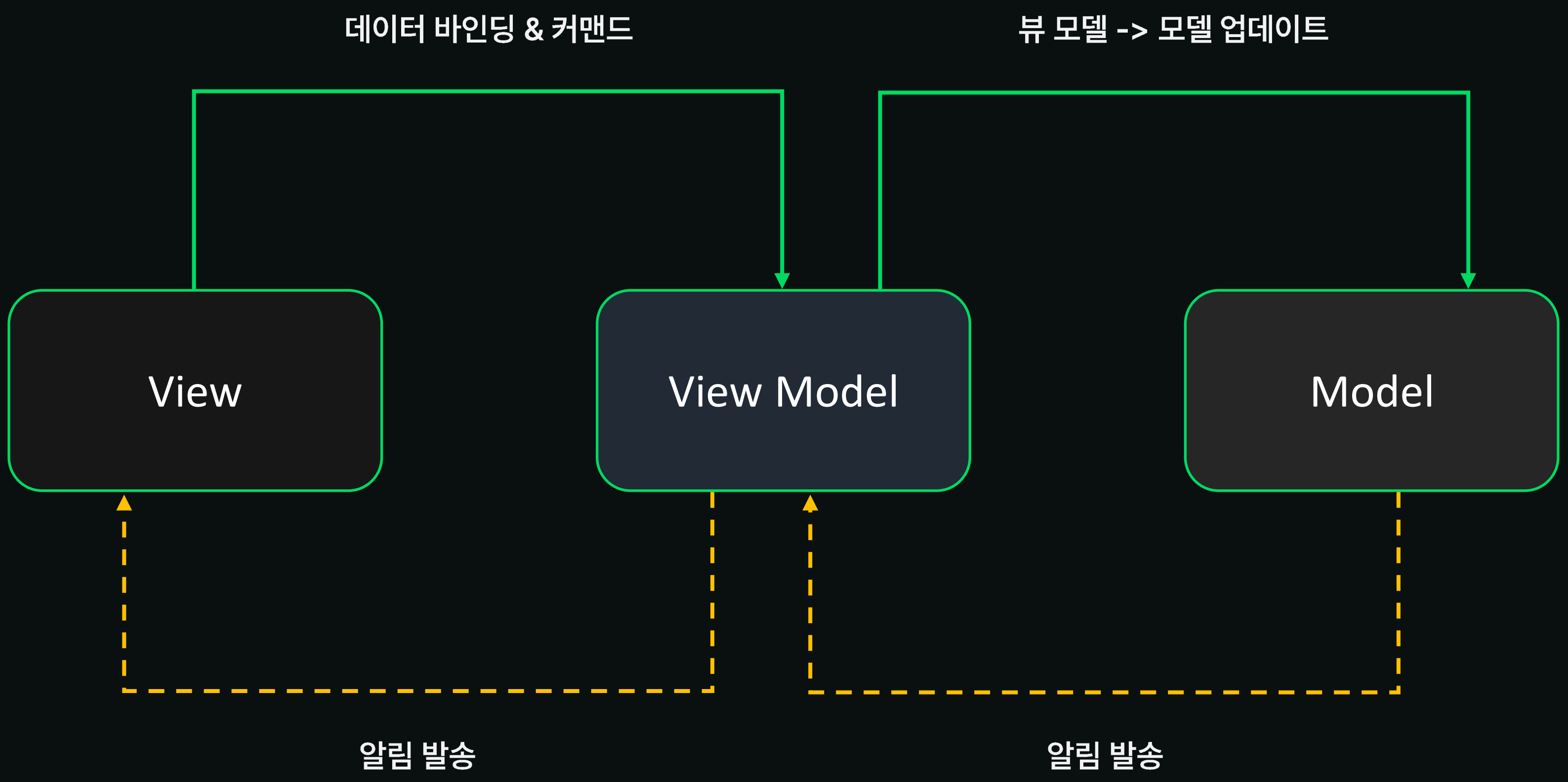
MauiReactor
실무 꿀팁 대방출

흑우마스터
jojangwon@kakao.com

MVVM Pattern



데이터 바인딩 & 커맨드

뷰 모델 -> 모델 업데이트

View

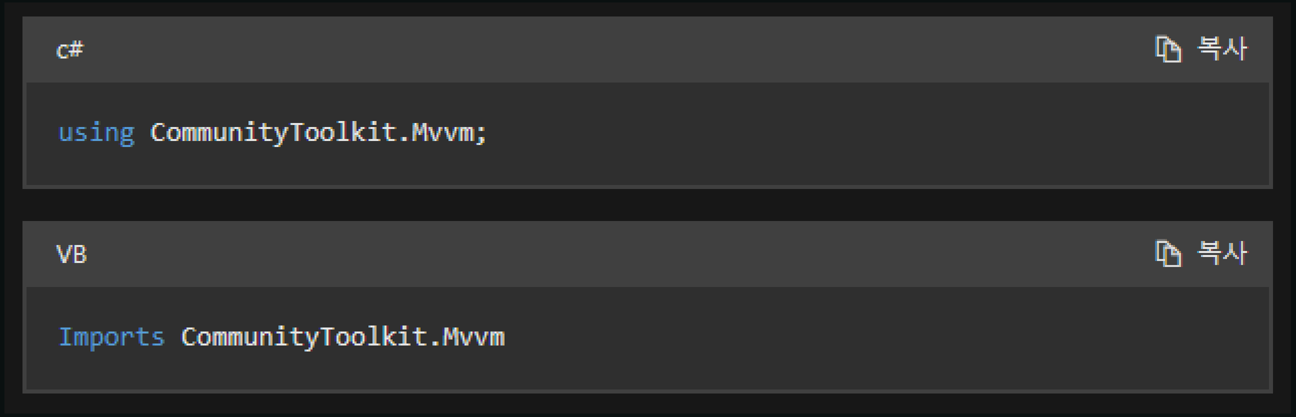
View Model

Model

알림 발송

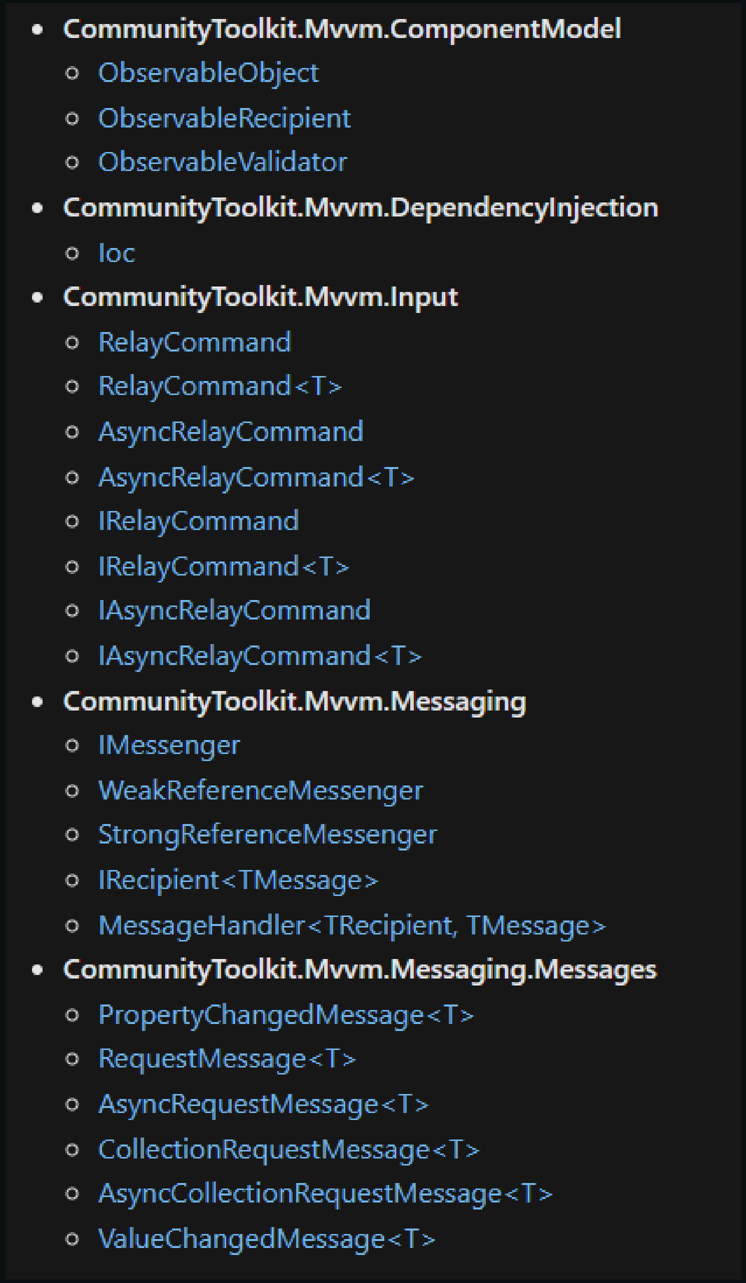
알림 발송

MVVM Pattern



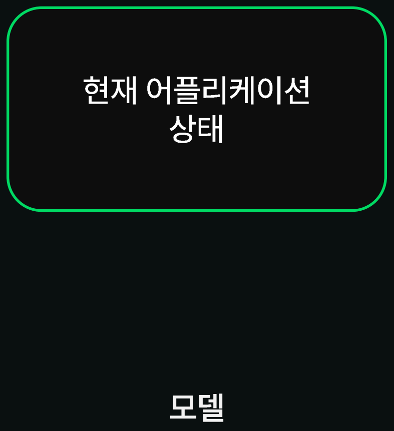
```csharp
using CommunityToolkit.Mvvm;
```

```vb
Imports CommunityToolkit.Mvvm
```



- **CommunityToolkit.Mvvm.ComponentModel**
  - ObservableObject
  - ObservableRecipient
  - ObservableValidator
- **CommunityToolkit.Mvvm.DependencyInjection**
  - Ioc
- **CommunityToolkit.Mvvm.Input**
  - RelayCommand
  - RelayCommand<T>
  - AsyncRelayCommand
  - AsyncRelayCommand<T>
  - IRelayCommand
  - IRelayCommand<T>
  - IAsyncRelayCommand
  - IAsyncRelayCommand<T>
- **CommunityToolkit.Mvvm.Messaging**
  - IMessenger
  - WeakReferenceMessenger
  - StrongReferenceMessenger
  - IRecipient<TMessage>
  - MessageHandler<TRecipient, TMessage>
- **CommunityToolkit.Mvvm.Messaging.Messages**
  - PropertyChangedMessage<T>
  - RequestMessage<T>
  - AsyncRequestMessage<T>
  - CollectionRequestMessage<T>
  - AsyncCollectionRequestMessage<T>
  - ValueChangedMessage<T>

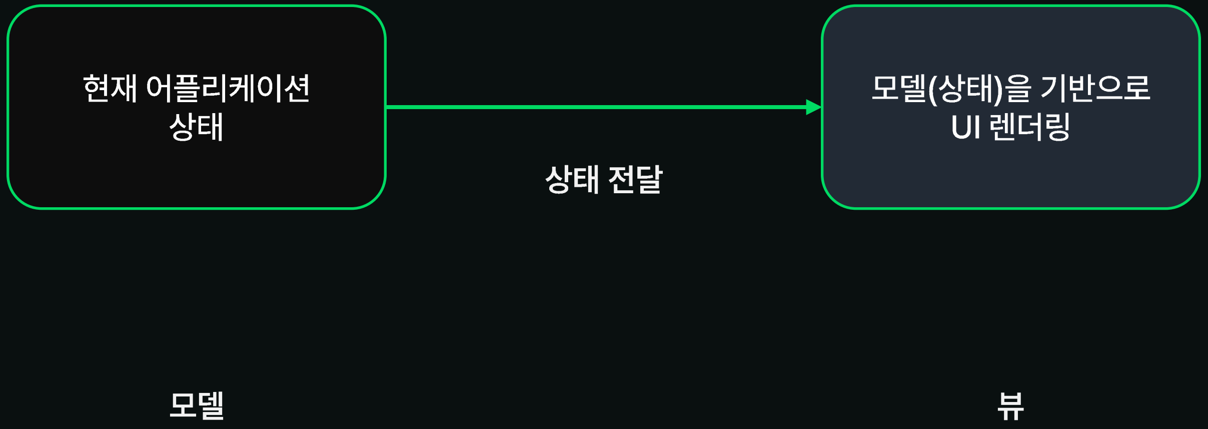
MVU Pattern



현재 어플리케이션
상태

모델

MVU Pattern



현재 어플리케이션 상태

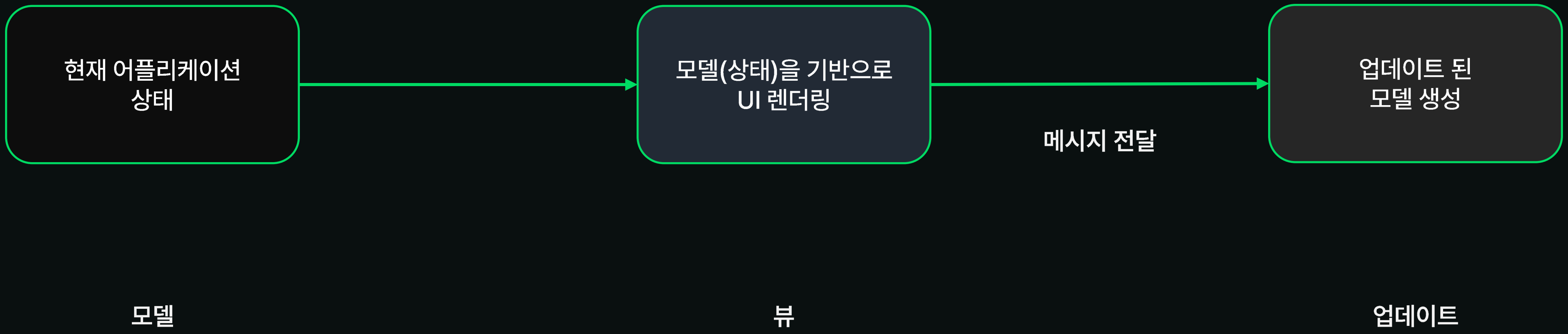
상태 전달

모델(상태)을 기반으로 UI 렌더링

모델

뷰

MVU Pattern



현재 어플리케이션 상태

모델(상태)을 기반으로 UI 렌더링

메시지 전달

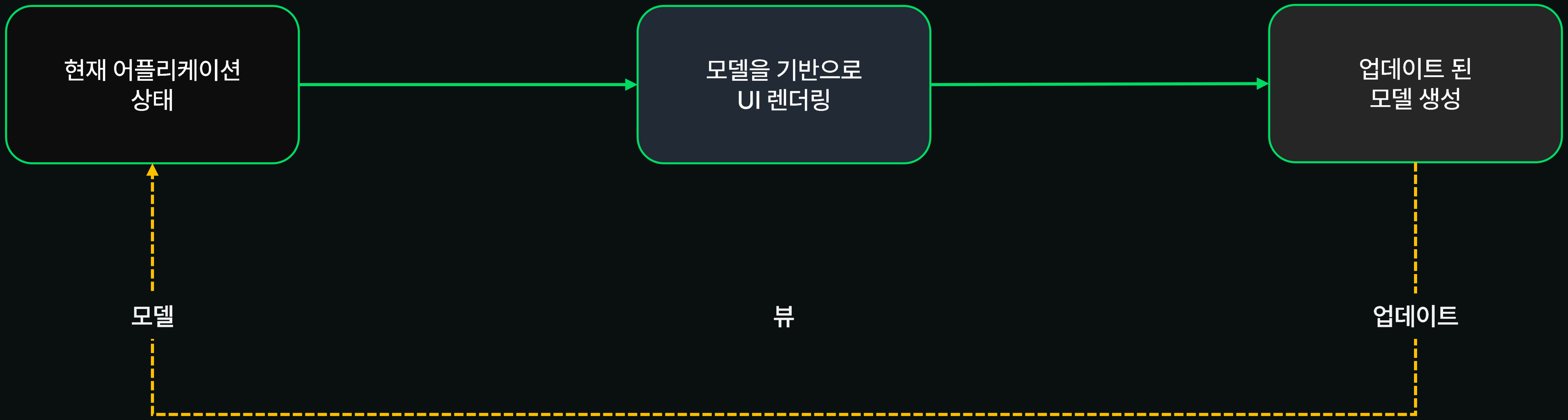
업데이트 된 모델 생성

모델

뷰

업데이트

MVU Pattern



현재 어플리케이션 상태

모델을 기반으로 UI 렌더링

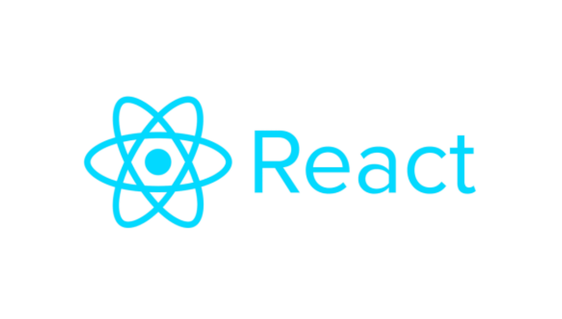
업데이트 된 모델 생성

모델

뷰

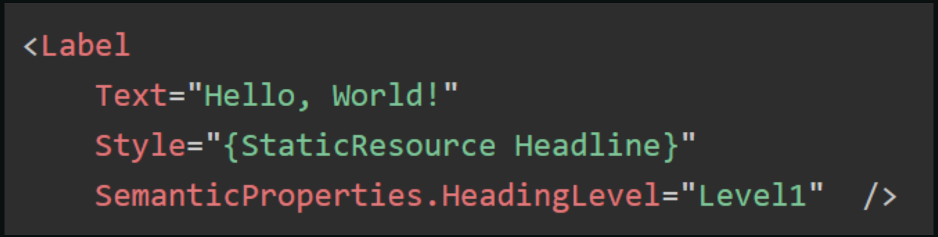
업데이트

MVU Pattern

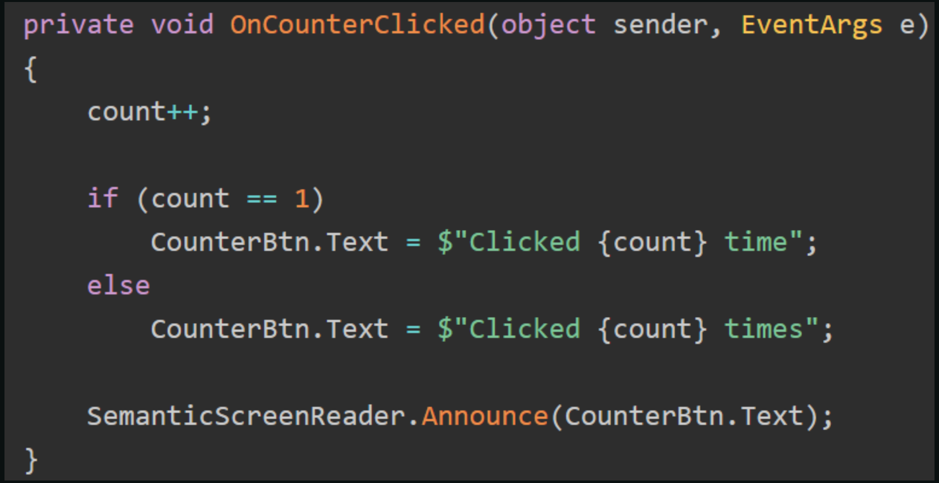


Better apps. Less code.

MVVM vs MVU

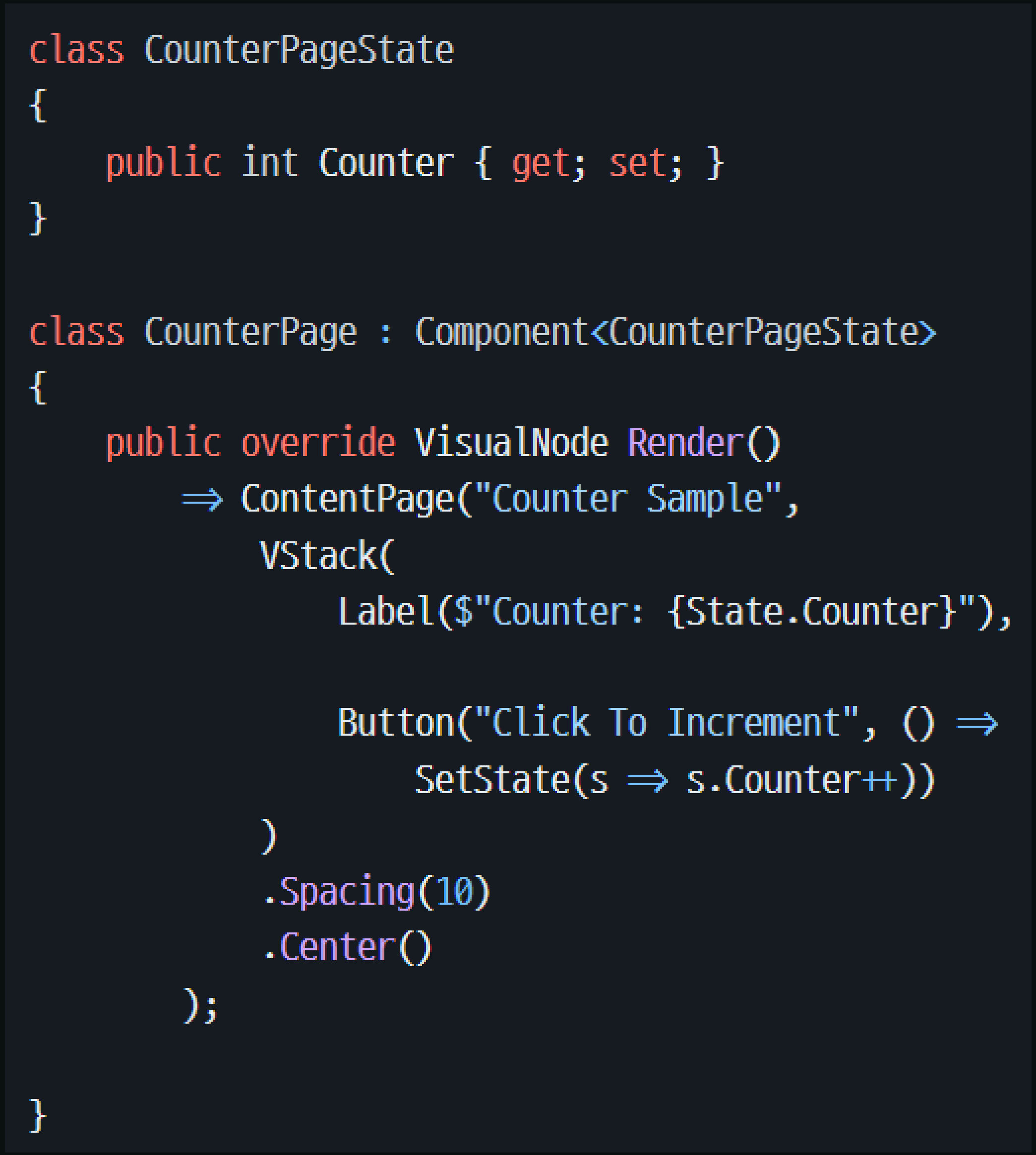


```xml
<Label
    Text="Hello, World!"
    Style="{StaticResource Headline}"
    SemanticProperties.HeadingLevel="Level1"  />
```



```csharp
private void OnCounterClicked(object sender, EventArgs e)
{
    count++;

    if (count == 1)
        CounterBtn.Text = $"Clicked {count} time";
    else
        CounterBtn.Text = $"Clicked {count} times";

    SemanticScreenReader.Announce(CounterBtn.Text);
}
```

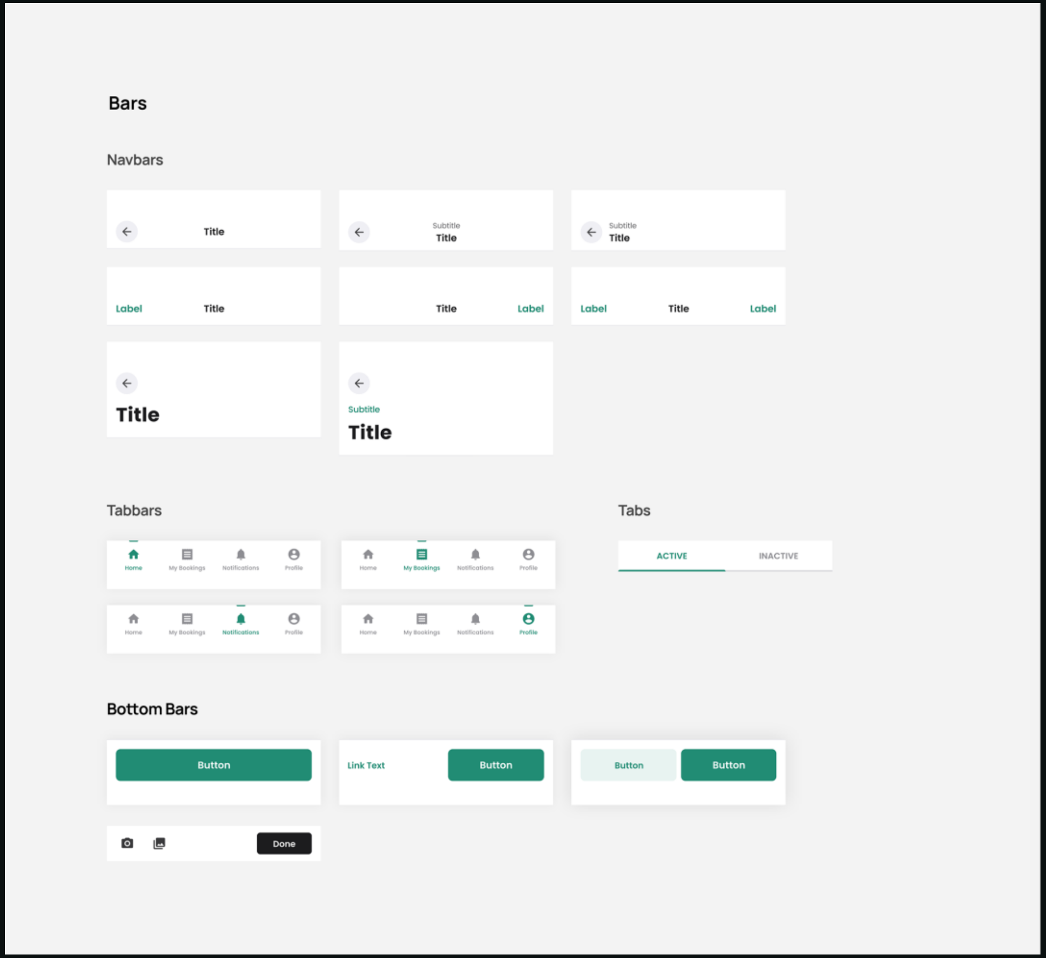
MVVM

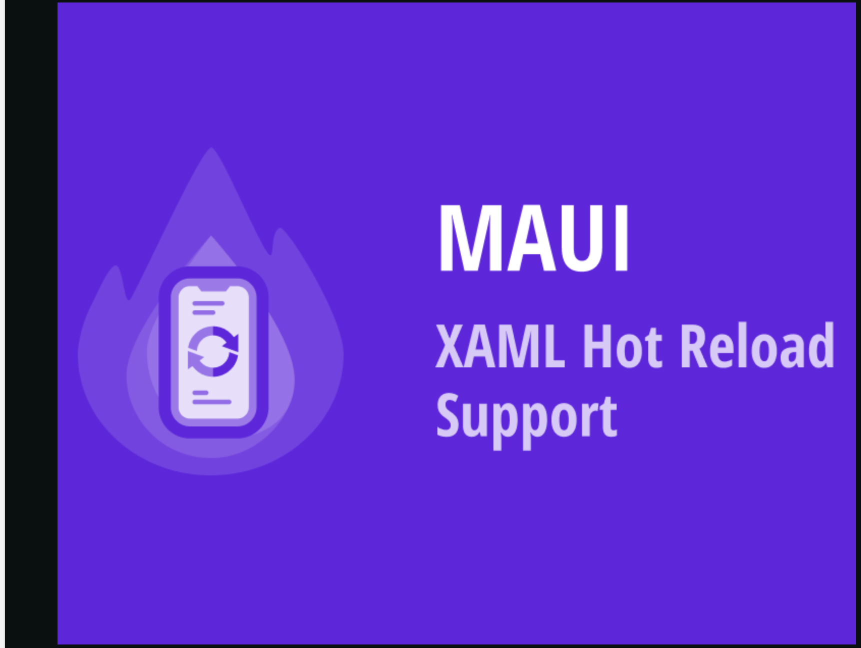


```csharp
class CounterPageState
{
    public int Counter { get; set; }
}

class CounterPage : Component<CounterPageState>
{
    public override VisualNode Render()
        => ContentPage("Counter Sample",
            VStack(
                Label($"Counter: {State.Counter}"),

                Button("Click To Increment", () =>
                    SetState(s => s.Counter++))
            )
            .Spacing(10)
            .Center()
        );

}
```

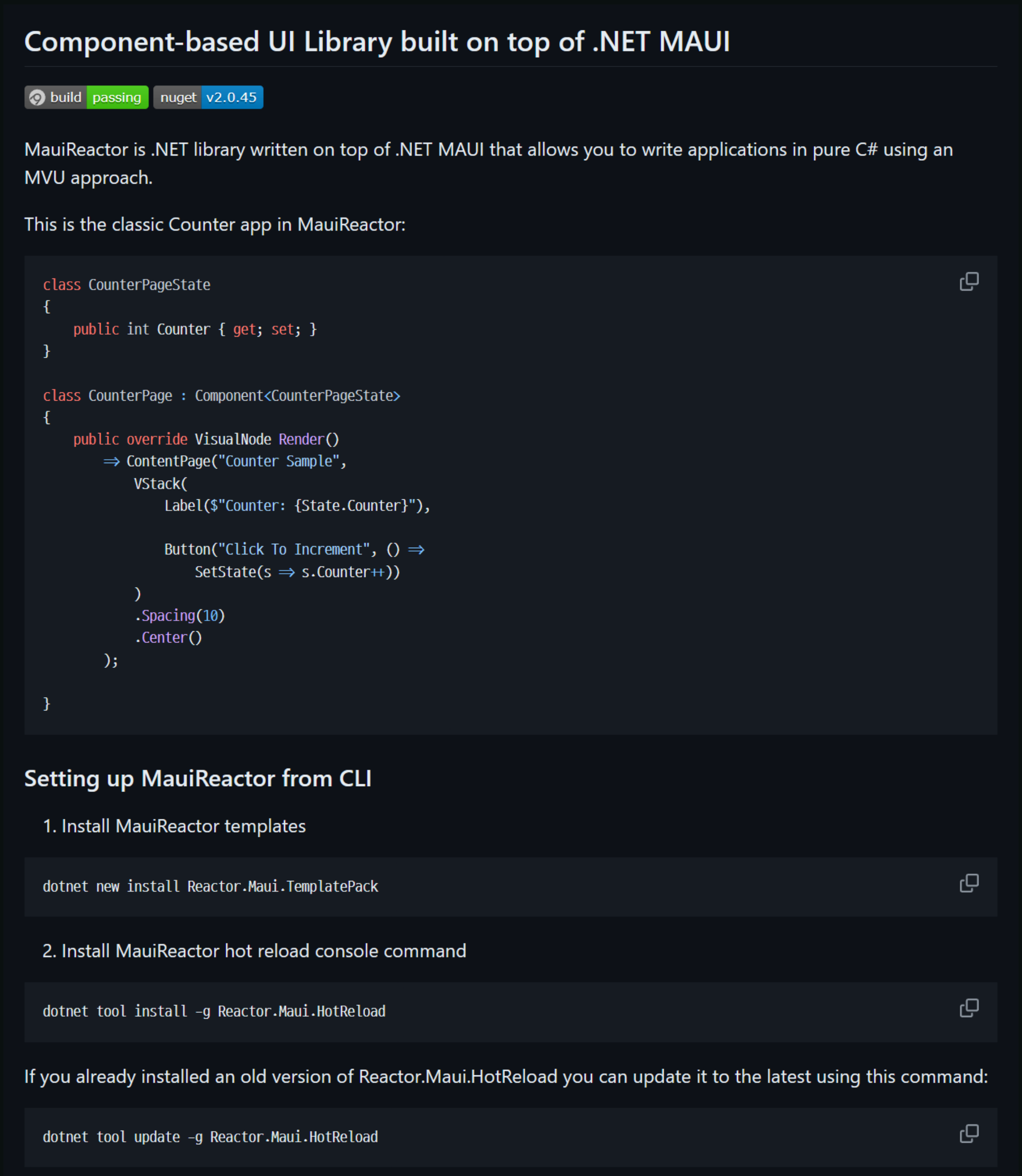
MVU

써보니 MVVM보단 MVU가 괜찮았습니다





MauiReactor



Component-based UI Library built on top of .NET MAUI

MauiReactor is .NET library written on top of .NET MAUI that allows you to write applications in pure C# using an MVU approach.

This is the classic Counter app in MauiReactor:

```
class CounterPageState
{
    public int Counter { get; set; }
}

class CounterPage : Component<CounterPageState>
{
    public override VisualNode Render()
        => ContentPage("Counter Sample",
            VStack(
                Label($"Counter: {State.Counter}"),

                Button("Click To Increment", () =>
                    SetState(s => s.Counter++))
            )
            .Spacing(10)
            .Center()
        );

}
```

Setting up MauiReactor from CLI

1. Install MauiReactor templates

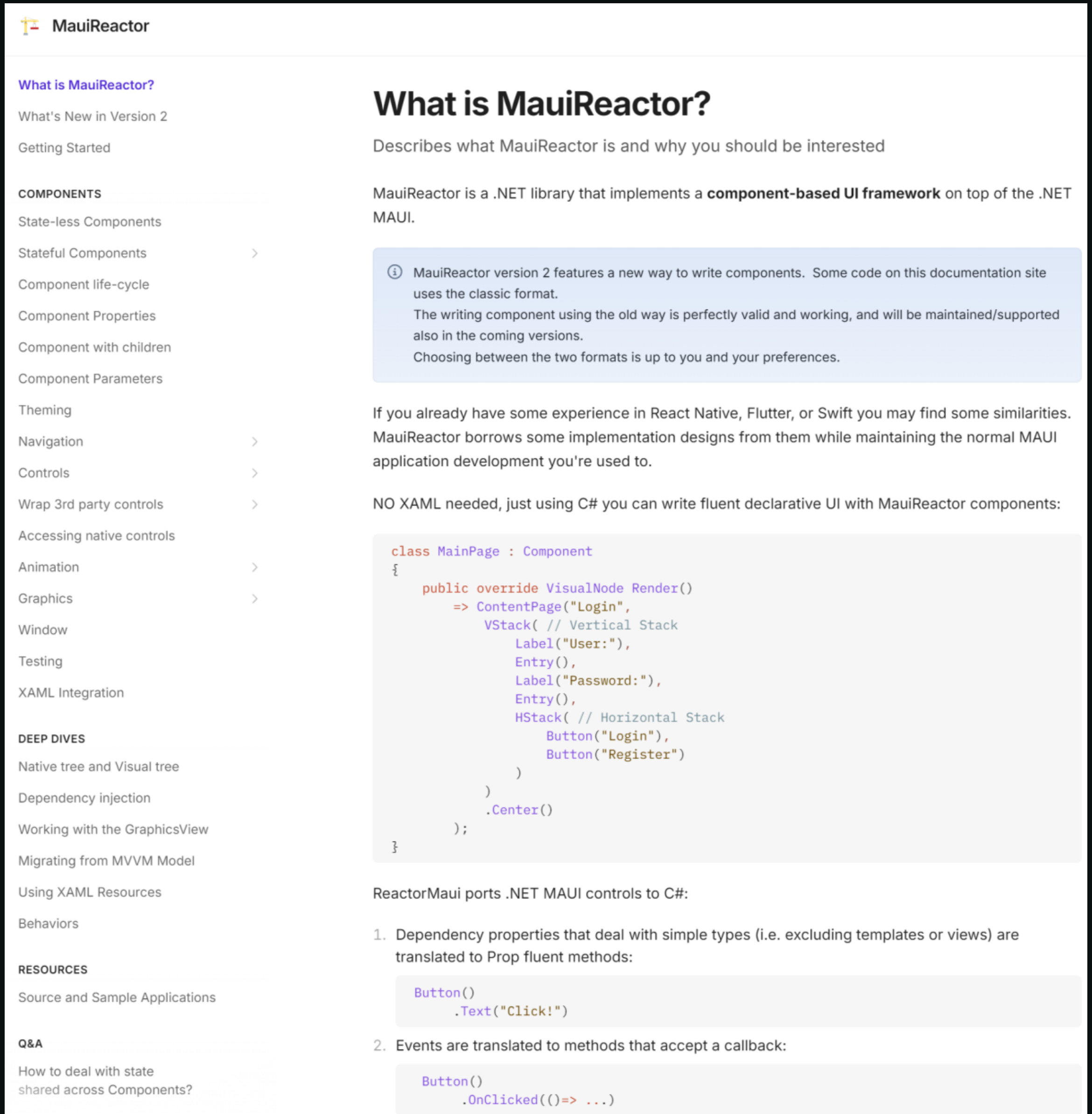
```
dotnet new install Reactor.Maui.TemplatePack
```

2. Install MauiReactor hot reload console command

```
dotnet tool install -g Reactor.Maui.HotReload
```

If you already installed an old version of Reactor.Maui.HotReload you can update it to the latest using this command:

```
dotnet tool update -g Reactor.Maui.HotReload
```



What is MauiReactor?

Describes what MauiReactor is and why you should be interested

MauiReactor is a .NET library that implements a **component-based UI framework** on top of the .NET MAUI.

> MauiReactor version 2 features a new way to write components. Some code on this documentation site uses the classic format.
> The writing component using the old way is perfectly valid and working, and will be maintained/supported also in the coming versions.
> Choosing between the two formats is up to you and your preferences.

If you already have some experience in React Native, Flutter, or Swift you may find some similarities. MauiReactor borrows some implementation designs from them while maintaining the normal MAUI application development you're used to.

NO XAML needed, just using C# you can write fluent declarative UI with MauiReactor components:

```
class MainPage : Component
{
    public override VisualNode Render()
        => ContentPage("Login",
            VStack( // Vertical Stack
                Label("User:"),
                Entry(),
                Label("Password:"),
                Entry(),
                HStack( // Horizontal Stack
                    Button("Login"),
                    Button("Register")
                )
            )
            .Center()
        );
}
```

ReactorMaui ports .NET MAUI controls to C#:

1. Dependency properties that deal with simple types (i.e. excluding templates or views) are translated to Prop fluent methods:

```
Button()
    .Text("Click!")
```

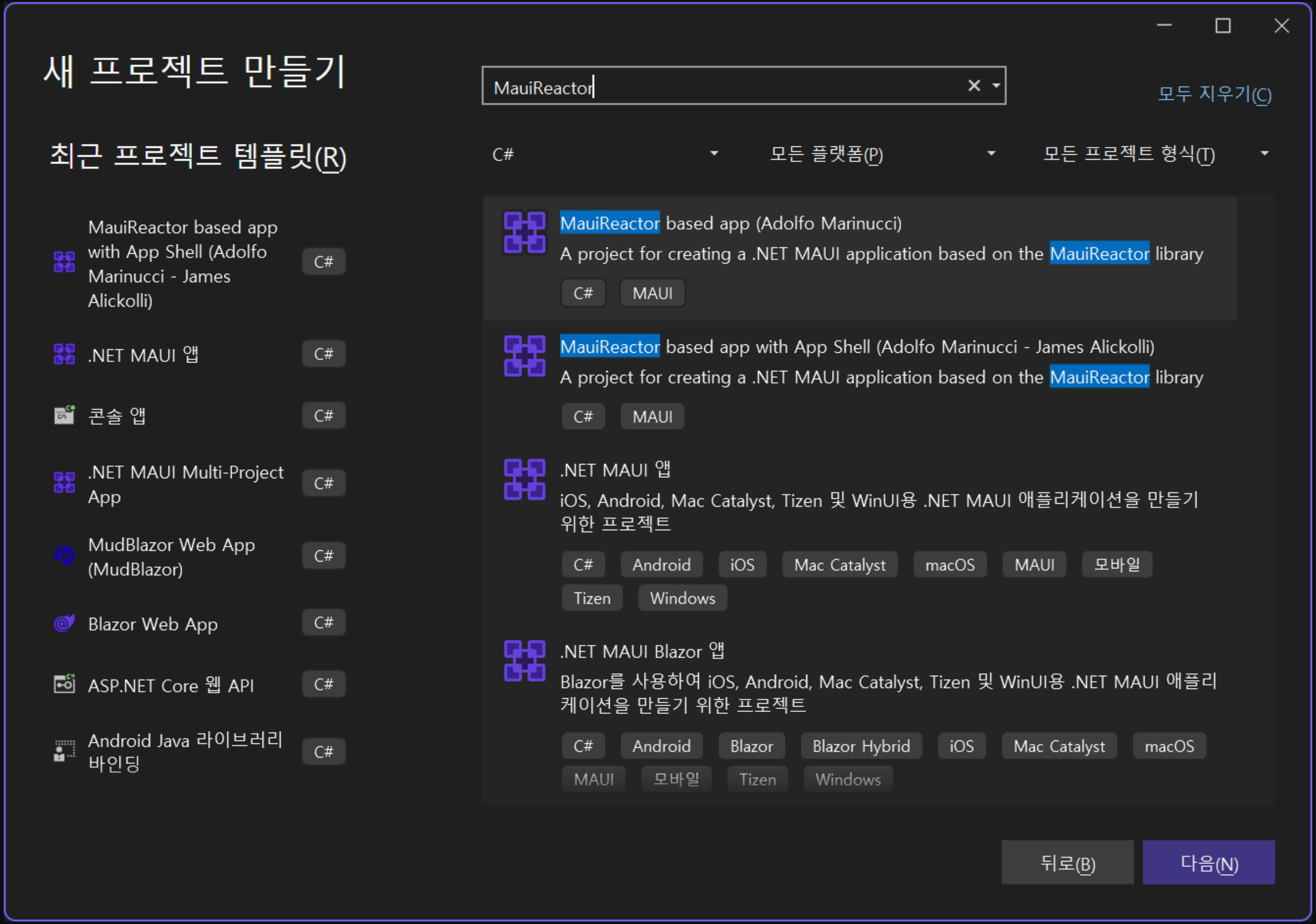
2. Events are translated to methods that accept a callback:

```
Button()
    .OnClicked(()=> ...)
```

https://github.com/adospace/reactorui-maui

https://adospace.gitbook.io/mauireactor

MauiReactor



새 프로젝트 만들기

MauiReactor

모두 지우기(C)

최근 프로젝트 템플릿(R)

C# · 모든 플랫폼(P) · 모든 프로젝트 형식(T)

- MauiReactor based app with App Shell (Adolfo Marinucci - James Alickolli) — C#
- .NET MAUI 앱 — C#
- 콘솔 앱 — C#
- .NET MAUI Multi-Project App — C#
- MudBlazor Web App (MudBlazor) — C#
- Blazor Web App — C#
- ASP.NET Core 웹 API — C#
- Android Java 라이브러리 바인딩 — C#

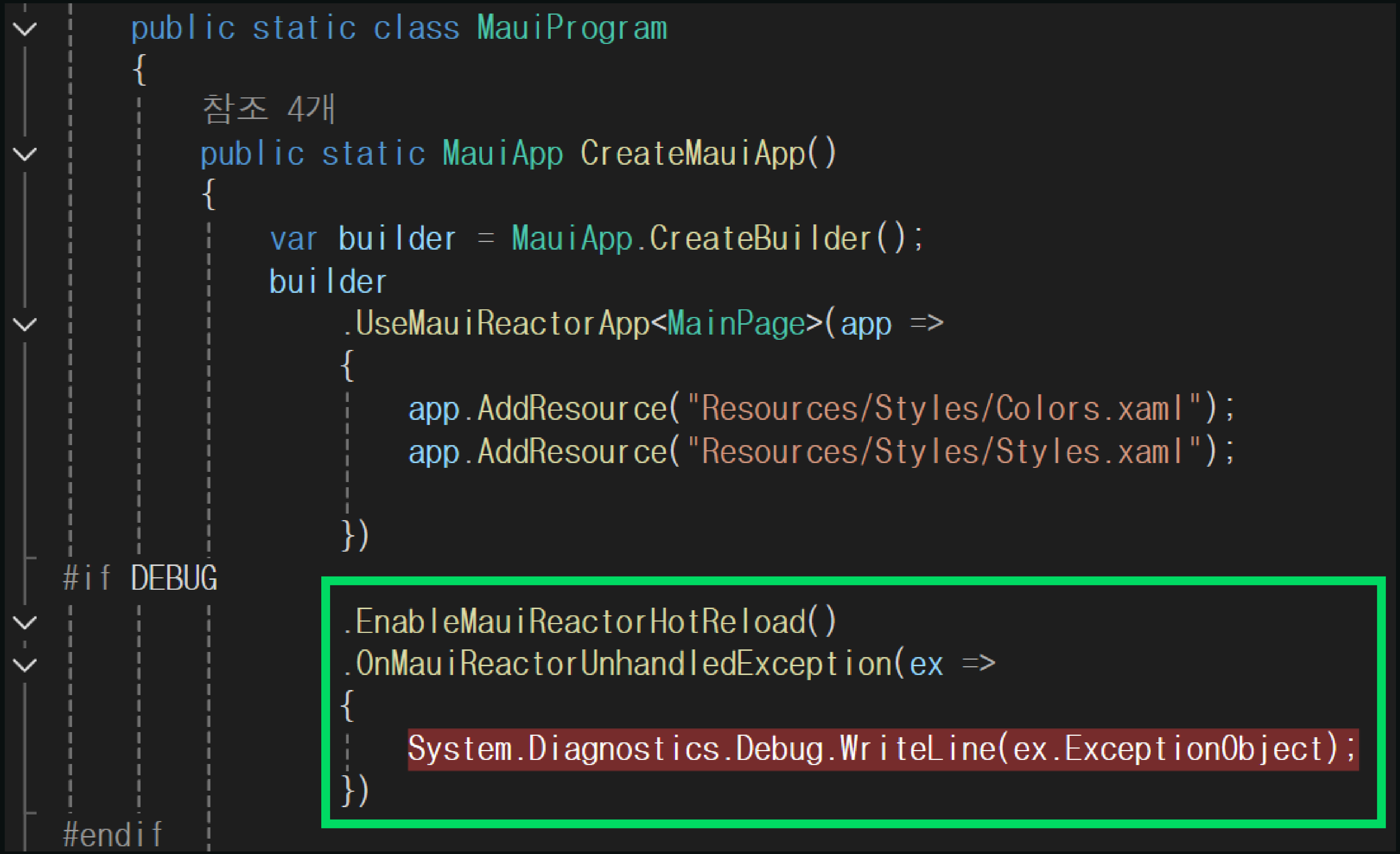
MauiReactor based app (Adolfo Marinucci)
A project for creating a .NET MAUI application based on the MauiReactor library
C# · MAUI

MauiReactor based app with App Shell (Adolfo Marinucci - James Alickolli)
A project for creating a .NET MAUI application based on the MauiReactor library
C# · MAUI

.NET MAUI 앱
iOS, Android, Mac Catalyst, Tizen 및 WinUI용 .NET MAUI 애플리케이션을 만들기 위한 프로젝트
C# · Android · iOS · Mac Catalyst · macOS · MAUI · 모바일 · Tizen · Windows

.NET MAUI Blazor 앱
Blazor를 사용하여 iOS, Android, Mac Catalyst, Tizen 및 WinUI용 .NET MAUI 애플리케이션을 만들기 위한 프로젝트
C# · Android · Blazor · Blazor Hybrid · iOS · Mac Catalyst · macOS · MAUI · 모바일 · Tizen · Windows

TIP #1 MauiReactorUnhandledExcetion 활용

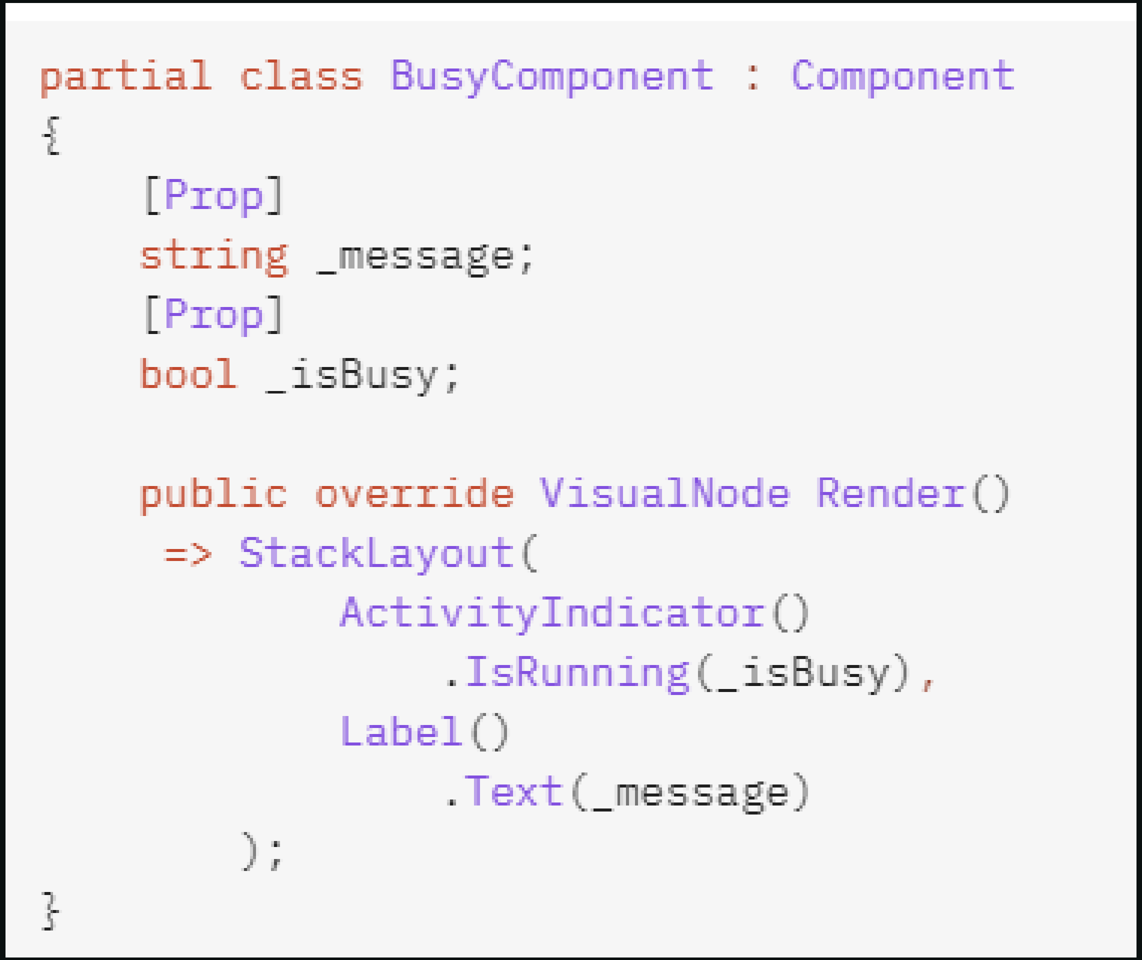


```
    public static class MauiProgram
    {
        참조 4개
        public static MauiApp CreateMauiApp()
        {
            var builder = MauiApp.CreateBuilder();
            builder
                .UseMauiReactorApp<MainPage>(app =>
                {
                    app.AddResource("Resources/Styles/Colors.xaml");
                    app.AddResource("Resources/Styles/Styles.xaml");

                })
#if DEBUG
                .EnableMauiReactorHotReload()
                .OnMauiReactorUnhandledException(ex =>
                {
                    System.Diagnostics.Debug.WriteLine(ex.ExceptionObject);
                })
#endif
```

TIP #2 Component에 반드시 State가 필요한가

특징	Stateful Component	Stateless Component
상태 관리	내부적으로 상태를 관리하고, 상태 변경에 따라 UI가 변경됨	상태를 관리하지 않음, 고정된 UI를 렌더링
UI 업데이트	상태가 변경될 때마다 리 렌더링	외부 데이터나 고정된 데이터에 따라 한 번 렌더링
사용 예시	버튼 클릭, 폼 입력, 동적 리스트 등	단순 텍스트, 이미지, 외부 데이터 표시
렌더링 성능	상태 변경 시마다 리 렌더링이 발생하여 상대적으로 비용이 있을 수 있음	상태 변경이 없으므로 리 렌더링 비용이 적음

TIP #3 Props 사용하기

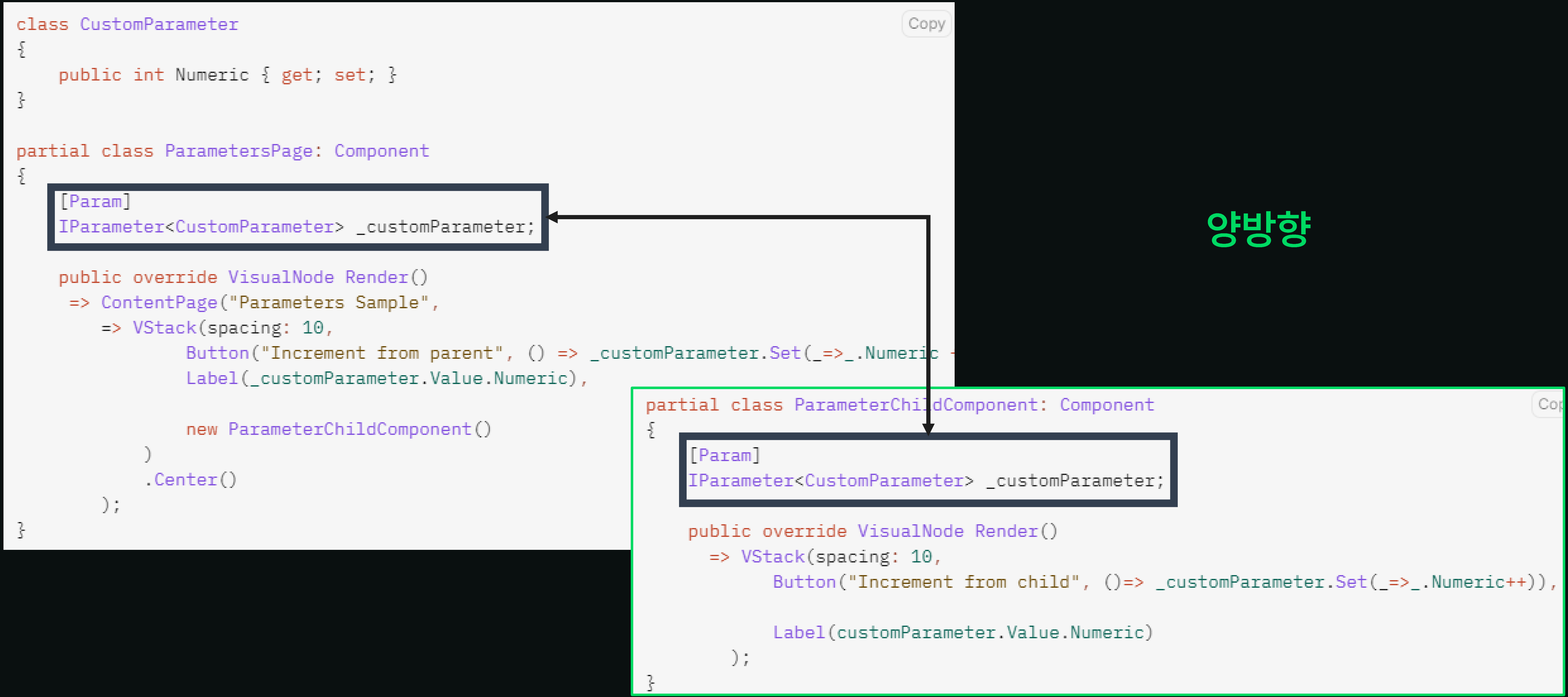


```
partial class BusyComponent : Component
{
    [Prop]
    string _message;
    [Prop]
    bool _isBusy;

    public override VisualNode Render()
     => StackLayout(
            ActivityIndicator()
                .IsRunning(_isBusy),
            Label()
                .Text(_message)
        );
}
```

단방향

Action을 통해서 하위 Component의 이벤트 추적 가능

TIP #4 Param 사용하기

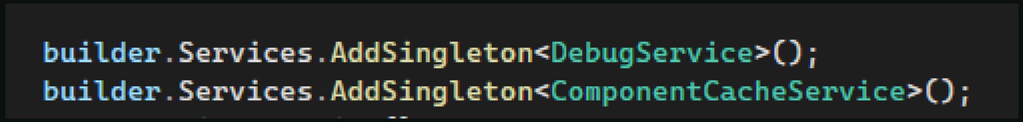


```
class CustomParameter
{
    public int Numeric { get; set; }
}

partial class ParametersPage: Component
{
    [Param]
    IParameter<CustomParameter> _customParameter;

    public override VisualNode Render()
     => ContentPage("Parameters Sample",
        => VStack(spacing: 10,
                Button("Increment from parent", () => _customParameter.Set(_=>_.Numeric +
                Label(_customParameter.Value.Numeric),

                new ParameterChildComponent()
            )
            .Center()
        );
}
```

```
partial class ParameterChildComponent: Component
{
    [Param]
    IParameter<CustomParameter> _customParameter;

    public override VisualNode Render()
      => VStack(spacing: 10,
            Button("Increment from child", ()=> _customParameter.Set(_=>_.Numeric++)),

            Label(customParameter.Value.Numeric)
        );
}
```

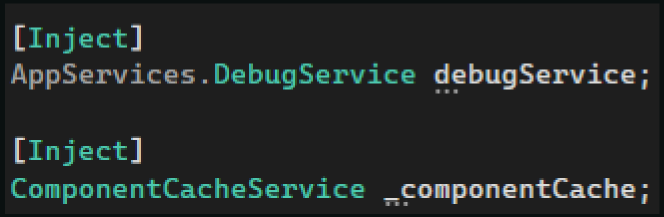
양방향

TIP #5 Inject 사용하기



```
builder.Services.AddSingleton<DebugService>();
builder.Services.AddSingleton<ComponentCacheService>();
```

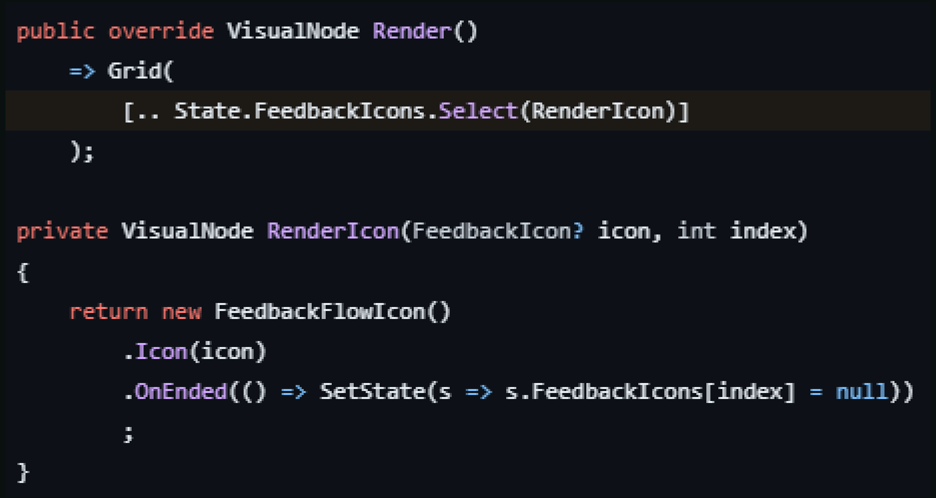
MauiProgram



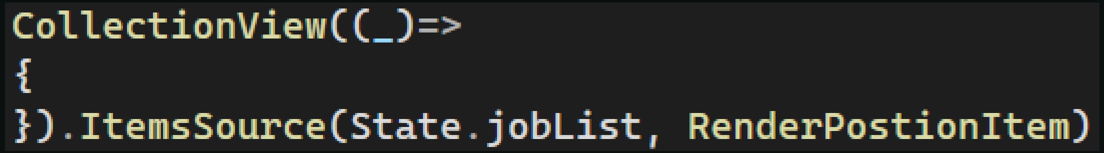
```
[Inject]
AppServices.DebugService debugService;

[Inject]
ComponentCacheService _componentCache;
```

Component

TIP #6 Collection 객체 표현하기

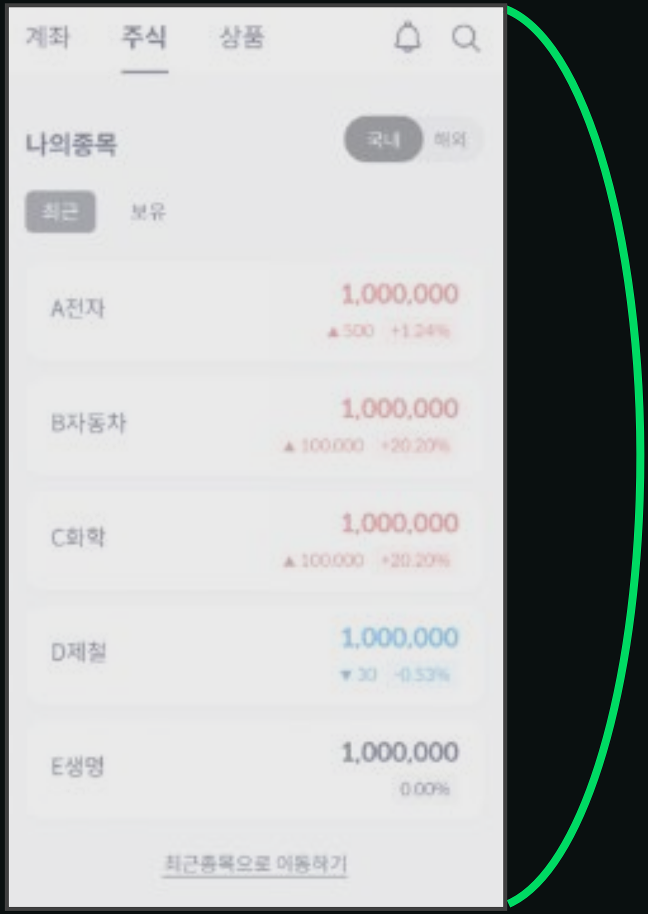


```
public override VisualNode Render()
    => Grid(
        [.. State.FeedbackIcons.Select(RenderIcon)]
    );

private VisualNode RenderIcon(FeedbackIcon? icon, int index)
{
    return new FeedbackFlowIcon()
        .Icon(icon)
        .OnEnded(() => SetState(s => s.FeedbackIcons[index] = null))
        ;
}
```



```
CollectionView((_)=>
{
}).ItemsSource(State.jobList, RenderPostionItem)
```

TIP #7 최대한 잘게 Component를 쪼개야 되는 이유



A전자 1,000,000 ▲500 +1.24%

B자동차 1,000,000 ▲100,000 +20.20%

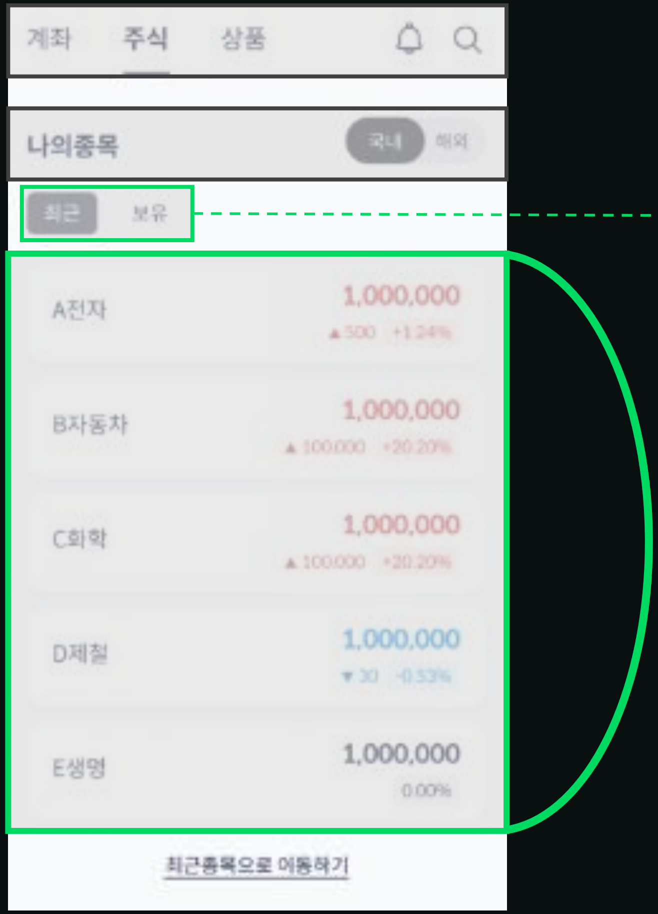
C화학 1,000,000 ▲100,000 +20.20%

D제철 1,000,000 ▼30 -0.53%

E생명 1,000,000 0.00%

최근종목으로 이동하기

"어 바꿀께 "



A전자 1,000,000 ▲500 +1.24%

B자동차 1,000,000 ▲100,000 +20.20%

C화학 1,000,000 ▲100,000 +20.20%

D제철 1,000,000 ▼30 -0.53%

E생명 1,000,000 0.00%

최근종목으로 이동하기

"리스트야! 보유 목록 불러와줘"

"보유 상태 리스트 API 호출 가즈아!"

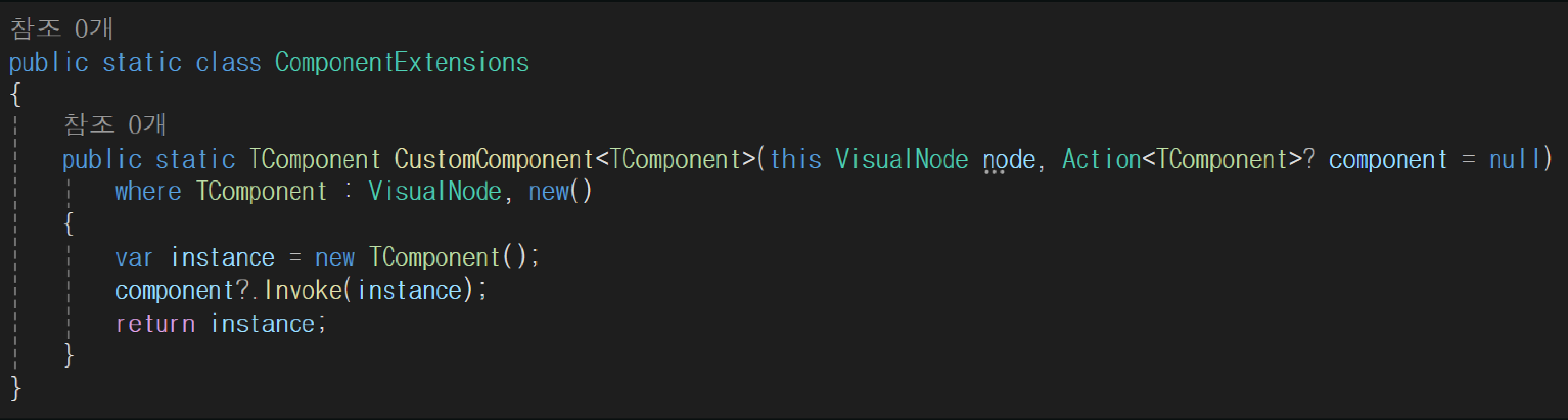
TIP #8 OnMounted는 Hot Reload가 다시 호출하지 않는다

OnMounted() = React의 componentDidMount(), useEffect()

네트워크 요청, 애니메이션 시작, 이벤트 리스너 등 비동기 작업 시

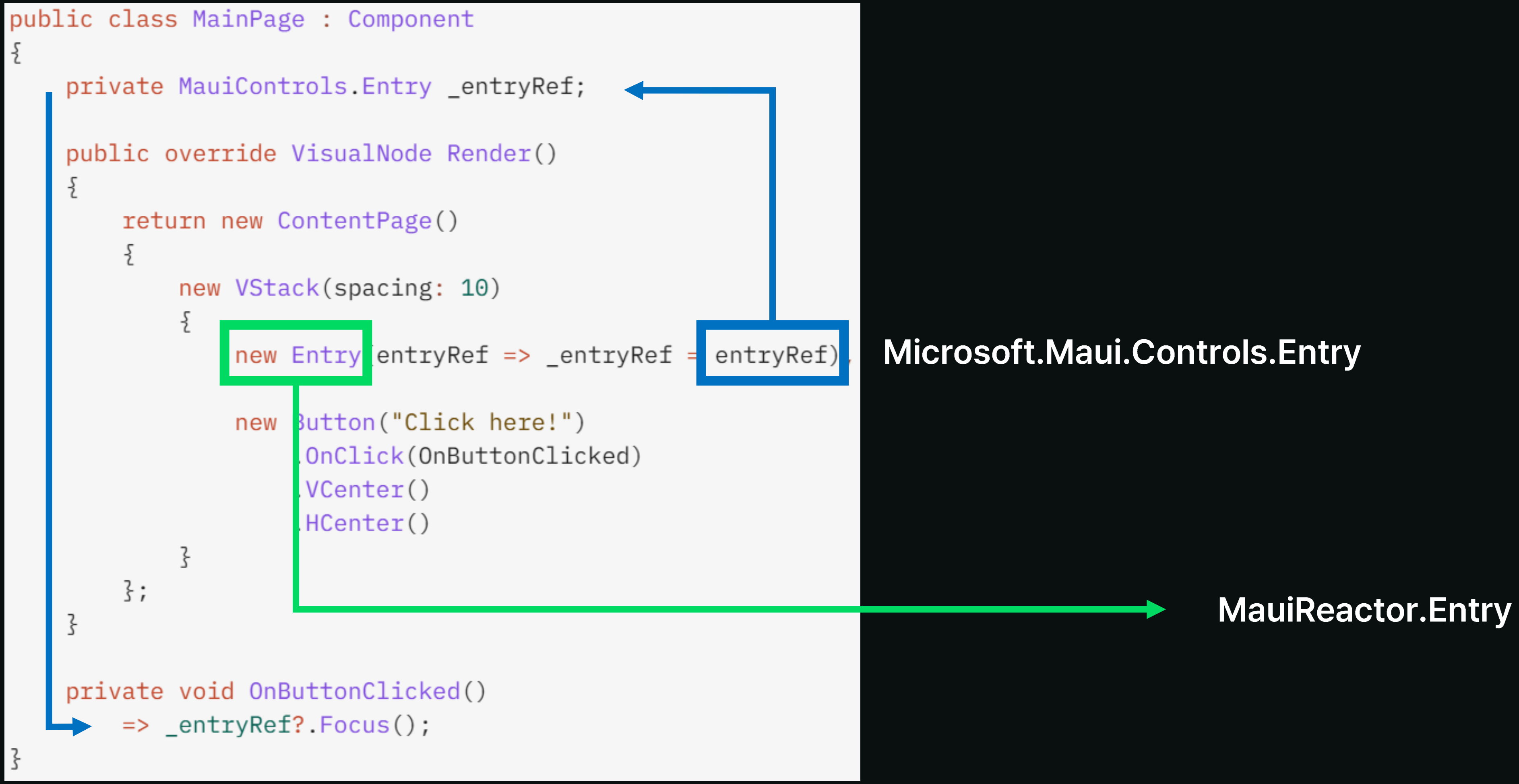
마운트 된 이후 (UI가 그려진 이후) 수행하는 것이 안전

TIP #9 Component 인스턴스 설정 및 반환 확장 메서드



```
참조 0개
public static class ComponentExtensions
{
    참조 0개
    public static TComponent CustomComponent<TComponent>(this VisualNode node, Action<TComponent>? component = null)
        where TComponent : VisualNode, new()
    {
        var instance = new TComponent();
        component?.Invoke(instance);
        return instance;
    }
}
```

TIP #10 네이티브 컨트롤 엑세스

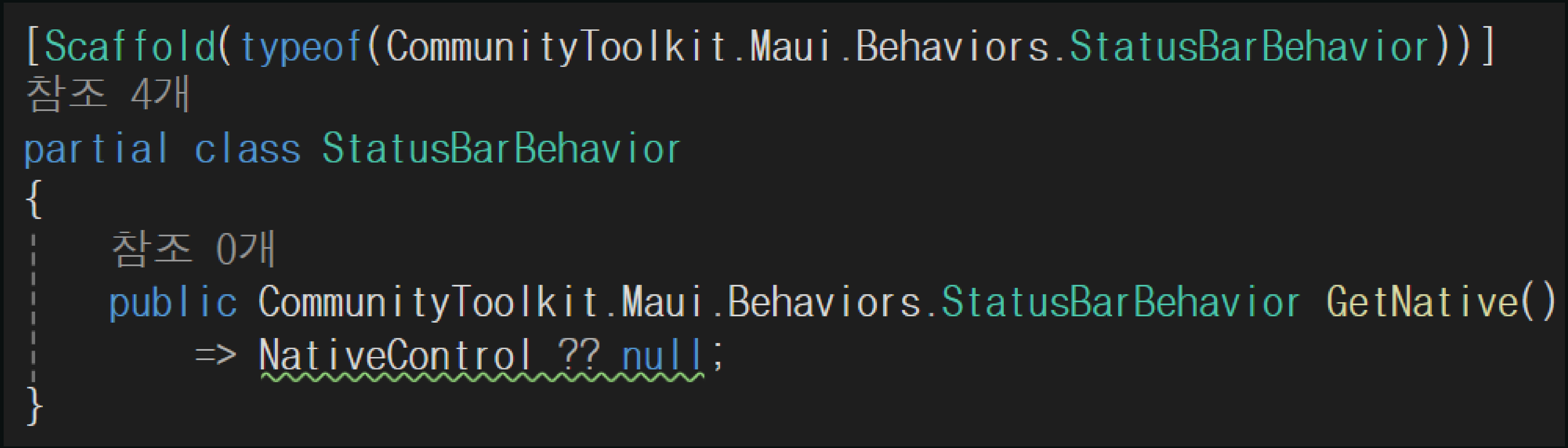


```
public class MainPage : Component
{
    private MauiControls.Entry _entryRef;

    public override VisualNode Render()
    {
        return new ContentPage()
        {
            new VStack(spacing: 10)
            {
                new Entry(entryRef => _entryRef = entryRef),

                new Button("Click here!")
                    .OnClick(OnButtonClicked)
                    .VCenter()
                    .HCenter()
            }
        };
    }

    private void OnButtonClicked()
        => _entryRef?.Focus();
}
```

Microsoft.Maui.Controls.Entry

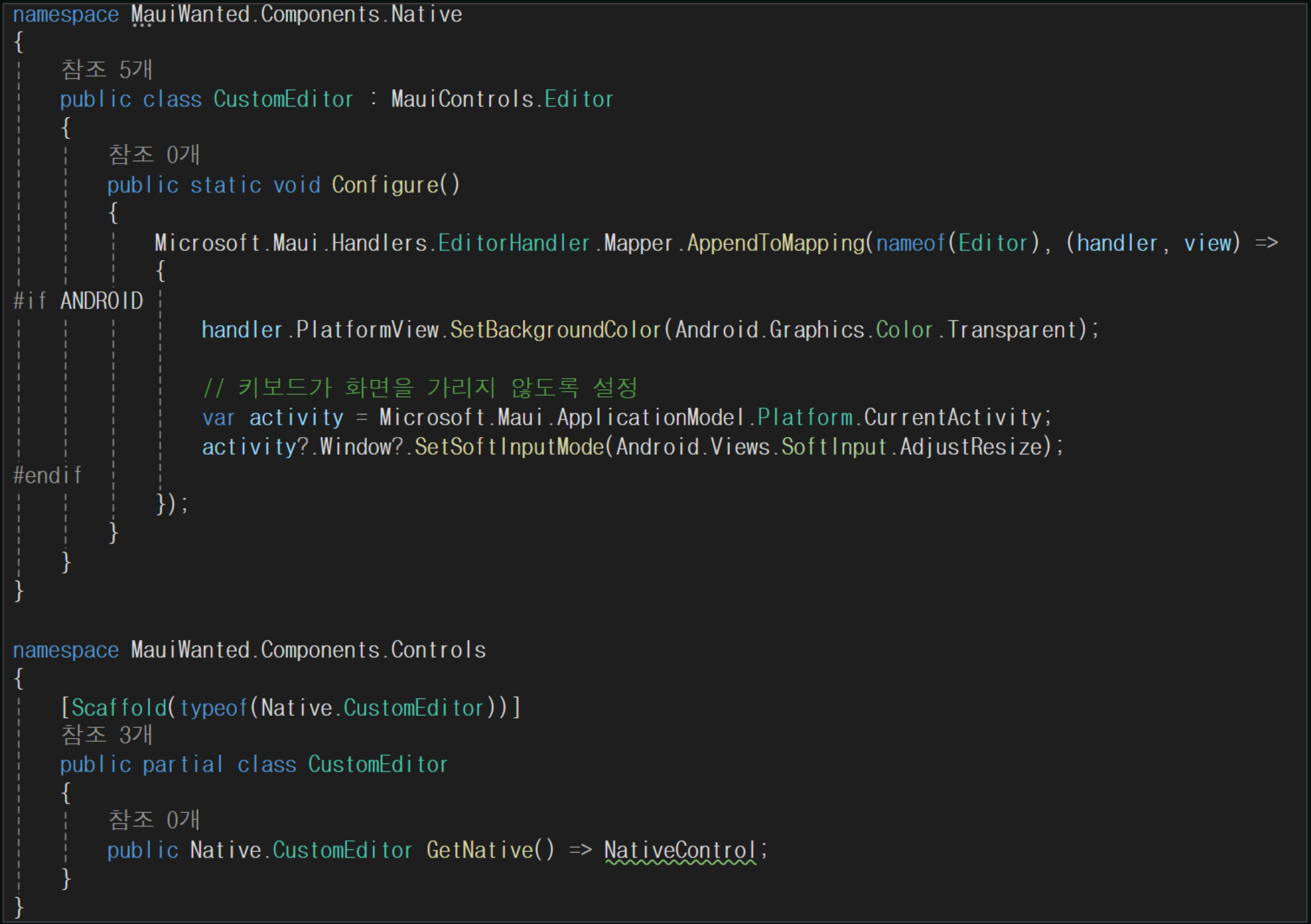
MauiReactor.Entry

TIP #11 Scaffold 사용하기 (외부 라이브러리)



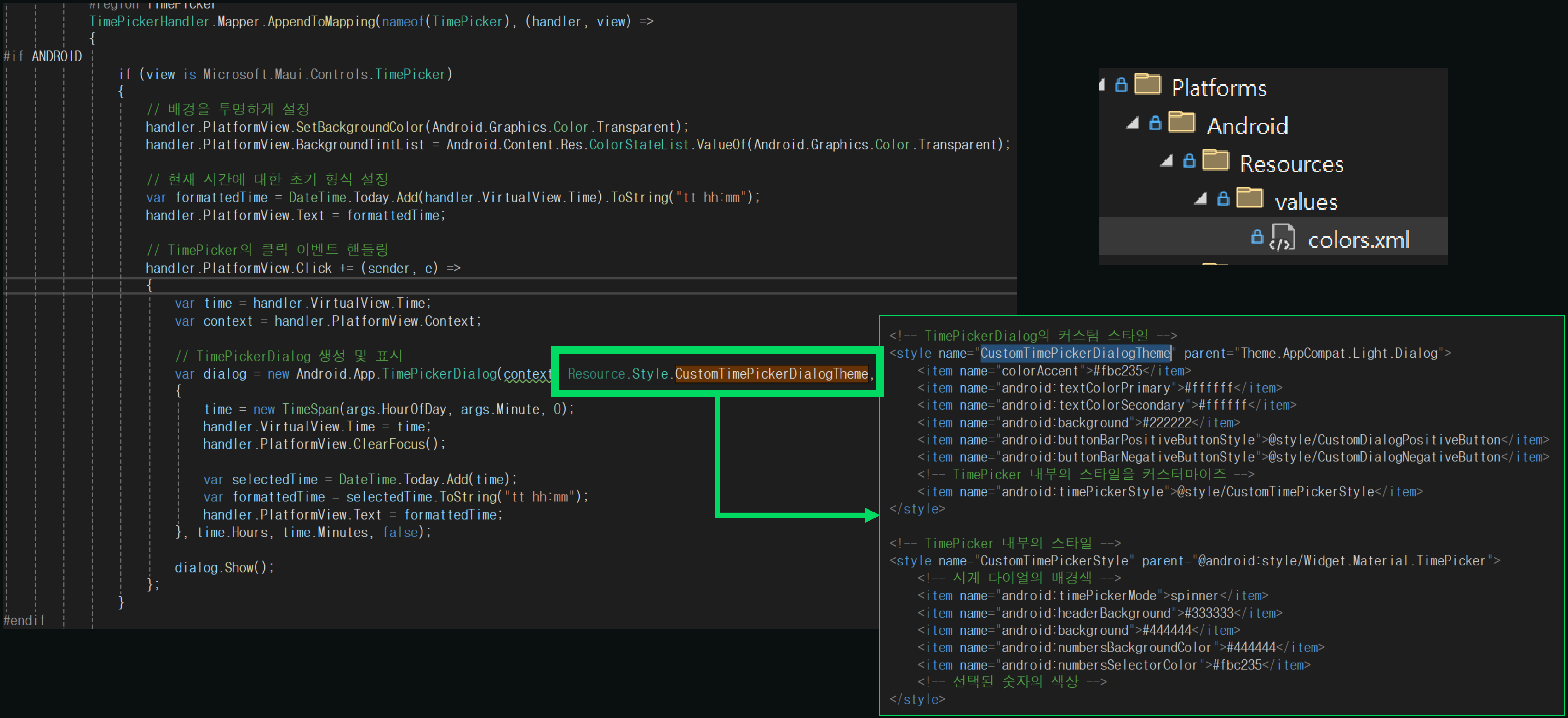
```
[Scaffold(typeof(CommunityToolkit.Maui.Behaviors.StatusBarBehavior))]
참조 4개
partial class StatusBarBehavior
{
    참조 0개
    public CommunityToolkit.Maui.Behaviors.StatusBarBehavior GetNative()
        => NativeControl ?? null;
}
```

TIP #12 Scaffold 사용하기 (컨트롤 커스텀이 필요한 경우)



```
namespace MauiWanted.Components.Native
{
    참조 5개
    public class CustomEditor : MauiControls.Editor
    {
        참조 0개
        public static void Configure()
        {
            Microsoft.Maui.Handlers.EditorHandler.Mapper.AppendToMapping(nameof(Editor), (handler, view) =>
            {
#if ANDROID
                handler.PlatformView.SetBackgroundColor(Android.Graphics.Color.Transparent);

                // 키보드가 화면을 가리지 않도록 설정
                var activity = Microsoft.Maui.ApplicationModel.Platform.CurrentActivity;
                activity?.Window?.SetSoftInputMode(Android.Views.SoftInput.AdjustResize);
#endif
            });
        }
    }
}

namespace MauiWanted.Components.Controls
{
    [Scaffold(typeof(Native.CustomEditor))]
    참조 3개
    public partial class CustomEditor
    {
        참조 0개
        public Native.CustomEditor GetNative() => NativeControl;
    }
}
```

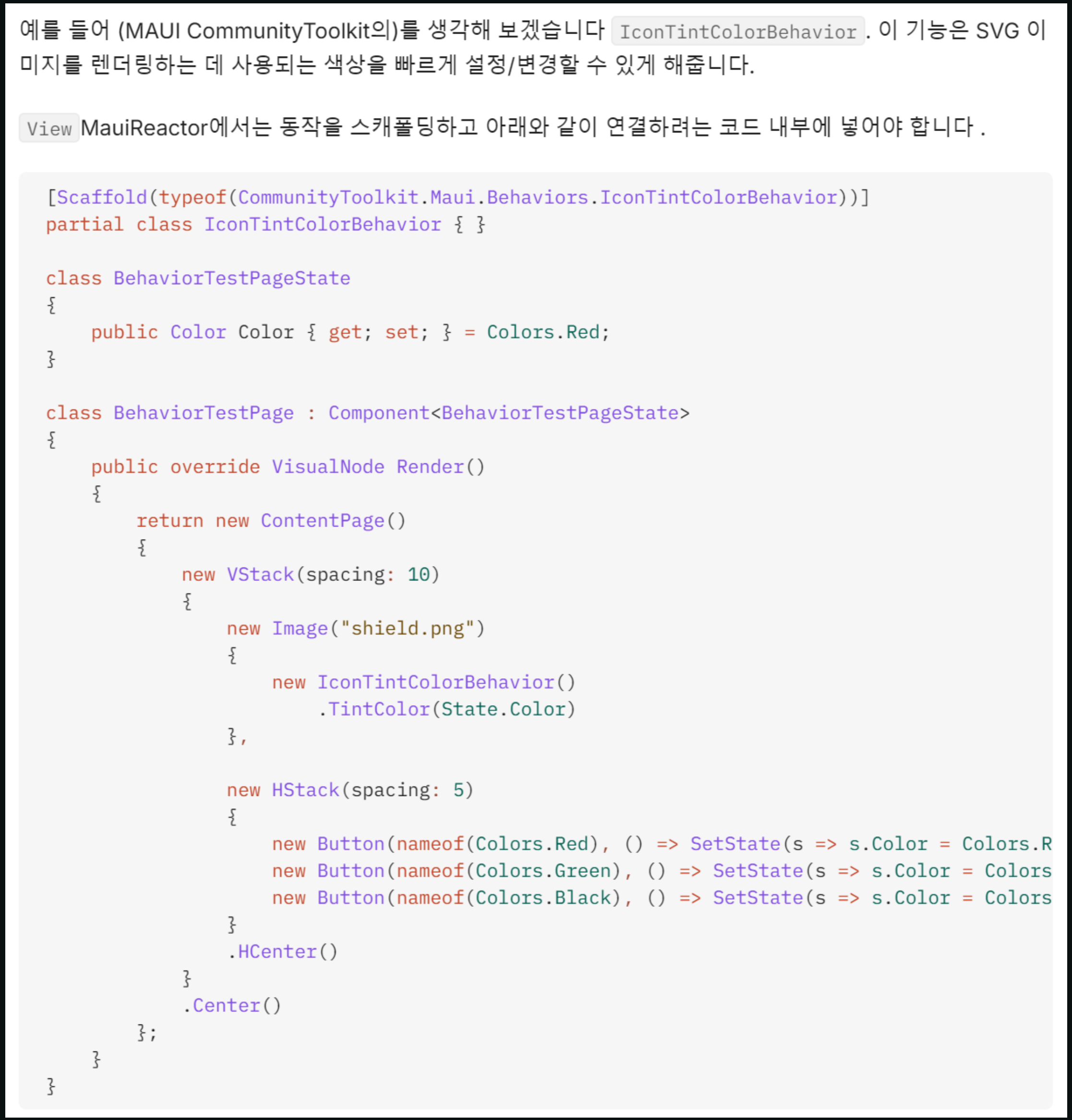
TIP #13 Scaffold 사용하기 (컨트롤 커스텀이 필요한 경우)



```
        #region TimePicker
        TimePickerHandler.Mapper.AppendToMapping(nameof(TimePicker), (handler, view) =>
        {
#if ANDROID
            if (view is Microsoft.Maui.Controls.TimePicker)
            {
                // 배경을 투명하게 설정
                handler.PlatformView.SetBackgroundColor(Android.Graphics.Color.Transparent);
                handler.PlatformView.BackgroundTintList = Android.Content.Res.ColorStateList.ValueOf(Android.Graphics.Color.Transparent);

                // 현재 시간에 대한 초기 형식 설정
                var formattedTime = DateTime.Today.Add(handler.VirtualView.Time).ToString("tt hh:mm");
                handler.PlatformView.Text = formattedTime;

                // TimePicker의 클릭 이벤트 핸들링
                handler.PlatformView.Click += (sender, e) =>
                {
                    var time = handler.VirtualView.Time;
                    var context = handler.PlatformView.Context;

                    // TimePickerDialog 생성 및 표시
                    var dialog = new Android.App.TimePickerDialog(context  Resource.Style.CustomTimePickerDialogTheme,
                    {
                        time = new TimeSpan(args.HourOfDay, args.Minute, 0);
                        handler.VirtualView.Time = time;
                        handler.PlatformView.ClearFocus();

                        var selectedTime = DateTime.Today.Add(time);
                        var formattedTime = selectedTime.ToString("tt hh:mm");
                        handler.PlatformView.Text = formattedTime;
                    }, time.Hours, time.Minutes, false);

                    dialog.Show();
                };
            }
#endif
```

```
<!-- TimePickerDialog의 커스텀 스타일 -->
<style name="CustomTimePickerDialogTheme" parent="Theme.AppCompat.Light.Dialog">
    <item name="colorAccent">#fbc235</item>
    <item name="android:textColorPrimary">#ffffff</item>
    <item name="android:textColorSecondary">#ffffff</item>
    <item name="android:background">#222222</item>
    <item name="android:buttonBarPositiveButtonStyle">@style/CustomDialogPositiveButton</item>
    <item name="android:buttonBarNegativeButtonStyle">@style/CustomDialogNegativeButton</item>
    <!-- TimePicker 내부의 스타일을 커스터마이즈 -->
    <item name="android:timePickerStyle">@style/CustomTimePickerStyle</item>
</style>

<!-- TimePicker 내부의 스타일 -->
<style name="CustomTimePickerStyle" parent="@android:style/Widget.Material.TimePicker">
    <!-- 시계 다이얼의 배경색 -->
    <item name="android:timePickerMode">spinner</item>
    <item name="android:headerBackground">#333333</item>
    <item name="android:background">#444444</item>
    <item name="android:numbersBackgroundColor">#444444</item>
    <item name="android:numbersSelectorColor">#fbc235</item>
    <!-- 선택된 숫자의 색상 -->
</style>
```

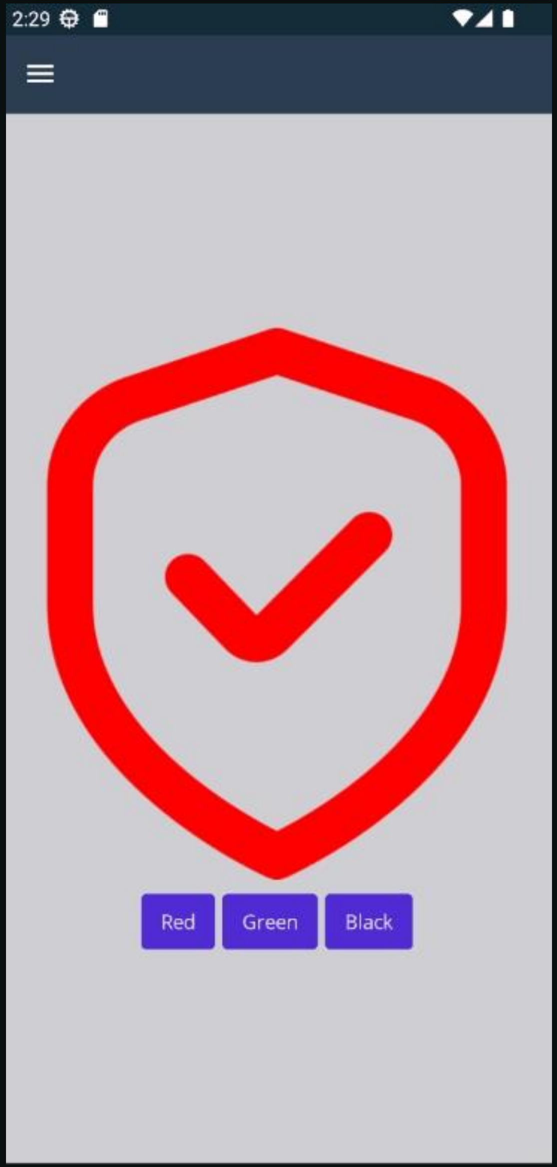
TIP #14 Behavior 사용



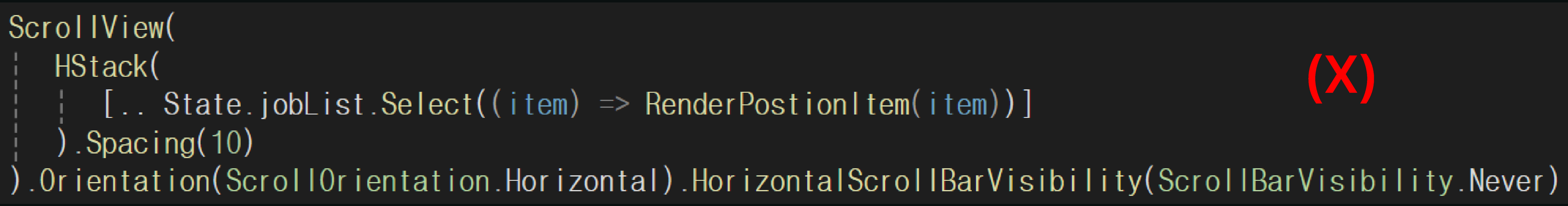
예를 들어 (MAUI CommunityToolkit의)를 생각해 보겠습니다 `IconTintColorBehavior`. 이 기능은 SVG 이미지를 렌더링하는 데 사용되는 색상을 빠르게 설정/변경할 수 있게 해줍니다.

`View` MauiReactor에서는 동작을 스캐폴딩하고 아래와 같이 연결하려는 코드 내부에 넣어야 합니다 .

```
[Scaffold(typeof(CommunityToolkit.Maui.Behaviors.IconTintColorBehavior))]
partial class IconTintColorBehavior { }

class BehaviorTestPageState
{
    public Color Color { get; set; } = Colors.Red;
}

class BehaviorTestPage : Component<BehaviorTestPageState>
{
    public override VisualNode Render()
    {
        return new ContentPage()
        {
            new VStack(spacing: 10)
            {
                new Image("shield.png")
                {
                    new IconTintColorBehavior()
                        .TintColor(State.Color)
                },

                new HStack(spacing: 5)
                {
                    new Button(nameof(Colors.Red), () => SetState(s => s.Color = Colors.R
                    new Button(nameof(Colors.Green), () => SetState(s => s.Color = Colors
                    new Button(nameof(Colors.Black), () => SetState(s => s.Color = Colors
                }
                .HCenter()
            }
            .Center()
        };
    }
}
```

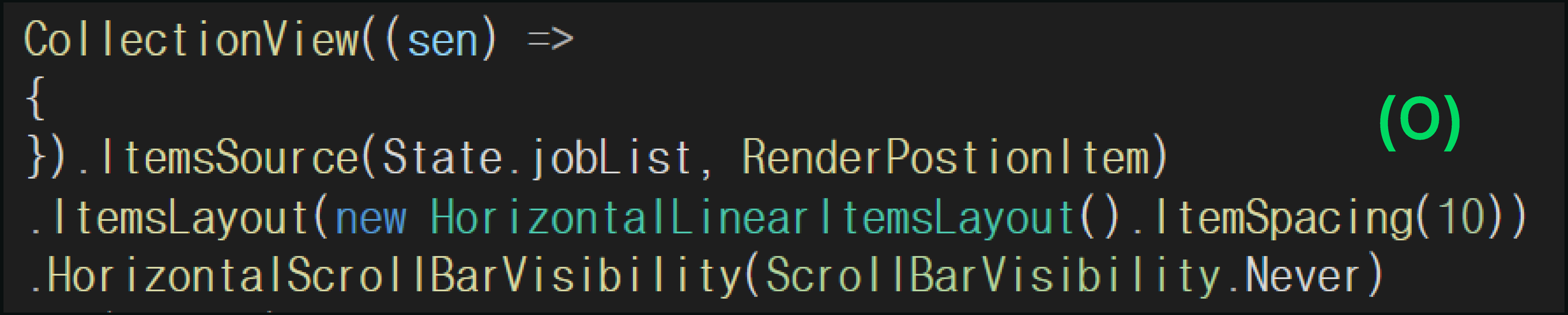


TIP #15 (ScrollView + VStack or HStack) 보다 CollectionView 사용



```
ScrollView(
    HStack(
        [.. State.jobList.Select((item) => RenderPostionItem(item))]
    ).Spacing(10)
).Orientation(ScrollOrientation.Horizontal).HorizontalScrollBarVisibility(ScrollBarVisibility.Never)
```

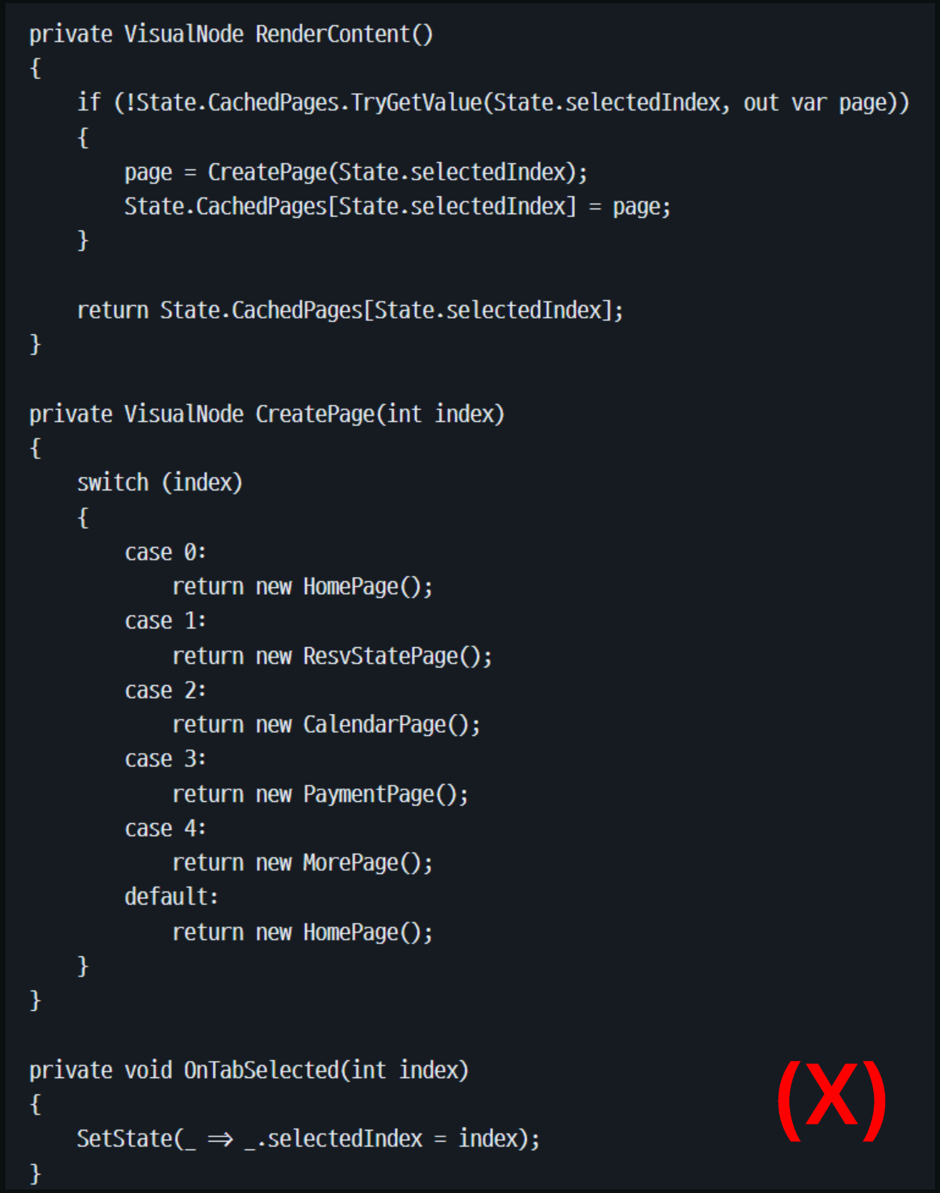
(X)



```
CollectionView((sen) =>
{
}).ItemsSource(State.jobList, RenderPostionItem)
.ItemsLayout(new HorizontalLinearItemsLayout().ItemSpacing(10))
.HorizontalScrollBarVisibility(ScrollBarVisibility.Never)
```

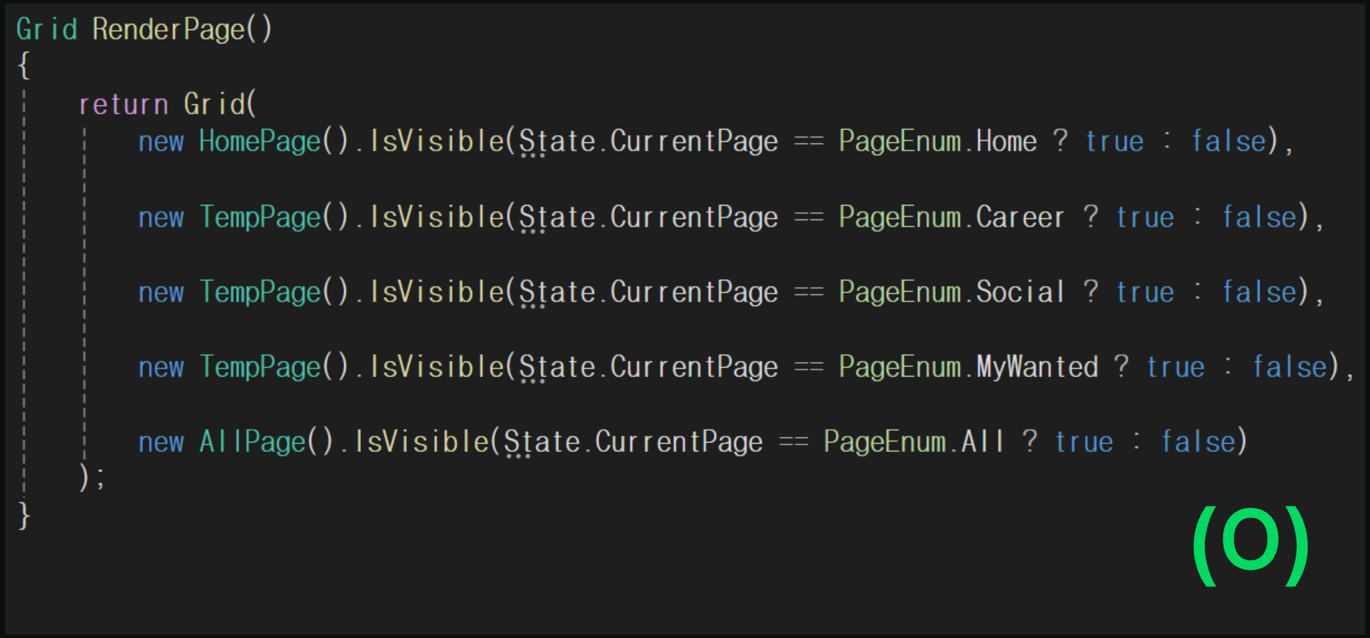
(O)

TIP #16 Bottom Tab



```
private VisualNode RenderContent()
{
    if (!State.CachedPages.TryGetValue(State.selectedIndex, out var page))
    {
        page = CreatePage(State.selectedIndex);
        State.CachedPages[State.selectedIndex] = page;
    }

    return State.CachedPages[State.selectedIndex];
}

private VisualNode CreatePage(int index)
{
    switch (index)
    {
        case 0:
            return new HomePage();
        case 1:
            return new ResvStatePage();
        case 2:
            return new CalendarPage();
        case 3:
            return new PaymentPage();
        case 4:
            return new MorePage();
        default:
            return new HomePage();
    }
}

private void OnTabSelected(int index)
{
    SetState(_ => _.selectedIndex = index);
}
```

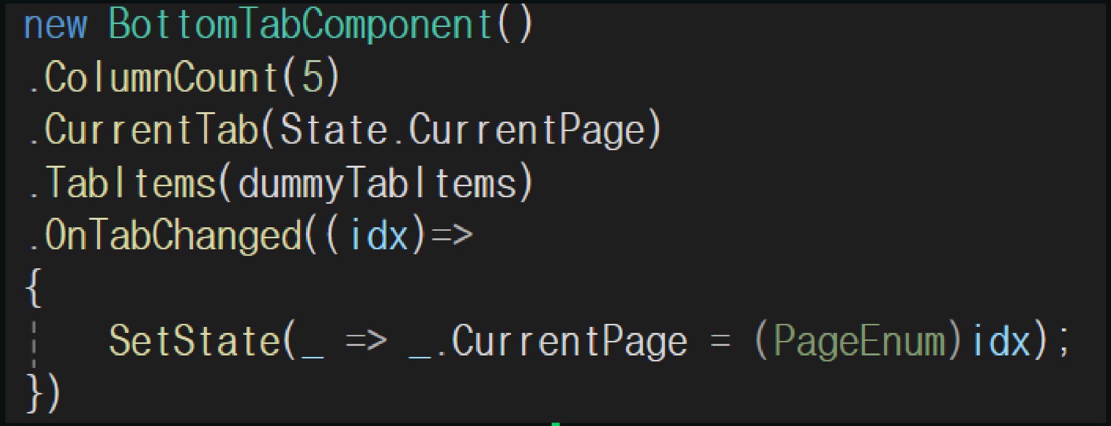
(X)



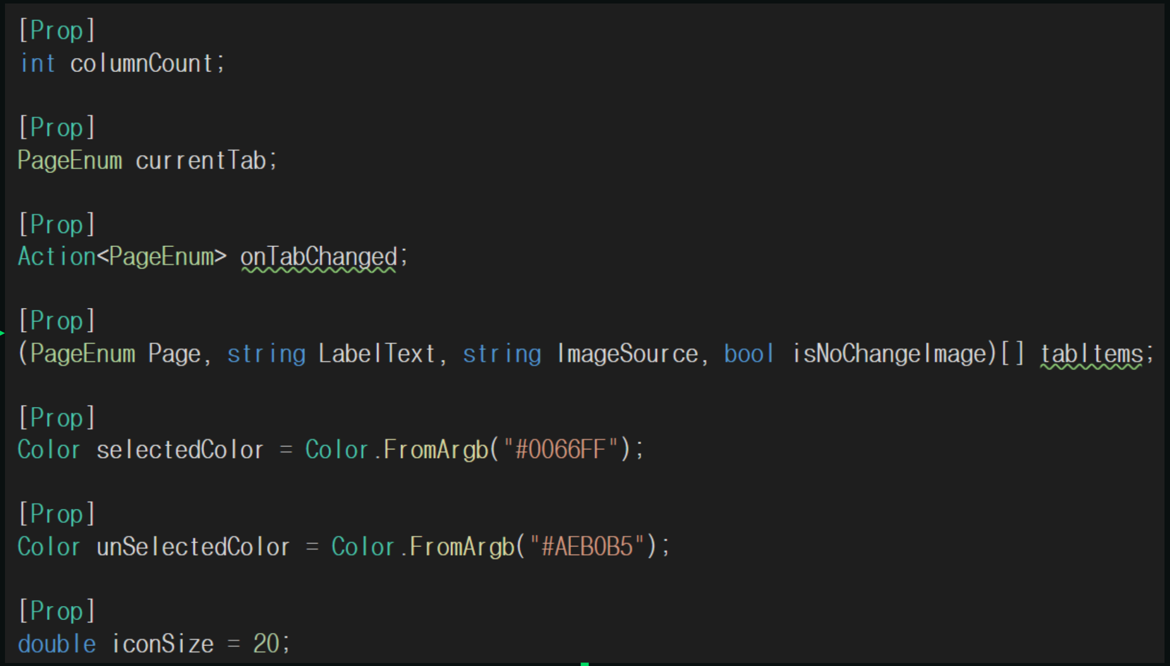
```
Grid RenderPage()
{
    return Grid(
        new HomePage().IsVisible(State.CurrentPage == PageEnum.Home ? true : false),

        new TempPage().IsVisible(State.CurrentPage == PageEnum.Career ? true : false),

        new TempPage().IsVisible(State.CurrentPage == PageEnum.Social ? true : false),

        new TempPage().IsVisible(State.CurrentPage == PageEnum.MyWanted ? true : false),

        new AllPage().IsVisible(State.CurrentPage == PageEnum.All ? true : false)
    );
}
```

(O)

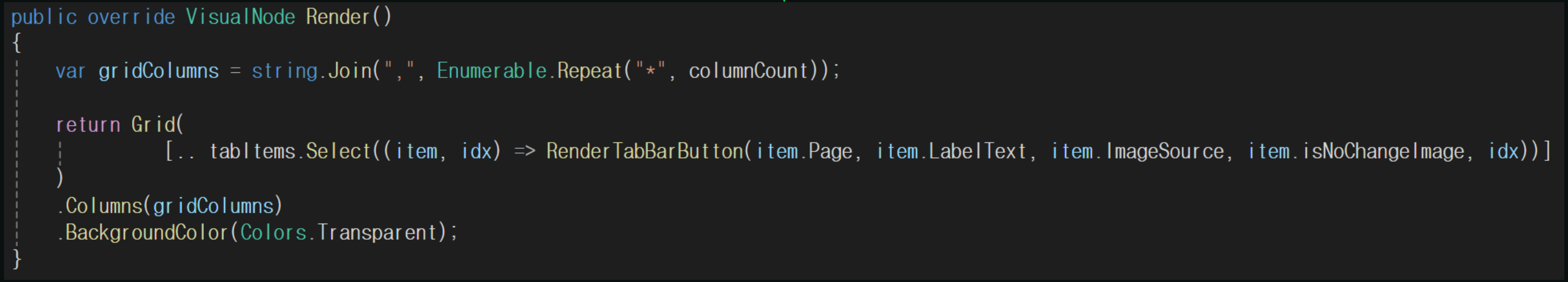
TIP #16 Bottom Tab



```
new BottomTabComponent()
.ColumnCount(5)
.CurrentTab(State.CurrentPage)
.TabItems(dummyTabItems)
.OnTabChanged((idx)=>
{
    SetState(_ => _.CurrentPage = (PageEnum)idx);
})
```

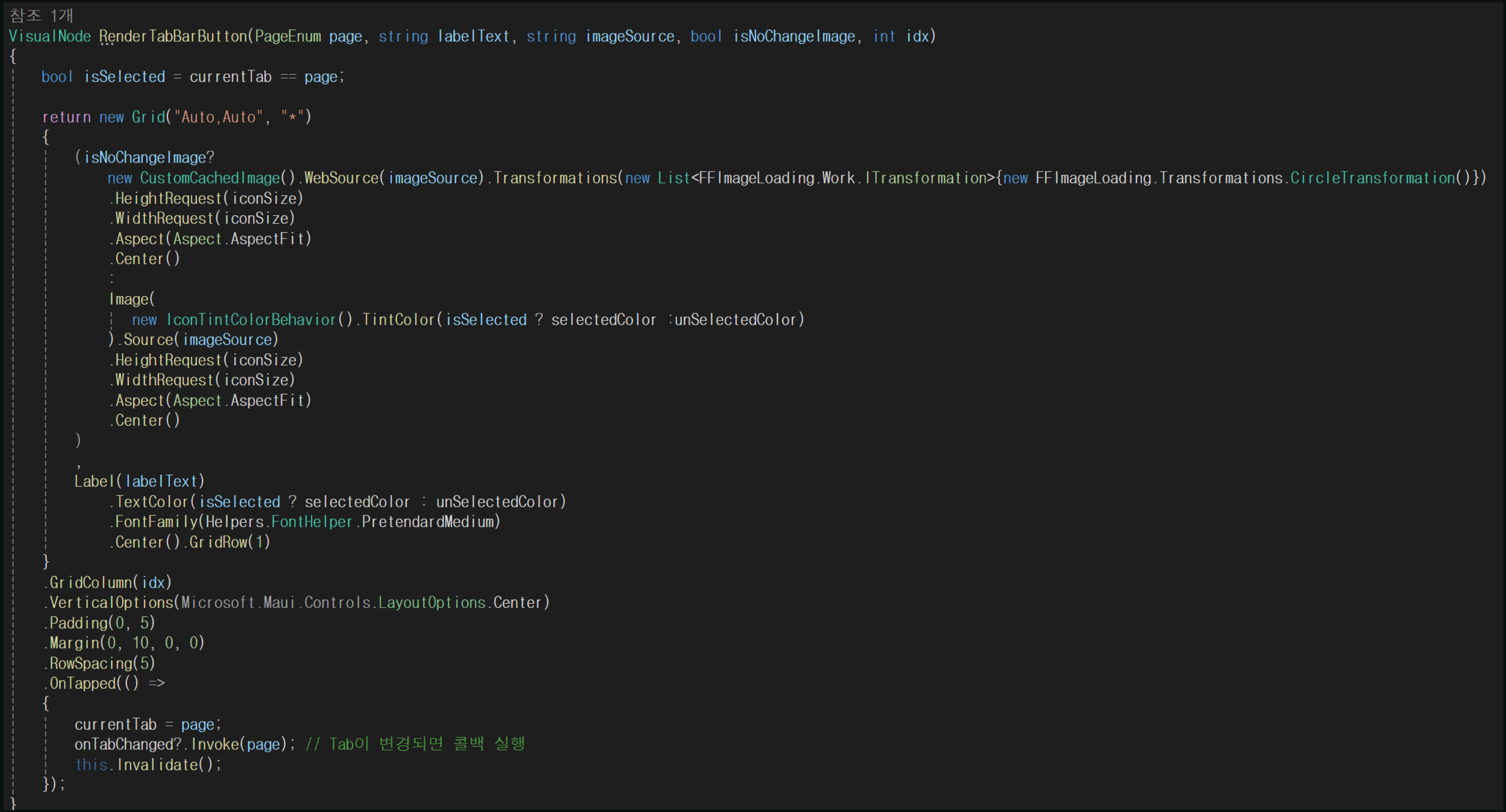


```
[Prop]
int columnCount;

[Prop]
PageEnum currentTab;

[Prop]
Action<PageEnum> onTabChanged;

[Prop]
(PageEnum Page, string LabelText, string ImageSource, bool isNoChangeImage)[] tabItems;

[Prop]
Color selectedColor = Color.FromArgb("#0066FF");

[Prop]
Color unSelectedColor = Color.FromArgb("#AEB0B5");

[Prop]
double iconSize = 20;
```



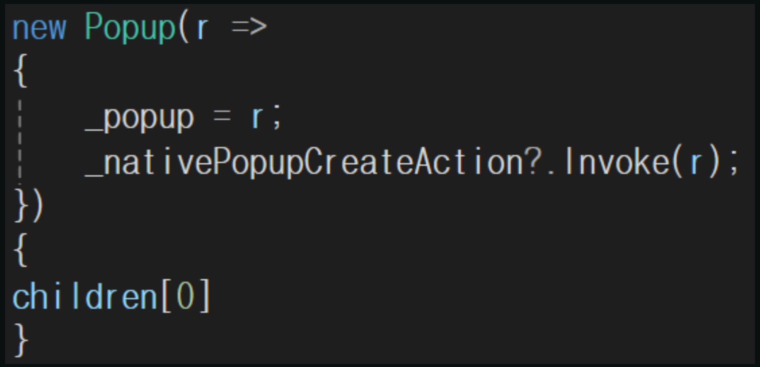
```
public override VisualNode Render()
{
    var gridColumns = string.Join(",", Enumerable.Repeat("*", columnCount));

    return Grid(
            [.. tabItems.Select((item, idx) => RenderTabBarButton(item.Page, item.LabelText, item.ImageSource, item.isNoChangeImage, idx))]
    )
    .Columns(gridColumns)
    .BackgroundColor(Colors.Transparent);
}
```

TIP #16 Bottom Tab

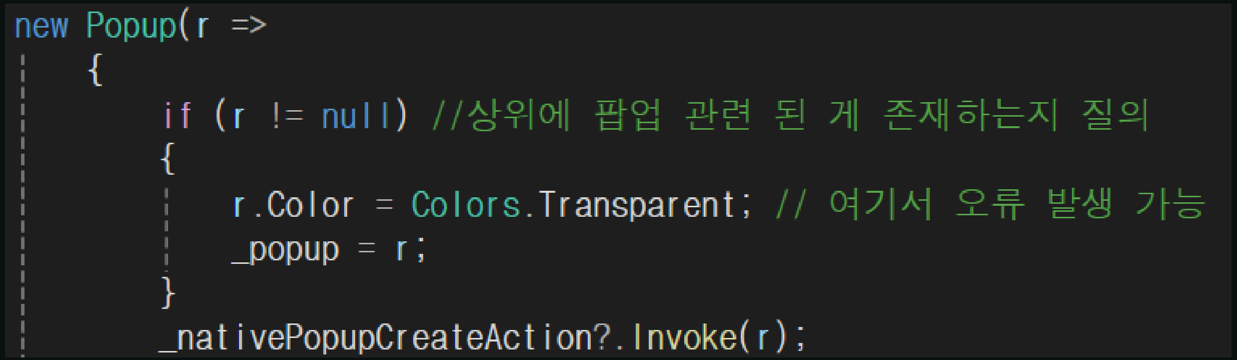


```
참조 1개
VisualNode RenderTabBarButton(PageEnum page, string labelText, string imageSource, bool isNoChangeImage, int idx)
{
    bool isSelected = currentTab == page;

    return new Grid("Auto,Auto", "*")
    {
        (isNoChangeImage?
            new CustomCachedImage().WebSource(imageSource).Transformations(new List<FFImageLoading.Work.ITransformation>{new FFImageLoading.Transformations.CircleTransformation()})
            .HeightRequest(iconSize)
            .WidthRequest(iconSize)
            .Aspect(Aspect.AspectFit)
            .Center()
            :
            Image(
                new IconTintColorBehavior().TintColor(isSelected ? selectedColor :unSelectedColor)
            ).Source(imageSource)
            .HeightRequest(iconSize)
            .WidthRequest(iconSize)
            .Aspect(Aspect.AspectFit)
            .Center()
        )
        ,
        Label(labelText)
            .TextColor(isSelected ? selectedColor : unSelectedColor)
            .FontFamily(Helpers.FontHelper.PretendardMedium)
            .Center().GridRow(1)
    }
    .GridColumn(idx)
    .VerticalOptions(Microsoft.Maui.Controls.LayoutOptions.Center)
    .Padding(0, 5)
    .Margin(0, 10, 0, 0)
    .RowSpacing(5)
    .OnTapped(() =>
    {
        currentTab = page;
        onTabChanged?.Invoke(page); // Tab이 변경되면 콜백 실행
        this.Invalidate();
    });
}
```

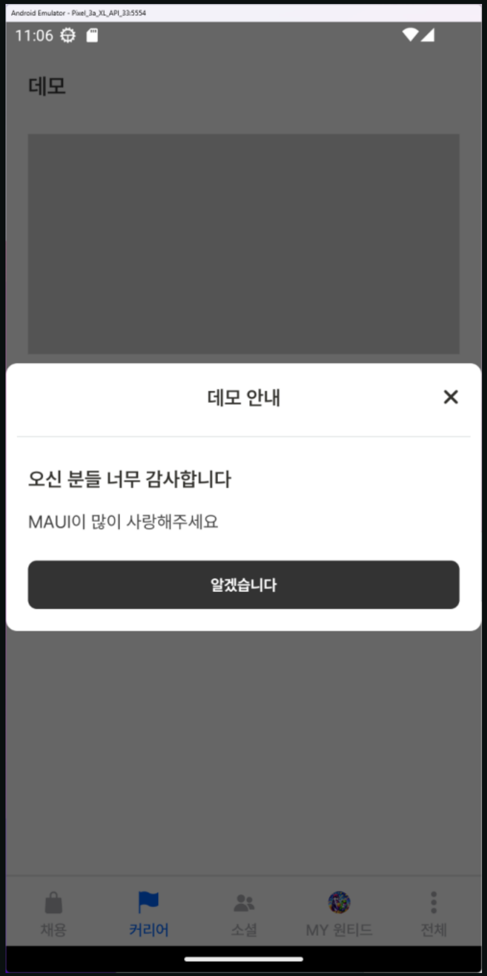
TIP #17 Custom Popup



```
new Popup(r =>
{
    _popup = r;
    _nativePopupCreateAction?.Invoke(r);
})
{
children[0]
}
```



```
new Popup(r =>
    {
        if (r != null) //상위에 팝업 관련 된 게 존재하는지 질의
        {
            r.Color = Colors.Transparent; // 여기서 오류 발생 가능
            _popup = r;
        }
        _nativePopupCreateAction?.Invoke(r);
```



Android Emulator - Pixel_3a_XL_API_33:5554

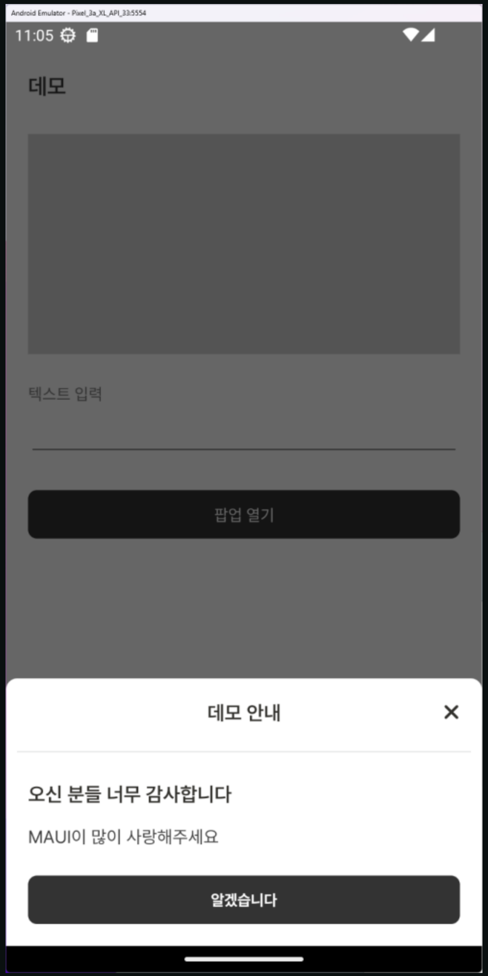
11:06

데모

데모 안내

오신 분들 너무 감사합니다

MAUI이 많이 사랑해주세요



Android Emulator - Pixel_3a_XL_API_33:5554

11:05

데모

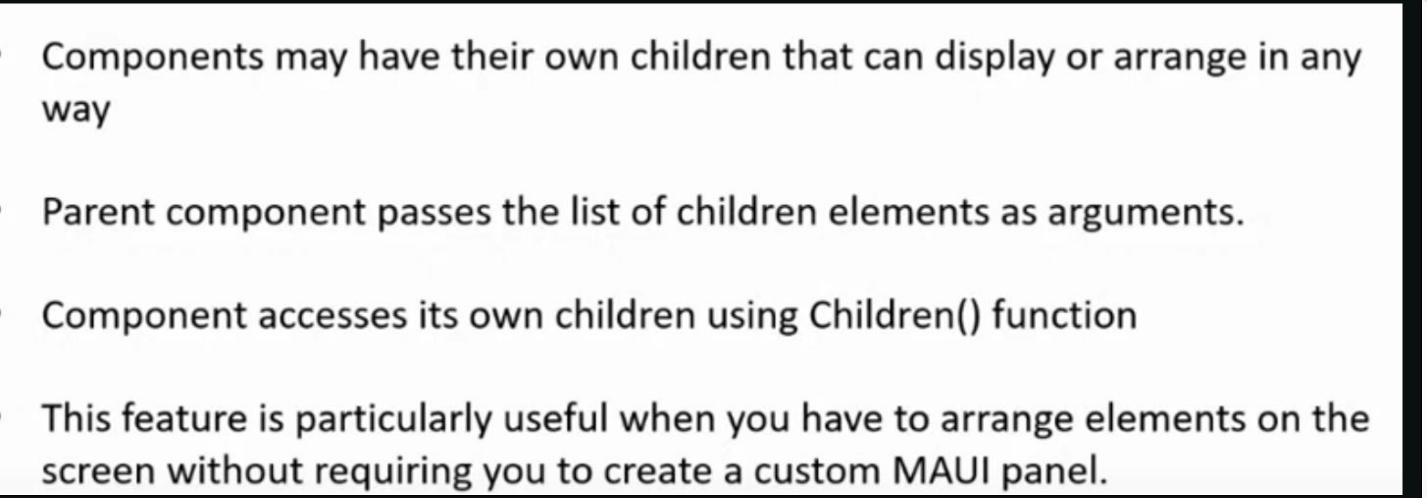
텍스트 입력

데모 안내

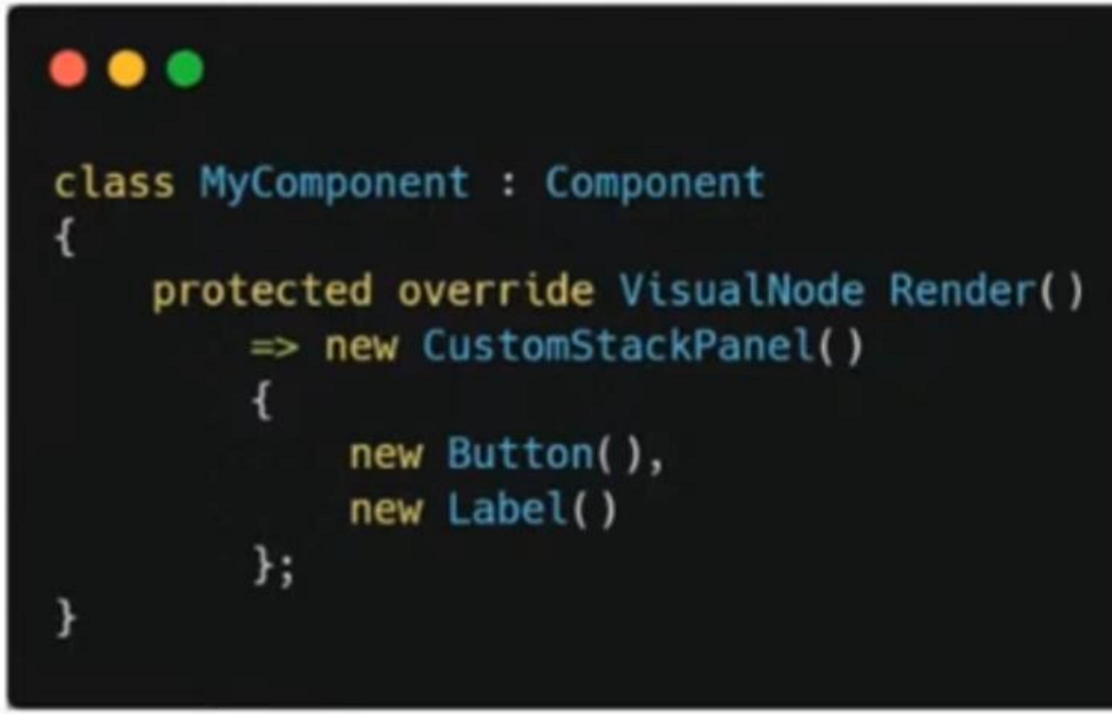
오신 분들 너무 감사합니다

MAUI이 많이 사랑해주세요

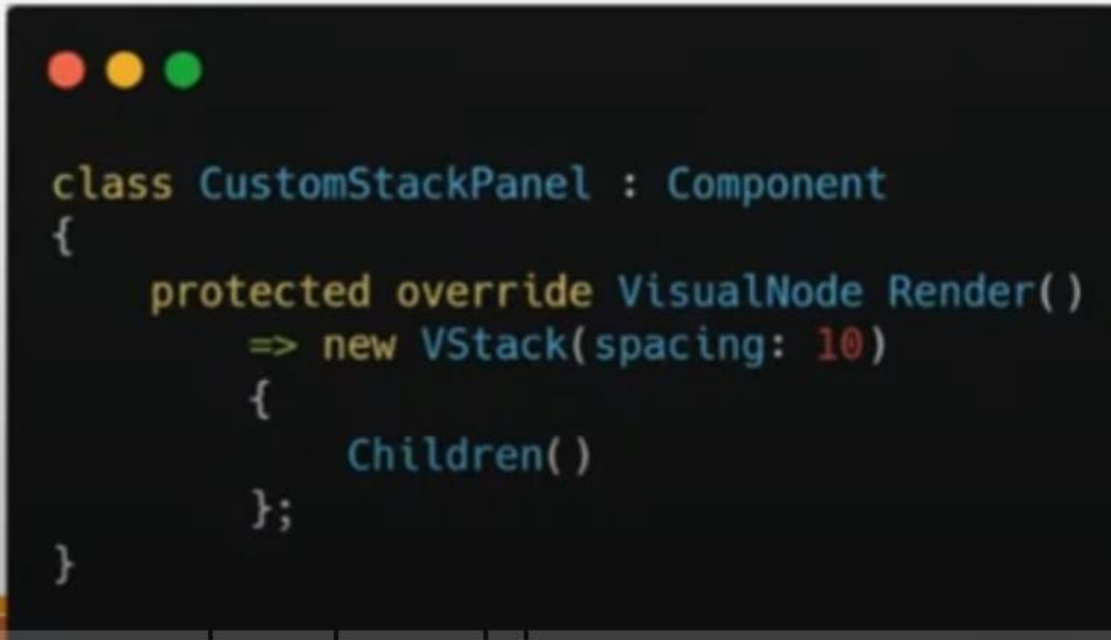
TIP #18 페이지를 위한 베이스 컴포넌트



- Components may have their own children that can display or arrange in any way
- Parent component passes the list of children elements as arguments.
- Component accesses its own children using Children() function
- This feature is particularly useful when you have to arrange elements on the screen without requiring you to create a custom MAUI panel.

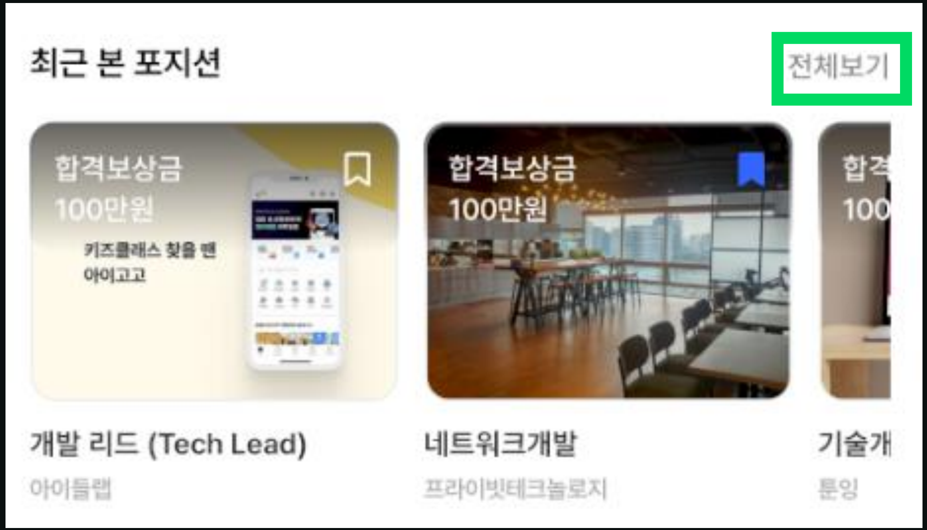


```
class MyComponent : Component
{
    protected override VisualNode Render()
        => new CustomStackPanel()
        {
            new Button(),
            new Label()
        };
}
```



```
class CustomStackPanel : Component
{
    protected override VisualNode Render()
        => new VStack(spacing: 10)
        {
            Children()
        };
}
```

TIP #19 반복 되는 UI는 컴포넌트에서 최대한 재사용

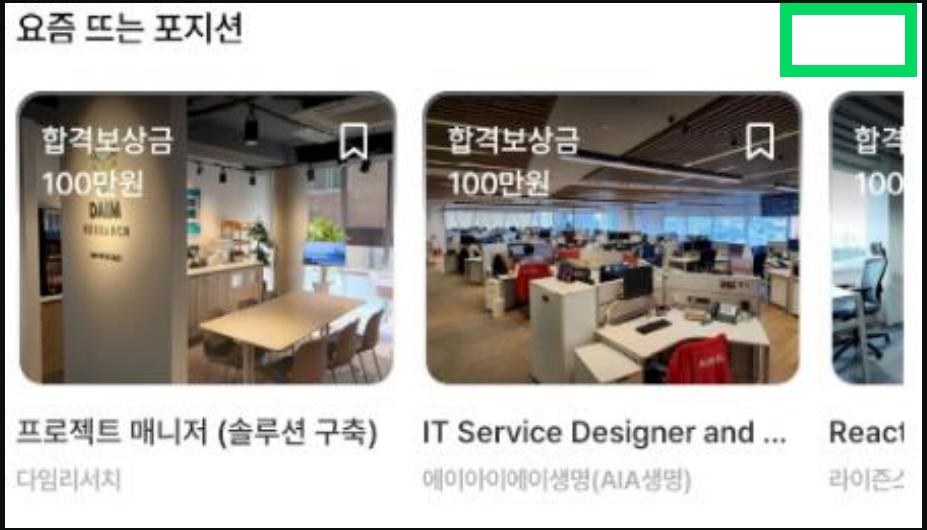


최근 본 포지션 전체보기

개발 리드 (Tech Lead)
아이들랩

네트워크개발
프라이빗테크놀로지

기술개
툰잉



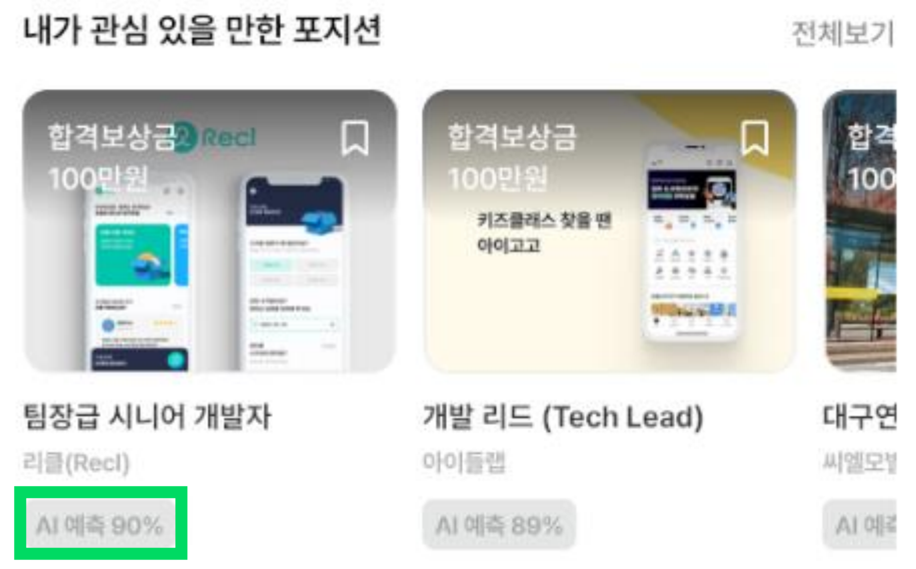
요즘 뜨는 포지션

프로젝트 매니저 (솔루션 구축)
다임리서치

IT Service Designer and ...
에이아이에이생명(AIA생명)

React
라이즈

- IsVisibleRightButton , RightButtonText
  RightButtonTextColor



내가 관심 있을 만한 포지션 전체보기

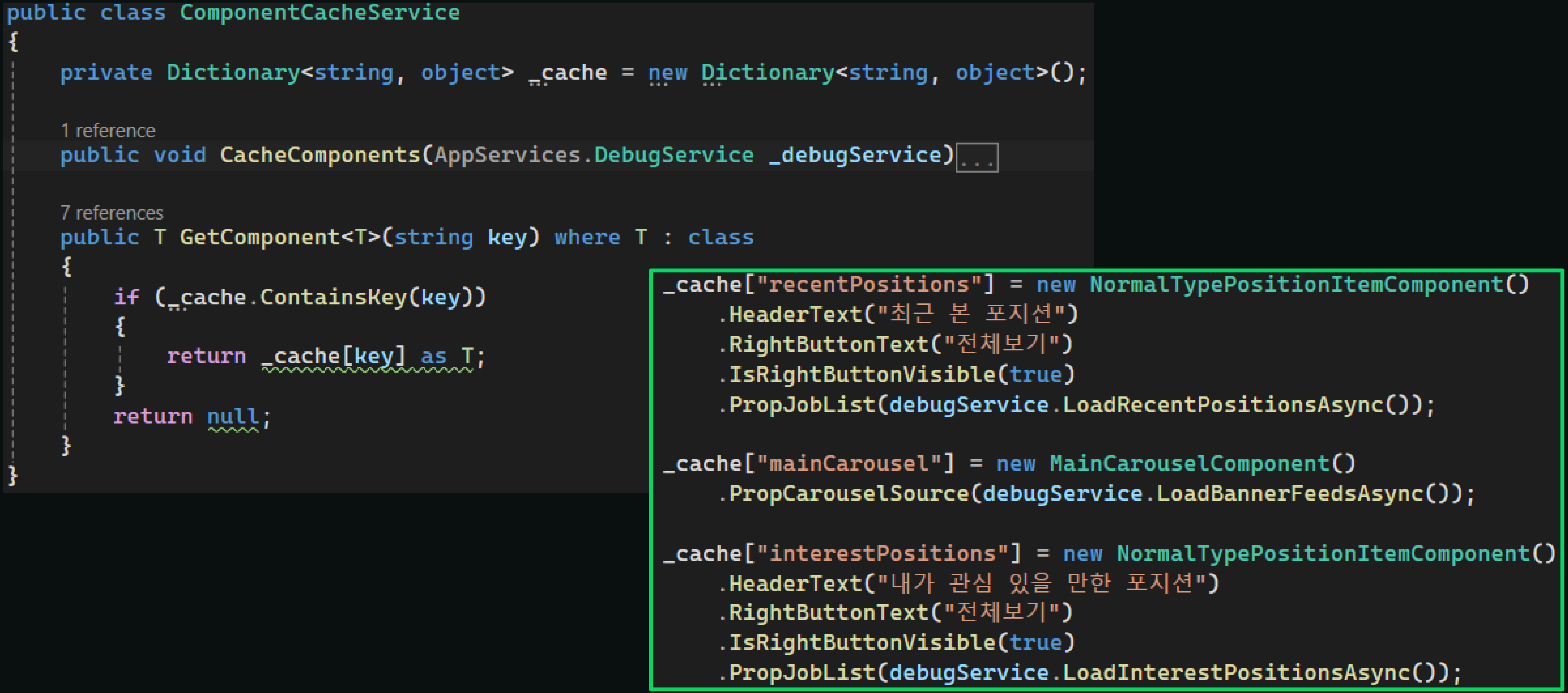
팀장급 시니어 개발자
리클(Recl)
AI 예측 90%

개발 리드 (Tech Lead)
아이들랩
AI 예측 89%

대구연
씨엘모빌
AI 예측

- IsVisiblePrediction, PredictionTextColor,
  PredictionBackColor

TIP #18 메인 탭에 포함 된 페이지 컴포넌트 최적화

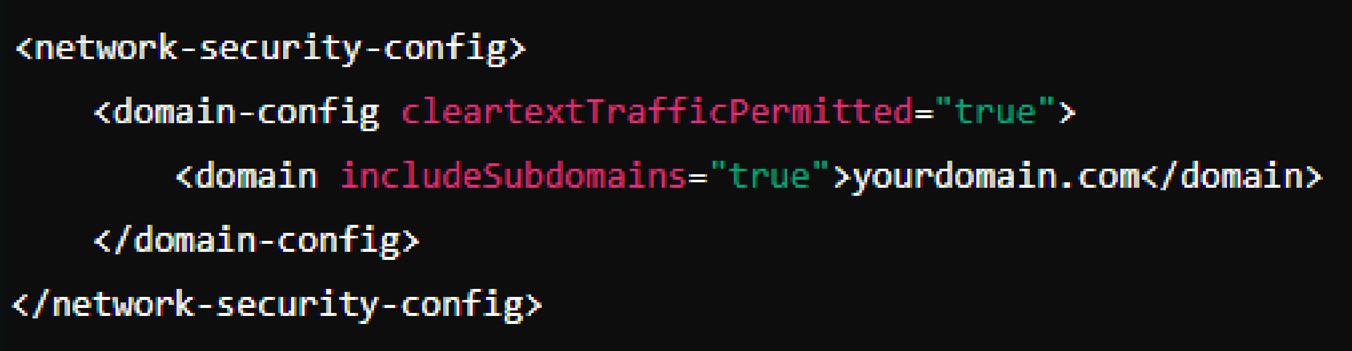


```
public class ComponentCacheService
{
    private Dictionary<string, object> _cache = new Dictionary<string, object>();

    1 reference
    public void CacheComponents(AppServices.DebugService _debugService)...

    7 references
    public T GetComponent<T>(string key) where T : class
    {
        if (_cache.ContainsKey(key))
        {
            return _cache[key] as T;
        }
        return null;
    }
}
```

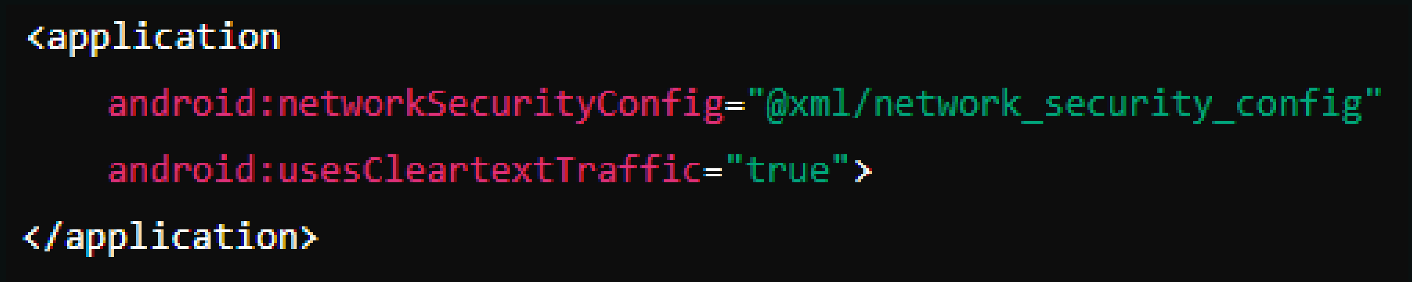
```
_cache["recentPositions"] = new NormalTypePositionItemComponent()
    .HeaderText("최근 본 포지션")
    .RightButtonText("전체보기")
    .IsRightButtonVisible(true)
    .PropJobList(debugService.LoadRecentPositionsAsync());

_cache["mainCarousel"] = new MainCarouselComponent()
    .PropCarouselSource(debugService.LoadBannerFeedsAsync());

_cache["interestPositions"] = new NormalTypePositionItemComponent()
    .HeaderText("내가 관심 있을 만한 포지션")
    .RightButtonText("전체보기")
    .IsRightButtonVisible(true)
    .PropJobList(debugService.LoadInterestPositionsAsync());
```

TIP #19 SSL 적용 이미지 다운로드 적용 방안 및 최적화

https://image.wanted.co.kr/optimize?src=https%3A%2F%2Fstatic.wanted.co.kr%2Fimages%2Fcompany%2F1243%2F8m4j0rzlzrcsbmoz__1080_790.jpg&w=338&q=100

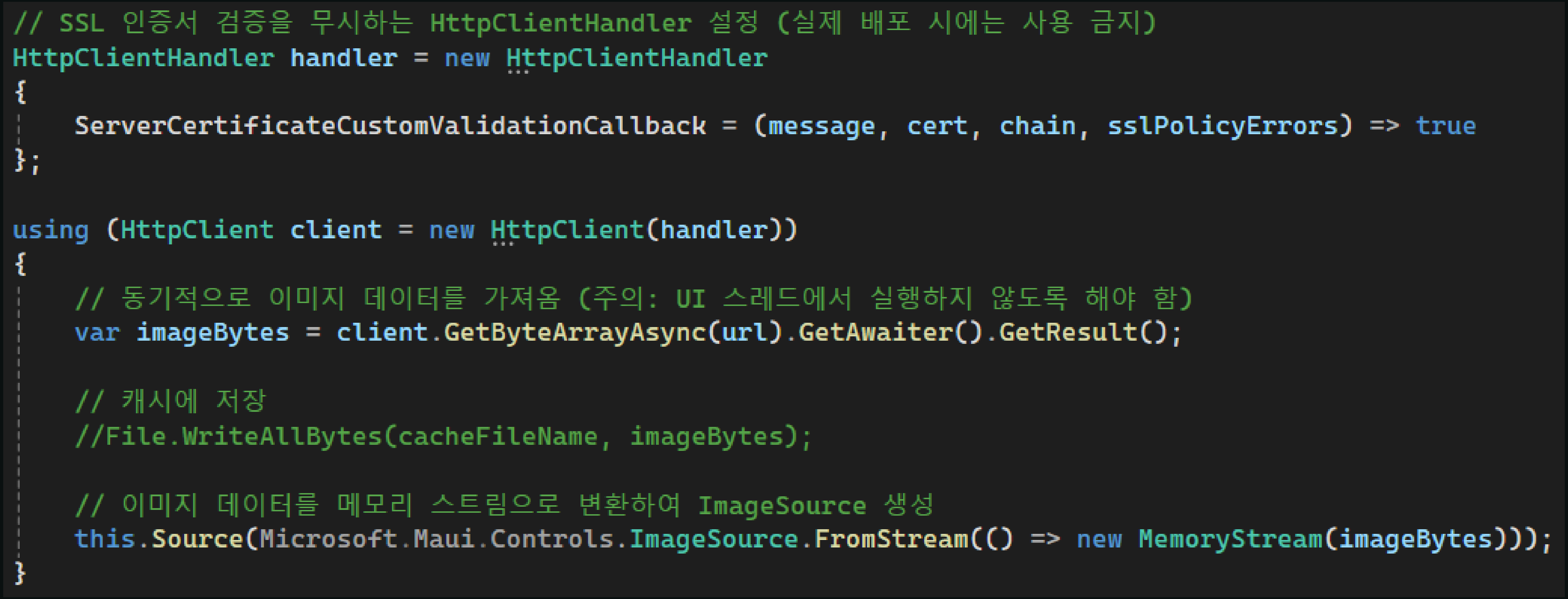


```
<network-security-config>
    <domain-config cleartextTrafficPermitted="true">
        <domain includeSubdomains="true">yourdomain.com</domain>
    </domain-config>
</network-security-config>
```



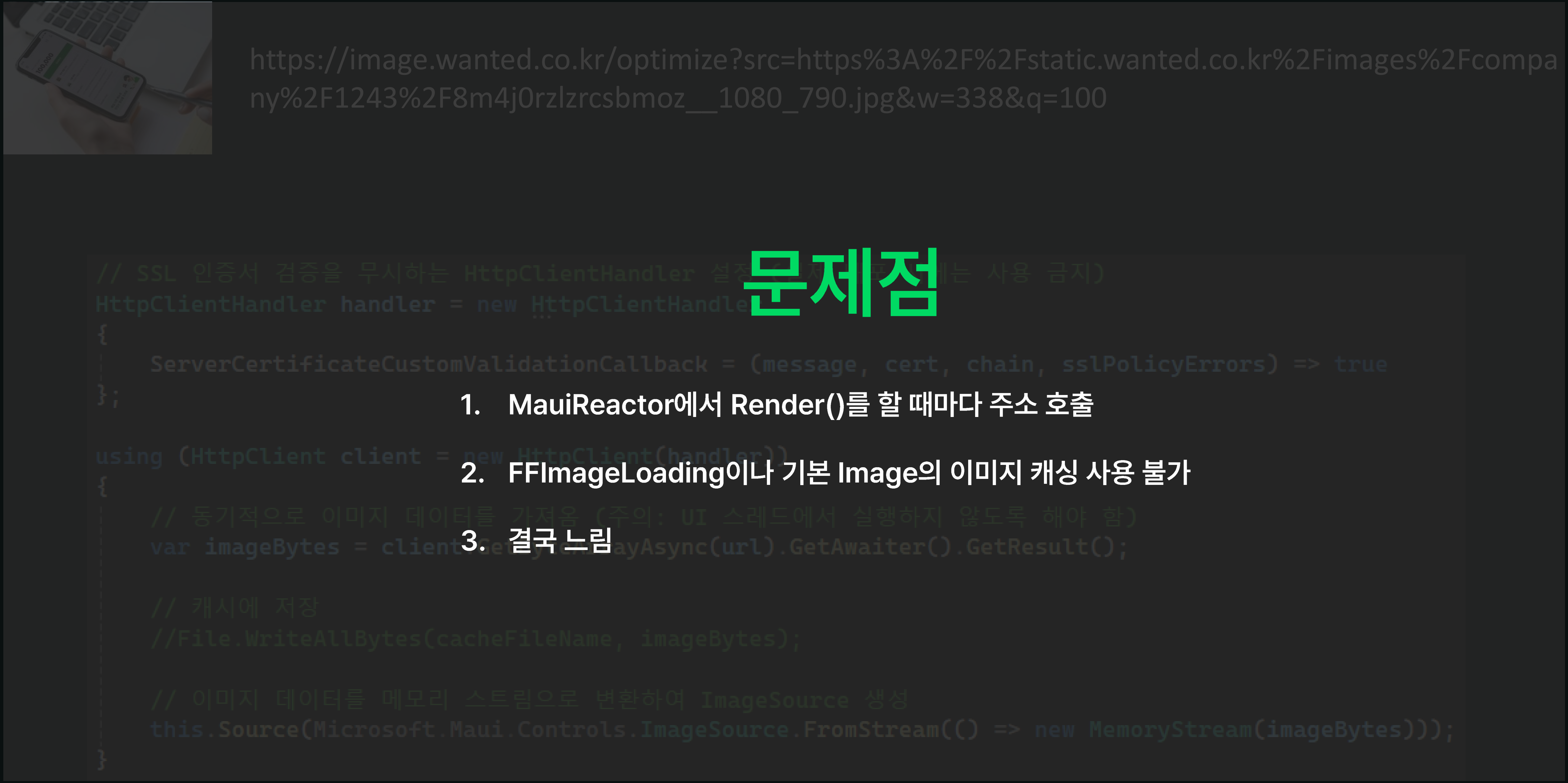
```
<application
    android:networkSecurityConfig="@xml/network_security_config"
    android:usesCleartextTraffic="true">
</application>
```

TIP #19 SSL 적용 이미지 다운로드 적용 방안 및 최적화



```
// SSL 인증서 검증을 무시하는 HttpClientHandler 설정 (실제 배포 시에는 사용 금지)
HttpClientHandler handler = new HttpClientHandler
{
    ServerCertificateCustomValidationCallback = (message, cert, chain, sslPolicyErrors) => true
};

using (HttpClient client = new HttpClient(handler))
{
    // 동기적으로 이미지 데이터를 가져옴 (주의: UI 스레드에서 실행하지 않도록 해야 함)
    var imageBytes = client.GetByteArrayAsync(url).GetAwaiter().GetResult();

    // 캐시에 저장
    //File.WriteAllBytes(cacheFileName, imageBytes);

    // 이미지 데이터를 메모리 스트림으로 변환하여 ImageSource 생성
    this.Source(Microsoft.Maui.Controls.ImageSource.FromStream(() => new MemoryStream(imageBytes)));
}
```

TIP #19 SSL 적용 이미지 다운로드 적용 방안 및 최적화



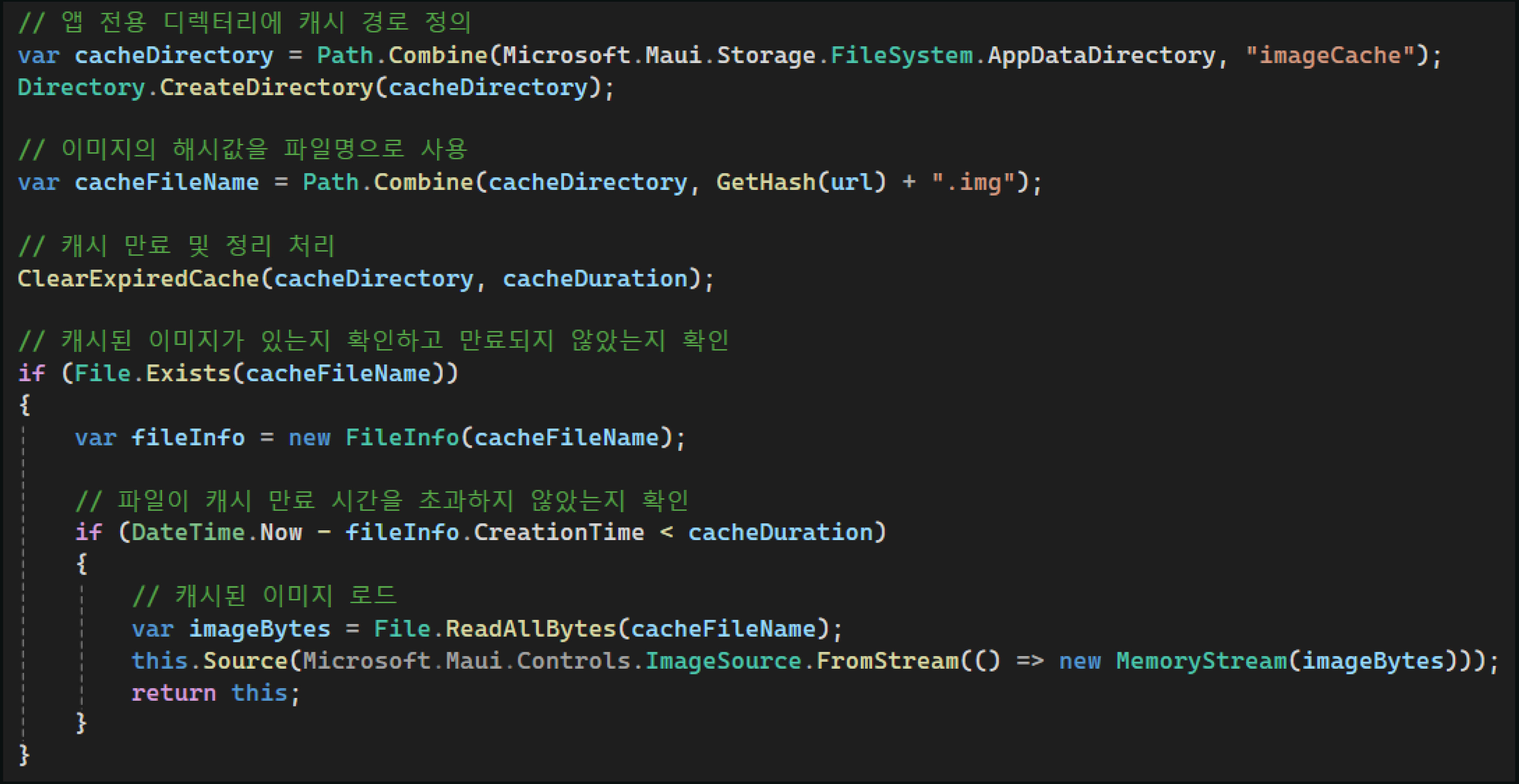
https://image.wanted.co.kr/optimize?src=https%3A%2F%2Fstatic.wanted.co.kr%2Fimages%2Fcompany%2F1243%2F8m4j0rzlzrcsbmoz__1080_790.jpg&w=338&q=100

```
// SSL 인증서 검증을 무시하는 HttpClientHandler 설정 (실제 환경에서는 사용 금지)
HttpClientHandler handler = new HttpClientHandler()
{
    ServerCertificateCustomValidationCallback = (message, cert, chain, sslPolicyErrors) => true
};

using (HttpClient client = new HttpClient(handler))
{
    // 동기적으로 이미지 데이터를 가져옴 (주의: UI 스레드에서 실행하지 않도록 해야 함)
    var imageBytes = client.GetByteArrayAsync(url).GetAwaiter().GetResult();

    // 캐시에 저장
    //File.WriteAllBytes(cacheFileName, imageBytes);

    // 이미지 데이터를 메모리 스트림으로 변환하여 ImageSource 생성
    this.Source(Microsoft.Maui.Controls.ImageSource.FromStream(() => new MemoryStream(imageBytes)));
}
```

문제점

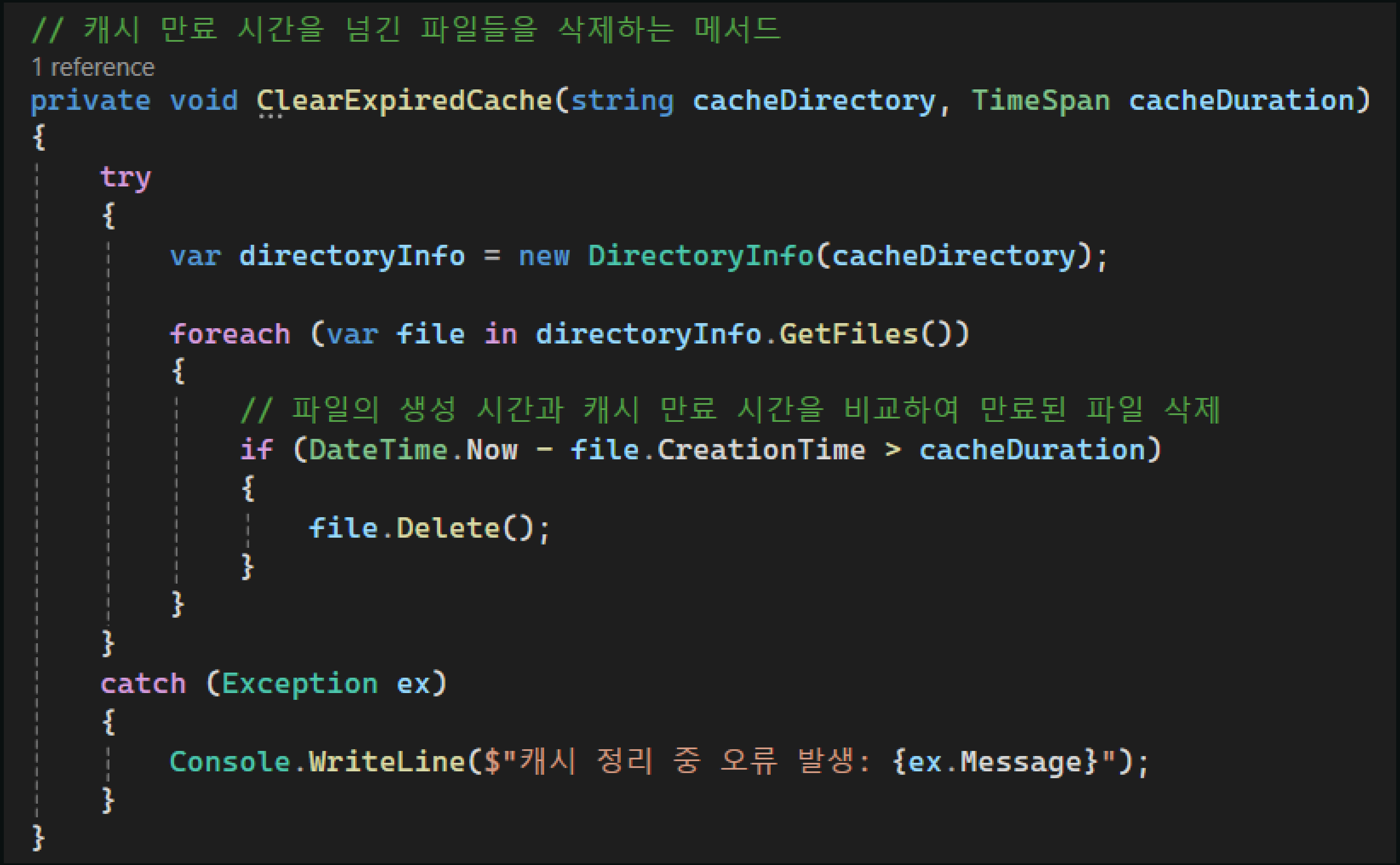
1. MauiReactor에서 Render()를 할 때마다 주소 호출
2. FFImageLoading이나 기본 Image의 이미지 캐싱 사용 불가
3. 결국 느림

TIP #19 SSL 적용 이미지 다운로드 적용 방안 및 최적화



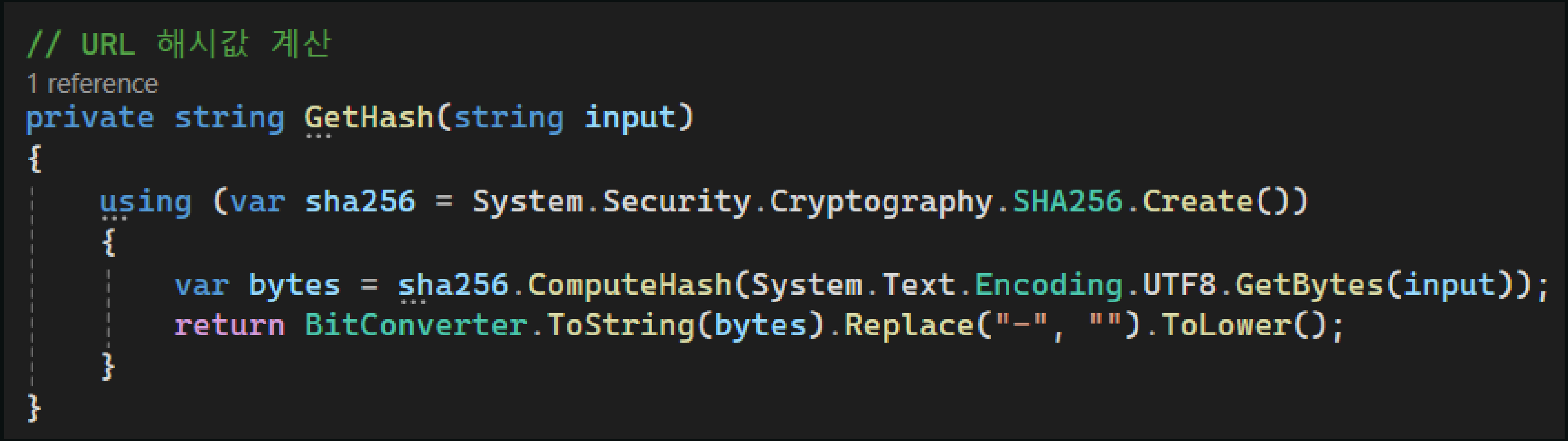
```csharp
// 앱 전용 디렉터리에 캐시 경로 정의
var cacheDirectory = Path.Combine(Microsoft.Maui.Storage.FileSystem.AppDataDirectory, "imageCache");
Directory.CreateDirectory(cacheDirectory);

// 이미지의 해시값을 파일명으로 사용
var cacheFileName = Path.Combine(cacheDirectory, GetHash(url) + ".img");

// 캐시 만료 및 정리 처리
ClearExpiredCache(cacheDirectory, cacheDuration);

// 캐시된 이미지가 있는지 확인하고 만료되지 않았는지 확인
if (File.Exists(cacheFileName))
{
    var fileInfo = new FileInfo(cacheFileName);

    // 파일이 캐시 만료 시간을 초과하지 않았는지 확인
    if (DateTime.Now - fileInfo.CreationTime < cacheDuration)
    {
        // 캐시된 이미지 로드
        var imageBytes = File.ReadAllBytes(cacheFileName);
        this.Source(Microsoft.Maui.Controls.ImageSource.FromStream(() => new MemoryStream(imageBytes)));
        return this;
    }
}
```

TIP #19 SSL 적용 이미지 다운로드 적용 방안 및 최적화



```csharp
// 캐시 만료 시간을 넘긴 파일들을 삭제하는 메서드
1 reference
private void ClearExpiredCache(string cacheDirectory, TimeSpan cacheDuration)
{
    try
    {
        var directoryInfo = new DirectoryInfo(cacheDirectory);

        foreach (var file in directoryInfo.GetFiles())
        {
            // 파일의 생성 시간과 캐시 만료 시간을 비교하여 만료된 파일 삭제
            if (DateTime.Now - file.CreationTime > cacheDuration)
            {
                file.Delete();
            }
        }
    }
    catch (Exception ex)
    {
        Console.WriteLine($"캐시 정리 중 오류 발생: {ex.Message}");
    }
}
```

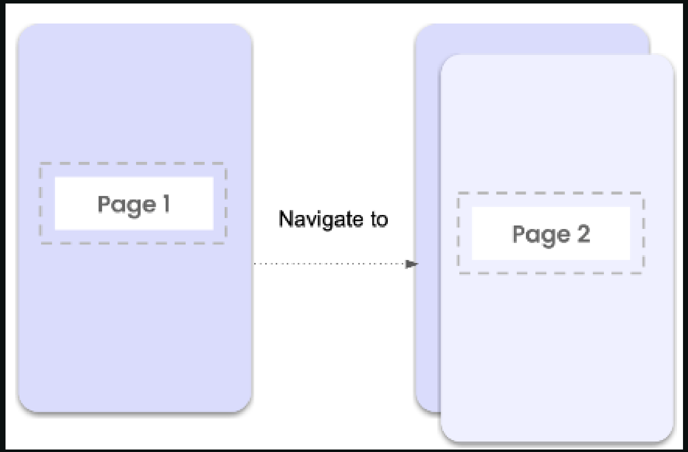
TIP #19 SSL 적용 이미지 다운로드 적용 방안 및 최적화



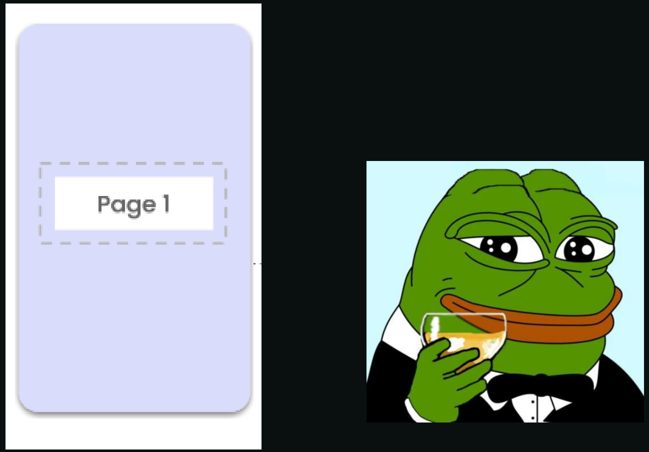
```
// URL 해시값 계산
1 reference
private string GetHash(string input)
{
    using (var sha256 = System.Security.Cryptography.SHA256.Create())
    {
        var bytes = sha256.ComputeHash(System.Text.Encoding.UTF8.GetBytes(input));
        return BitConverter.ToString(bytes).Replace("-", "").ToLower();
    }
}
```

TIP #20 Memory Stream은 가급적 피하자

MemoryStream



Navigation
Dispose()



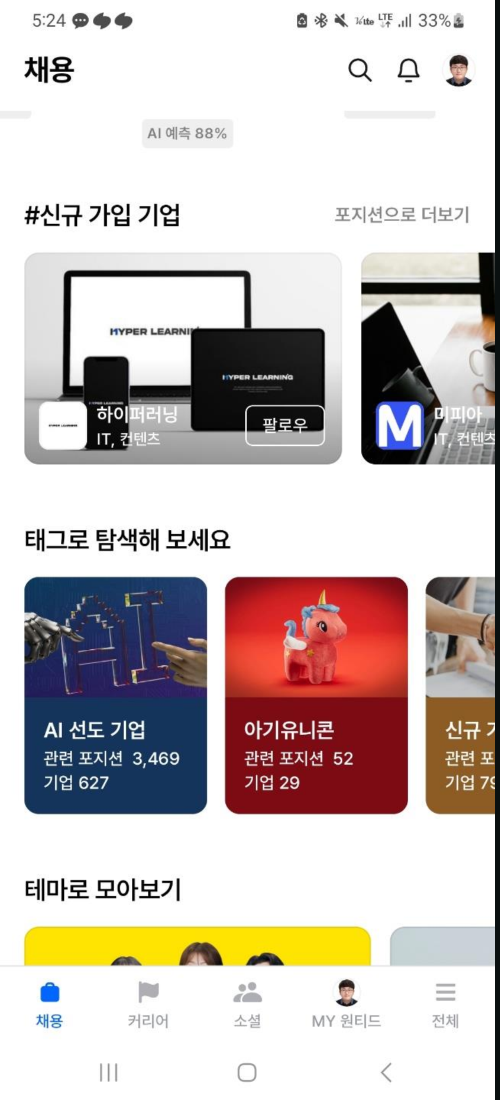
Render() + MemoryStream
GoBack()

TIP #21 레이아웃 간소화

```
.Set(Microsoft.Maui.Controls.CompressedLayout.IsHeadlessProperty, true)
```

Demo

짭티드 (= 원티드)



5:24 33%

채용

AI 예측 88%

#신규 가입 기업

포지션으로 더보기

하이퍼러닝
IT, 컨텐츠

팔로우

미피아
IT, 컨텐츠

태그로 탐색해 보세요

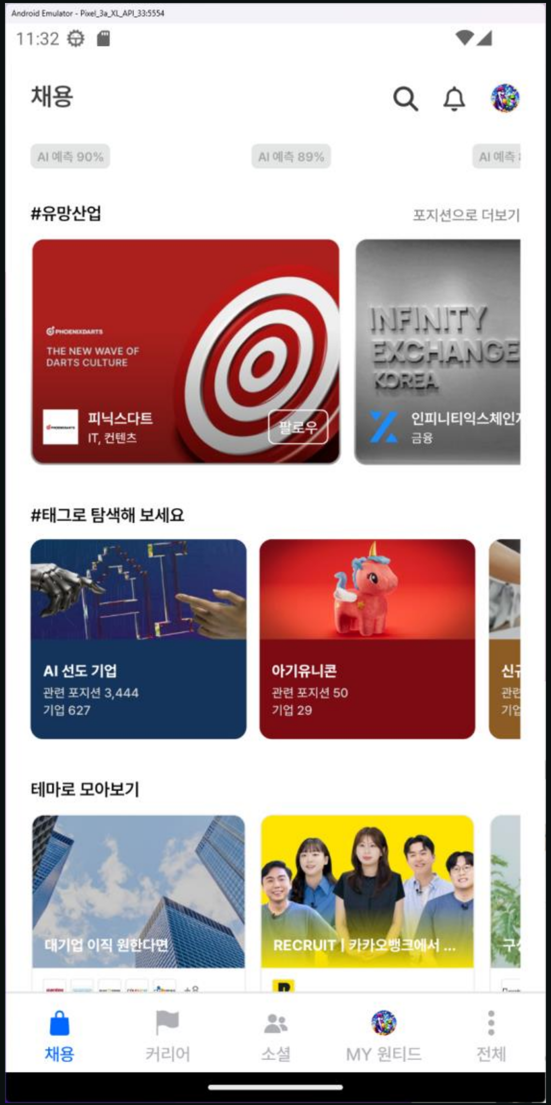
AI 선도 기업
관련 포지션 3,469
기업 627

아기유니콘
관련 포지션 52
기업 29

신규 7
관련 포
기업 79

테마로 모아보기

채용 커리어 소셜 MY 원티드 전체



Android Emulator - Pixel_3a_XL_API_33:5554

11:32

채용

AI 예측 90%

AI 예측 89%

AI 예측

#유망산업

포지션으로 더보기

THE NEW WAVE OF DARTS CULTURE

피닉스다트
IT, 컨텐츠

팔로우

INFINITY EXCHANGE KOREA

인피니티익스체인기
금융

#태그로 탐색해 보세요

AI 선도 기업
관련 포지션 3,444
기업 627

아기유니콘
관련 포지션 50
기업 29

테마로 모아보기

대기업 이직 원한다면

RECRUIT | 카카오뱅크에서 ...

채용 커리어 소셜 MY 원티드 전체

만들다가 어려우시다면!

.NET MAUI 단톡방

https://open.kakao.com/o/g8IzoOm

닷넷데브

https://forum.dotnetdev.kr

유튜브

흑우마스터의 야매코딩