

.NET MAUI Entwicklung auf Manjaro Linux - Setup Guide

Eine vollständige Anleitung zur Einrichtung von .NET MAUI mit VS Code auf Manjaro Linux für Cross-Platform App-Entwicklung.

Voraussetzungen

- Manjaro Linux (aktuell)
- Internetverbindung
- Admin-Rechte (sudo)

Installation

1. Grundlegende Pakete installieren

```
bash

# System aktualisieren
sudo pacman -Syu

# .NET 8 SDK installieren
sudo pacman -S dotnet-sdk

# Java für Android-Entwicklung
sudo pacman -S jdk11-openjdk

# Git (falls nicht vorhanden)
sudo pacman -S git

# yay AUR Helper (falls nicht installiert)
sudo pacman -S yay
```

2. Android-Entwicklungsumgebung

```
bash

# Android Studio installieren (empfohlen)
yay -S android-studio

# Alternative: Nur Android SDK Tools
# yay -S android-sdk android-sdk-platform-tools android-sdk-build-tools
```

3. .NET MAUI Workload

```
bash

# MAUI Workload installieren
dotnet workload install maui

# Installation überprüfen
dotnet workload list
```

Konfiguration

Umgebungsvariablen einrichten

Öffne deine Shell-Konfigurationsdatei:

```
bash

nano ~/.bashrc
# oder für zsh-Nutzer
nano ~/.zshrc
```

Füge folgende Zeilen hinzu:

```
bash

# Android SDK Pfade
export ANDROID_HOME=$HOME/Android/Sdk
export PATH=$PATH:$ANDROID_HOME/tools:$ANDROID_HOME/platform-tools

# Java Home
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk

# .NET Tools
export PATH=$PATH:$HOME/.dotnet/tools
```

Konfiguration neu laden:

```
bash

source ~/.bashrc
# oder
source ~/.zshrc
```

VS Code Setup

Extensions installieren

1. VS Code öffnen
2. Extensions Tab (Ctrl+Shift+X)
3. Folgende Extensions installieren:
 - **C# Dev Kit** (Microsoft)
 - **.NET MAUI** (Microsoft)
 - **C#** (wird automatisch installiert)

Android Studio erstes Setup

```
bash

# Android Studio starten
android-studio
```

1. Setup Wizard durchlaufen
2. Android SDK installieren lassen
3. Mindestens ein Android Virtual Device (AVD) erstellen

Erstes Projekt erstellen

Neues MAUI-Projekt

```
bash

# Projektordner erstellen
mkdir ~/MauiProjekte
cd ~/MauiProjekte

# Neues MAUI-Projekt erstellen
dotnet new maui -n MeineMauiApp
cd MeineMauiApp

# Projekt in VS Code öffnen
code .
```

Projekt kompilieren und testen

```
bash

# Android-Version kompilieren
dotnet build -f net8.0-android

# App auf Android-Emulator starten
dotnet run -f net8.0-android
```

WebView Integration

HTML-Content in MAUI WebView

Beispiel für eine einfache WebView-Integration:

```
xml

<!-- MainPage.xaml -->
<ContentPage x:Class="MeineMauiApp.MainPage">
    <ScrollView>
        <VerticalStackLayout>
            <WebView x:Name="MyWebView"
                    HeightRequest="400"
                    Source="https://example.com" />
        </VerticalStackLayout>
    </ScrollView>
</ContentPage>
```

Lokales HTML laden

```
csharp

// MainPage.xaml.cs
public partial class MainPage : ContentPage
{
    public MainPage()
    {
        InitializeComponent();

        // HTML-String direkt laden
        var htmlContent = @"
<html>
<body>
    <h1>Hallo MAUI!</h1>
    <p>Das ist mein HTML-Content</p>
</body>
</html>";

        MyWebView.Source = new HtmlWebViewSource
        {
            Html = htmlContent
        };
    }
}
```

Nützliche Befehle

```
bash

# Verfügbare Targets anzeigen
dotnet build --help

# Alle installierten Workloads auflisten
dotnet workload list

# MAUI Templates anzeigen
dotnet new list maui

# Projekt für Android kompilieren
dotnet build -f net8.0-android

# Abhängigkeiten wiederherstellen
dotnet restore

# Projekt bereinigen
dotnet clean
```

Troubleshooting

Häufige Probleme

Android SDK nicht gefunden:

```
bash

# SDK-Pfad überprüfen
echo $ANDROID_HOME
ls $ANDROID_HOME

# Falls leer, Android Studio öffnen und SDK installieren
```

MAUI Workload Fehler:

```
bash

# Workload neu installieren
dotnet workload install maui --force
```

Java-Probleme:

```
bash

# Java-Version überprüfen
java -version
echo $JAVA_HOME
```

Emulator startet nicht:

```
bash

# AVD-Manager über Android Studio öffnen
# Oder command line:
$ANDROID_HOME/emulator/emulator -list-avds
$ANDROID_HOME/emulator/emulator -avd <avd-name>
```

Unterstützte Plattformen

Plattform	Entwicklung auf Linux	Status
Android	✅ Vollständig	Unterstützt
iOS	❌ Nicht möglich	Erfordert macOS
Windows	❌ Nicht möglich	Erfordert Windows
macOS	❌ Nicht möglich	Erfordert macOS

Weiterführende Ressourcen

- [.NET MAUI Dokumentation](#)
- [VS Code C# Dev Kit](#)
- [Android Developer Guide](#)
- [MAUI WebView Dokumentation](#)

Fertig!

Du kannst jetzt Cross-Platform Apps mit .NET MAUI entwickeln und HTML-Content nahtlos in deine Android-Apps integrieren. Viel Erfolg bei deinen Projekten!

Erstellt für Manjaro Linux / Letzte Aktualisierung: Juni 2025