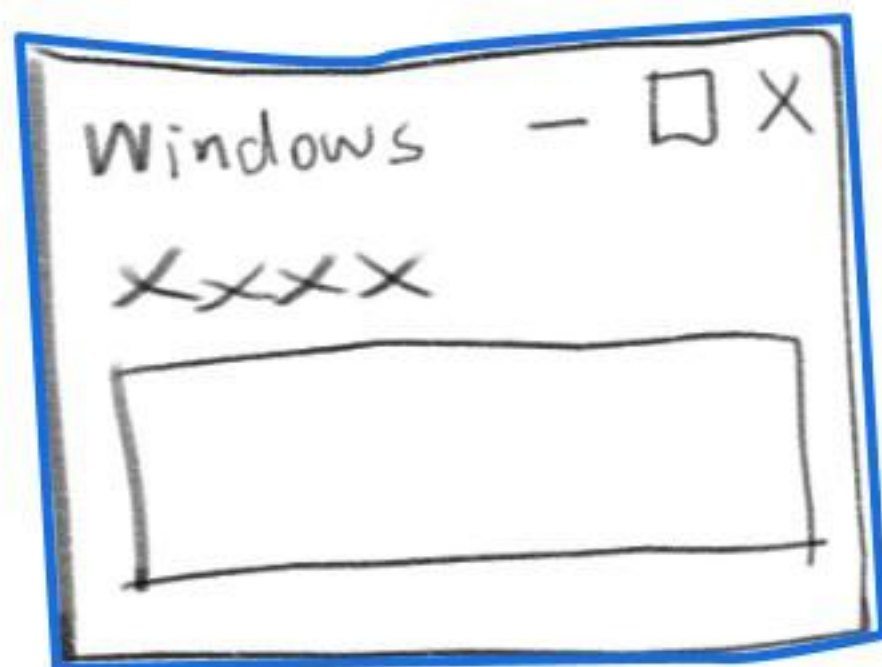
Azure

云

下一步(N)

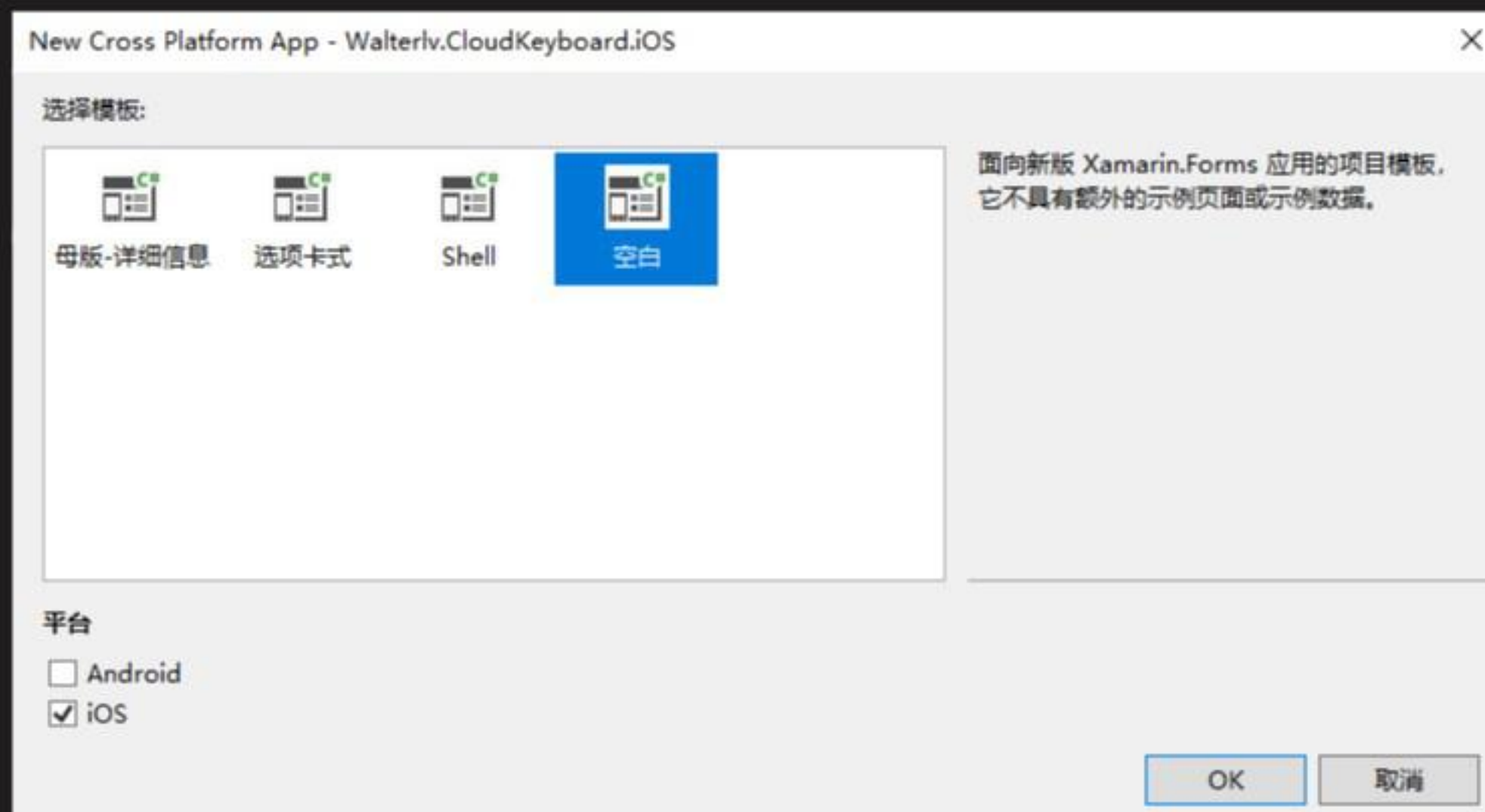
辅助调试





Xamarin





添加新项目

最近使用的项目模板



ASP.NET Core Web 应用程序

C#

搜索项目模板



语言

平台

项目类型

筛选依据: iOS

[清除筛选器](#)

Broadcast Upload 扩展(iOS)

创建 Broadcast Upload 扩展。该扩展可接收影片剪辑并将其发送给服务

C# iOS 移动



调用目录扩展(iOS)

创建可以将电话号码与姓名相关联或者告知系统应阻止号码的呼叫目录扩展

C# iOS 移动



自定义键盘扩展(iOS)

自定义键盘应用程序扩展。

C# iOS 移动



文档选取器扩展(iOS)

文档选取器应用程序扩展。

C# iOS 移动



Intents 扩展(iOS)

创建可接收系统意向对象并执行相关任务的 Intents 扩展

C# iOS 移动



Intents UI 扩展(iOS)

创建允许在成功处理意向后自定义 Siri 或 Maps 接口的 Intents UI 扩展

C# iOS 移动

下一步(N)

KeyboardViewController.cs

Walterlv.CloudKeyboard.iOS.Extension

Walterlv.CloudKeyboard.iOS.Extension.KeyboardViewController

ViewDidLoad()

```
10 32 public override void ViewDidLoad()
33 {
34     base.ViewDidLoad();
35
36     // Perform custom UI setup here
37     nextKeyboardButton = new UIButton(UIButtonType.System);
38
39     nextKeyboardButton.SetTitle("Next Keyboard", UIControlState.Normal);
40     nextKeyboardButton.SizeToFit();
41     nextKeyboardButton.TranslatesAutoresizingMaskIntoConstraints = false;
42     nextKeyboardButton.AddTarget(vent.TouchUpInside);
43
44     View.AddSubview(nextKeyboardButton);
45
46     var nextKeyboardButtonLeftSideConstraint = NSLayoutConstraint.Create(nextKeyboardButton, NSLayoutConstraintAttribute.Left, NSLayoutConstraintAttribute.Right, View, 0, 0);
47     var nextKeyboardButtonBottomConstraint = NSLayoutConstraint.Create(nextKeyboardButton, NSLayoutConstraintAttribute.Bottom, NSLayoutConstraintAttribute.Top, View, 0, 0);
48     View.AddConstraints(new[] { nextKeyboardButtonLeftSideConstraint, nextKeyboardButtonBottomConstraint });
49 }
50
51 0 个引用
10 52 public override void TextWillChange(UITextInput textInput)
53 {
54     // The app is about to change the document's contents. Perform any preparation here.
55 }
56
57 0 个引用
10 57 public override void TextDidChange(UITextInput textInput)
58 {
59     // The app has just changed the document's contents, the document context has been updated.
60     UIColor textColor = null;
61
62     if (TextDocumentProxy.KeyboardAppearance == UIKeyboardAppearance.Dark)
```

CVTE

解决方案"Walterlv.CloudKeyboard" (6 个项目)

Walterlv.CloudKeyboard.Cli

Walterlv.CloudKeyboard.Core

Walterlv.CloudKeyboard.iOS

Connected Services

Properties

引用

Asset Catalogs

本机引用

Resources

AppDelegate.cs

Entitlements.plist

Info.plist

Main.cs

Walterlv.CloudKeyboard.iOS.Extension

Properties

引用

Asset Catalogs

本机引用

AppDelegate.cs

Entitlements.plist

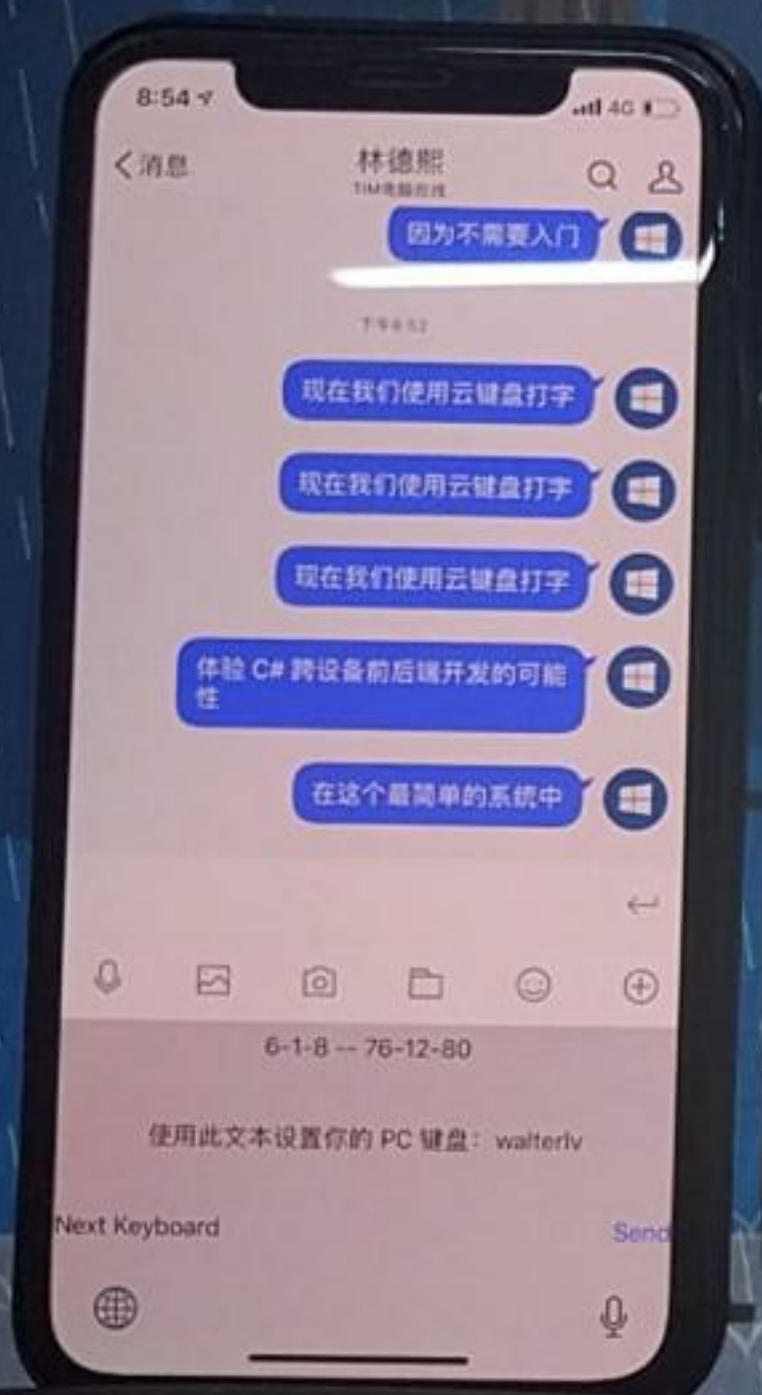
Info.plist

KeyboardViewController.cs

Main.cs

Walterlv.CloudKeyboard.WebApi

Walterlv.CloudKeyboard.Xaml



云键盘 - walterlv

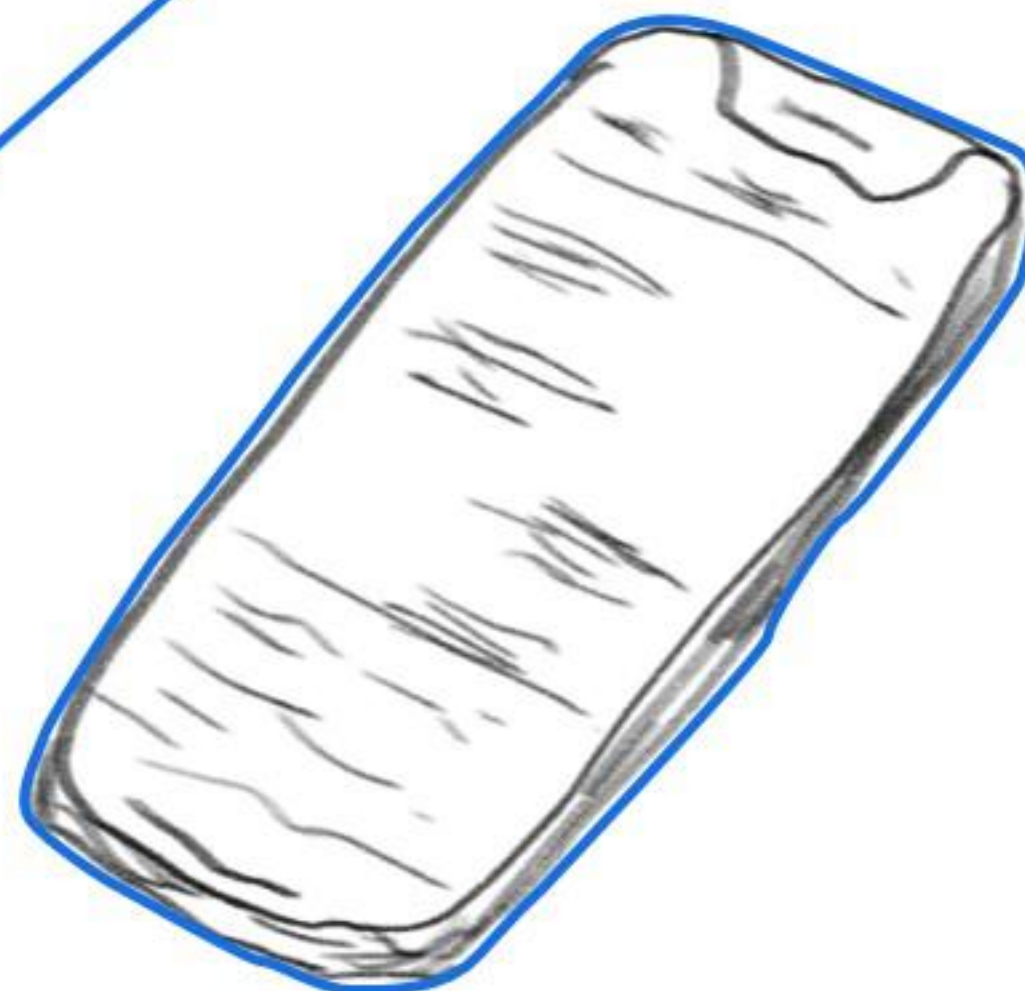
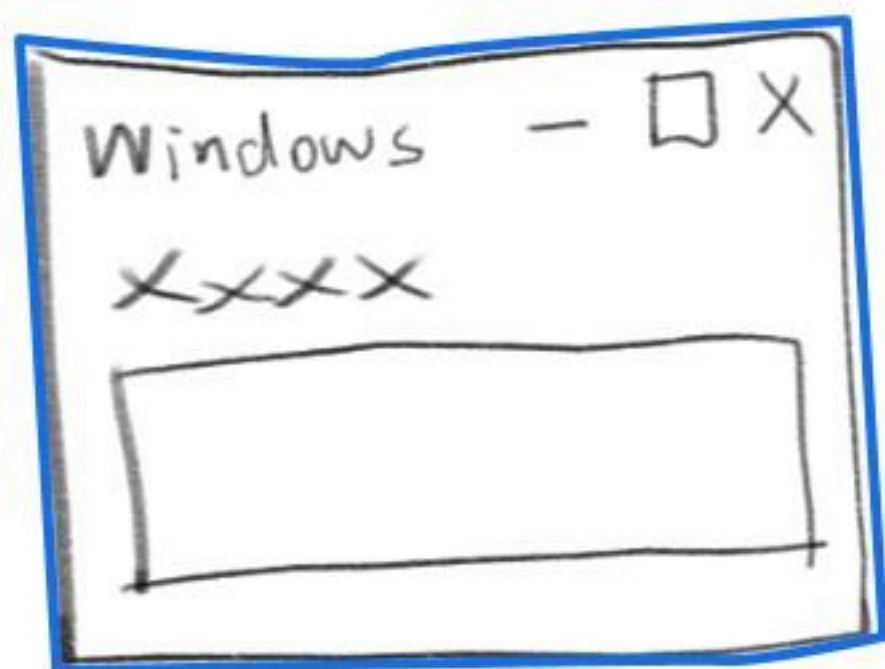
walterlv 的云键盘

请在你的 iPhone 中将设备切换至 Cloud 输入法

在这里打字，将实时同步到你的 iPhone.....

按下 Enter 确认，按下 Ctrl+Enter 输入换行符。

确认





毅菌：这么多客户端，
怎么分发给大家用呀？

林子：做个前端？

毅菌：我前端烂，不
会调样式

林子：先搞起来？



CVTE
Dream·Future



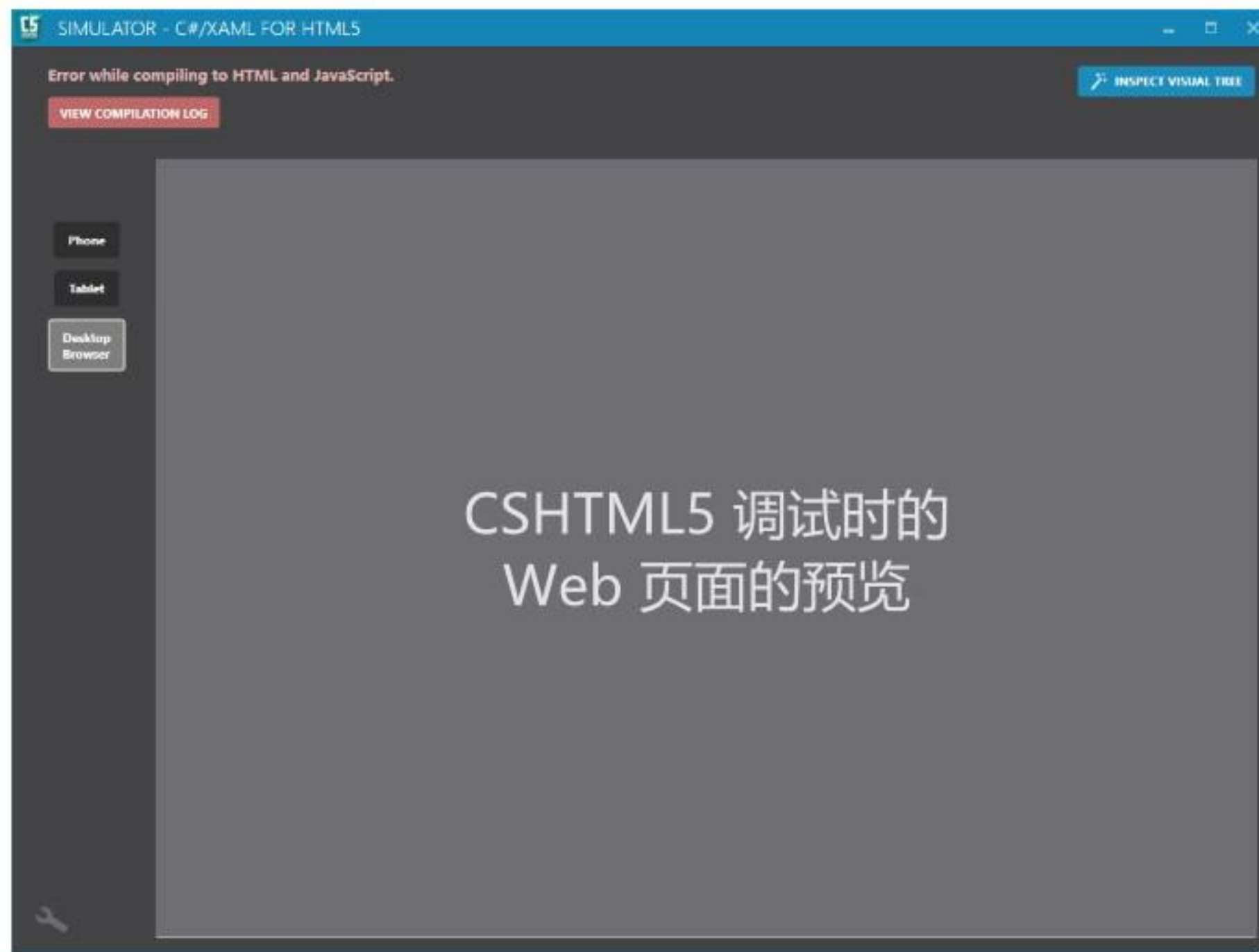
@MV10：用 CSHTML5



@MV10

XAML
+ C#

$$C\# + XAML \xRightarrow[\triangle]{CSHTML5} HTML5 + JS$$





C#

Asp .NET Core

CVTE

桌面端

XAML + C#

.NET Fx

移动端

XAML + C#

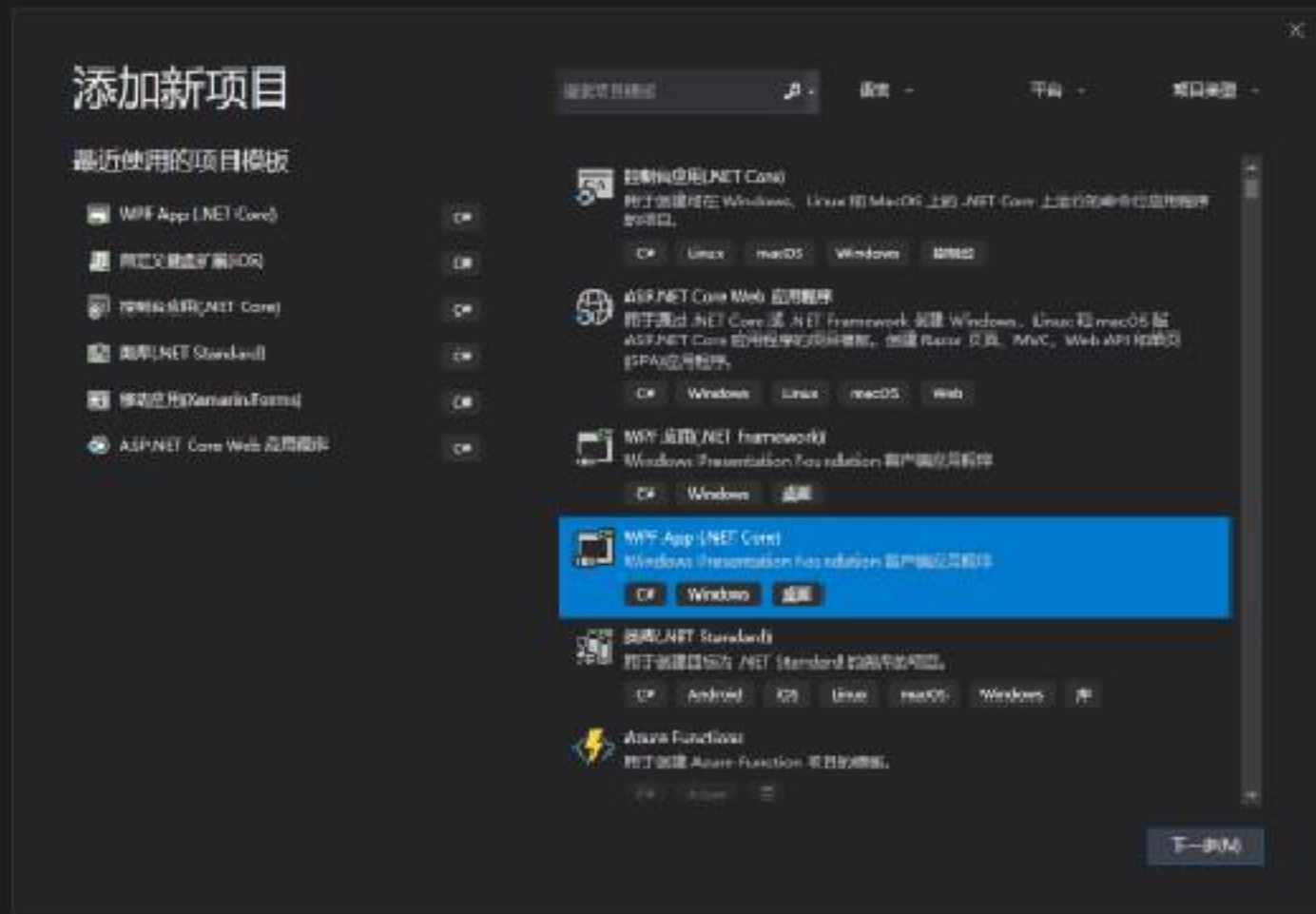
Xamarin

Web 前端

XAML + C#

CSHTML5

Windows App on .NET Core



WPF on .NET Core

```
<Project Sdk="Microsoft.NET.Sdk.WindowsDesktop">  
  <PropertyGroup>  
    <OutputType>Exe</OutputType>  
    <TargetFramework>netcoreapp3.0</TargetFramework>  
    <UseWPF>true</UseWPF>  
  </PropertyGroup>  
</Project>
```


WPF & Windows Forms on .NET Core

```
<Project Sdk="Microsoft.NET.Sdk.WindowsDesktop">  
  <PropertyGroup>  
    <OutputType>Exe</OutputType>  
    <TargetFramework>netcoreapp3.0</TargetFramework>  
    <UseWPF>>true</UseWPF>  
    <UseWindowsForms>>true</UseWindowsForms>  
  </PropertyGroup>  
</Project>
```



将旧 csproj 迁移成 SDK 风格的 csproj

迁移
博客



将 WPF 项目迁移到 .NET Core 3

为什么将 Windows 桌面应用程序迁移到 .NET Core?



.NET Core 是 .NET 家族中最具创新性的一个
多种部署方式：

- Side-by-side 支持
- 计算机全局或应用本地框架
- 自包含的 exe

运行时、基础类库和新的语言特性
性能提升

专门针对桌面应用的提升

UI 框架上的改进

- XAML Islands 和 XAML Controls

可以直接访问全部 Windows 10 API

希沃白板

.NET Core 3 - 已同步

云课件

同步

开始授课

文本

形状

多媒体

表格

课堂活动

思维导图

函数

学科工具

属性

动画

+ 新建页面

1

2

3

4

5

6

7

8

9

Index 和 Range 类型

```
Index i1 = 3; // number 3 from beginning
Index i2 = ^4; // number 4 from end
int[] a = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 };
Console.WriteLine($"{a[i1]}, {a[i2]}"); // "3, 6"

var slice = a[i1..i2]; // { 3, 4, 5 }
```

布局与背景

空白

更改布局

背景

更多背景

本地图片

应用主题

课件第 1 页, 共 26 页

设置投影 | 备注 | 教室 | 开始授课 | 16:9 | 63%

迁移可能需要用到的 NuGet 包



Microsoft.Windows.Compatibility

Microsoft.DotNet.Analyzers.Compatibility

控制台

C#



C#

Asp .NET Core

桌面端

XAML + C#

.NET Core 3

移动端

XAML + C#

Xamarin

Web 前端

XAML + C#

CSHTML5

没有 GUI，更容易跨平台

跨平台

XAML + C#

Avalonia

跨平台

XAML + C#

Xamarin

XAML 编译+Roslyn 分析+VS 插件
XAML + C# 跨平台

C# 还能做?

ML.NET
Roslyn
IoT

.....

C#

跨设备/前后端
开发探索



吕毅



扫一扫
获取 希沃白板 5 课件



人月神话中说：

两个人的效率不会是一个人的两倍，因为还有沟通成本。如果团队成员更多，那么沟通成本会更高。

像这种级别的小项目，我们自己做就好了，不用劳烦一个团队。