Android CI Workflow Setup for Dioxus

Quick Start

1. **Create the workflow directory** (if it doesn't exist):
   mkdir -p .github/workflows
2. **Save the YAML file** as `.github/workflows/android-test.yml` in your repository root
3. **Commit and push**:
   git add .github/workflows/android-test.yml
   git commit -m "Add Android CI workflow"
   git push
4. **Check the Actions tab** on GitHub to see your workflow run

Configuration Details

Current Settings

- **Android API Level**: 36
- **Emulator**: Pixel_7_API_36
- **Package Name**: amp-android
- **Build Command**: `dx serve --package amp-android --android --headless`
- **Test Timeout**: 30 minutes
- **Trigger Events**: Push to main, Pull requests, Manual dispatch

What the Workflow Does

Step	Purpose
Checkout code	Clones your repository
Setup Rust	Installs Rust with aarch64-linux-android target
Cache Gradle	Speeds up builds with cached dependencies
Install Dioxus CLI	Installs dx command-line tool
Setup Android SDK	Installs Android SDK, NDK (r25c), and CMake
Create emulator	Sets up Pixel_7_API_36 AVD
Start emulator	Launches Android emulator with hardware acceleration
Install dependencies	Runs cargo fetch
Build Android APK	Runs your exact command: `dx serve --package amp-android --android --headless`
Run tests	Executes cargo test --target aarch64-linux-android
Check logs	Dumps emulator and device logs on failure
Upload artifacts	Saves test results and reports
Upload screenshots	Saves screenshots on failure
Stop emulator	Cleans up emulator process
Print summary	Shows final status

Workflow Triggers

The workflow automatically runs when:

- ✅ Code is pushed to the `main` branch
- ✅ A pull request is opened targeting `main`
- ✅ You manually trigger it from the Actions tab (workflow_dispatch)

Customization Guide

Change the Android API Level

Edit `.github/workflows/android-test.yml` and modify:
env:
API_LEVEL: 36 # Change to your desired API level (21-36)

Change the Emulator Name

env:
EMULATOR_NAME: Pixel_7_API_36 # Change to any AVD name

Change the Package Name

Find this line:

- name: Build Android APK
  run: dx serve --package amp-android --android --headless
  Replace `amp-android` with your package name.

Adjust Build Timeout

timeout-minutes: 30 # Change to desired timeout in minutes

Modify Trigger Branches

Change the `on` section:
on:
push:
branches:
- main # Add or change branch names
pull_request:
branches:
- main
workflow_dispatch

Viewing Results

1. **On GitHub**: Go to your repository → Actions tab → Select the workflow run
2. **Console Output**: See real-time build logs
3. **Artifacts**: Download test results and screenshots from the Artifacts section
4. **Logs**: Check emulator and device logs if the build fails

Troubleshooting

"Emulator failed to start"

- The emulator typically takes 3-5 minutes to boot
- Ensure the API level matches your system-images
- Check Android SDK setup in logs

"dx serve command not found"

- Verify Dioxus CLI installed correctly
- Check Rust toolchain is set up properly

"aarch64-linux-android target not installed"

- The workflow automatically installs this
- If issues persist, check Rust toolchain step in logs

"Gradle cache issues"

- The workflow caches ~/.gradle/caches
- Clear cache if needed in GitHub Actions settings

Environment Variables

The workflow uses these environment variables (can be customized):

ANDROID_SDK_VERSION: 34
API_LEVEL: 36
EMULATOR_NAME: Pixel_7_API_36

File Contents

The generated `android-test.yml` file contains:

- Complete YAML configuration for GitHub Actions
- All necessary steps to build and test Android
- Error handling and artifact collection
- Emulator lifecycle management

Next Steps

1. Save the YAML file to `.github/workflows/android-test.yml`
2. Commit and push to your repository
3. Navigate to the Actions tab on GitHub
4. Watch your first Android CI run!

Support

For issues with:

- **Dioxus**: Check dioxus.dev
- **GitHub Actions**: See GitHub Actions docs
- **Android Emulator**: Check Android docs