

Developer Guide

1. Introduction

1.1 Aims

This document aims to provide comprehensive and detailed guidance for team members involved in Android software development, covering all aspects from environment construction to code specifications, function development, testing and deployment, to ensure that development work proceeds smoothly and produces high-quality applications.

1.2 Scope

The guide is applicable to the development of Android app of involving front-end, back-end and related interface development.

1.3 Target audience

Suitable for software engineers, development team leaders, testers and anyone involved in the mobile payment app development project. ”

2. Development Environment Construction

2.1 Hardware requirements

Processor: Intel Core i5 and above; Memory: 8GB and above
An Android phone that can be connected to a computer via usb

2.2 Development Tools

2.2.1 Visual Studio Code

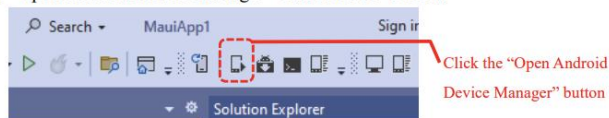
Downloading link: <https://code.visualstudio.com/>

2.2.2 Visual Studio 2022

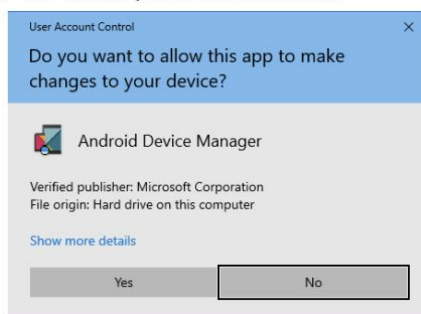
Downloading link: <https://visualstudio.microsoft.com/zh-hans/vs/>

2.2.3 Android emulator

Click the “Open Android Device Manager” button on the toolbar.



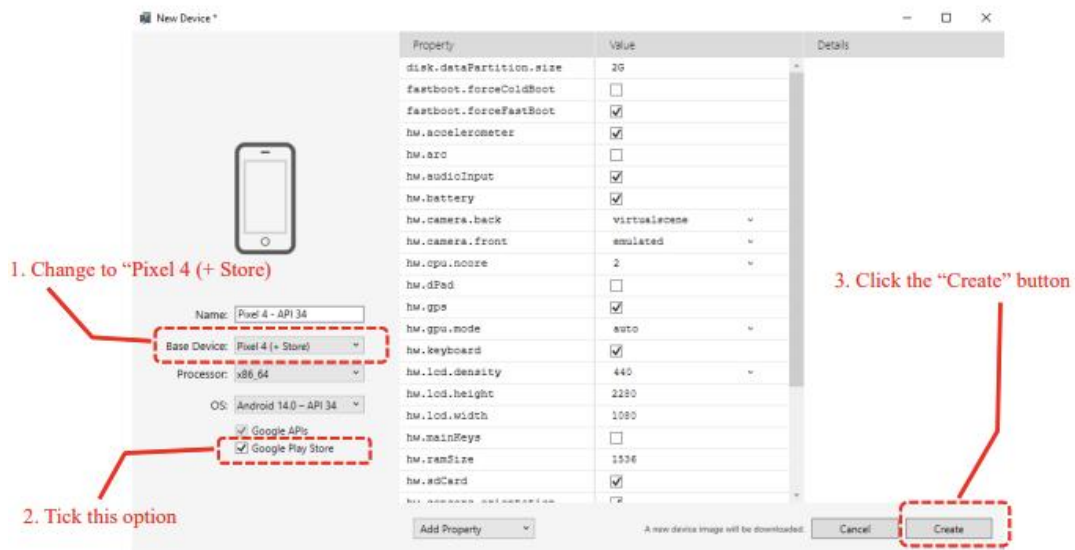
Click the “Yes” button if you see the screen below.



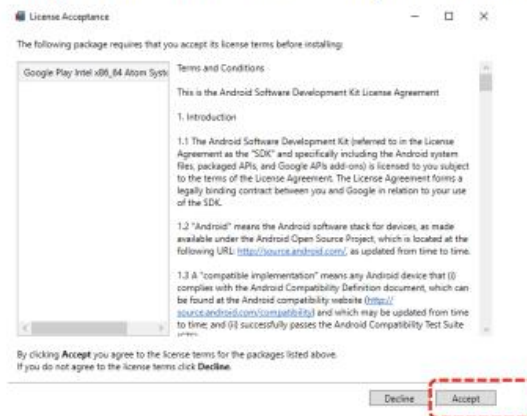
7. Click the “New” button on the toolbar.



8. Change the “Base Device” to “Pixel 4 (+ Store)”. Tick the “Google Play Store” option. Click the “Create” button to create an Android emulator.



9. Click the “Accept” button in the “License Acceptance” window.



2.3 Programming Language
Python (Django framework)
C# (.NET MAUI framework)

2.4 Database
MySQL

3. Function development

3.1 Account information modification

Using the website to modify <http://127.0.0.1:8000/admin/>

```
asset > admin.py
1 from django.contrib import admin
2 from .models import InventoryItem, Category
3
4 admin.site.register(InventoryItem)
5 admin.site.register(Category)
6
7 # Register your models here.
```

3.2 Adding the information to the database

```
from django.contrib import admin
from django.urls import path, include
from .views import Index, SignUpView, Dashboard, AddItem, EditItem, DeleteItem, InventoryItemViewSet
from django.contrib.auth import views as auth_views
from rest_framework.routers import DefaultRouter

router = DefaultRouter()
router.register(r'api', InventoryItemViewSet, basename='inventoryitem')

urlpatterns = [
    path('', Index.as_view(), name='index'),
    path('dashboard/', Dashboard.as_view(), name='dashboard'),
    path('add-item/', AddItem.as_view(), name='add-item'),
    path('edit-item/<int:pk>', EditItem.as_view(), name='edit-item'),
    path('delete-item/<int:pk>', DeleteItem.as_view(), name='delete-item'),
    path('signup/', SignUpView.as_view(), name='signup'),
    path('login/', auth_views.LoginView.as_view(template_name='asset/login.html'), name='login'),
    path('logout/', auth_views.LogoutView.as_view(template_name='asset/logout.html'), name='logout'),

    # Automatically include ViewSet routes
    path('', include(router.urls)),
]
```

3.3 Showing the information on the App

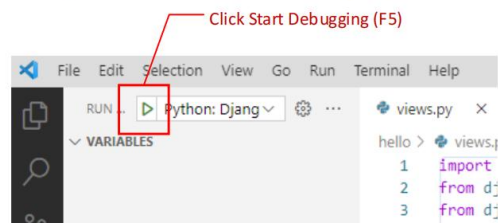
```
64 // Reset the item detail labels to default values
65 LabelProduct.Text = "Name: ";
66 LabelBrand.Text = "Artist: ";
67 LabelIngredients.Text = "Description: ";
68 LabelCategories.Text = "Dimensions: ";
69 LabelMessage.Text = string.Empty;
70 LabelMessage.TextColor = Colors.Black;
71 ImageCover.Source =
72     ImageSource.FromFile("image_coming_soon.png");
73
74
75 private void ParseInventoryJSON(string json)
76 {
77     try
78     {
79         // reset labels
80         LabelProduct.Text = "Name: ";
81         LabelBrand.Text = "Artist: ";
82         LabelIngredients.Text = "Description: ";
83         LabelCategories.Text = "Dimensions: ";
84         using (var jsonDocument = JsonDocument.Parse(json))
85         {
86             var root = jsonDocument.RootElement;
87
88             LabelProduct.Text = "Name: " + root.GetProperty("name").GetString();
89             LabelBrand.Text = "Artist: " + root.GetProperty("artist").GetString();
90             LabelIngredients.Text = "Description: " + root.GetProperty("description").GetString();
91             LabelCategories.Text = "Dimensions: " + root.GetProperty("dimension").GetString();
92
93             if (root.TryGetProperty("image", out var imageUrl))
94             {
95                 ImageCover.Source = ImageSource.FromUri(new Uri(imageUrl.GetString()));
96             }
97             else
98             {
99                 ImageCover.Source = ImageSource.FromFile("image_coming_soon.png");
100             }
101
102             LabelMessage.Text = "Artwork information loaded";
103             LabelMessage.TextColor = Colors.Green;
104         }
105     }
106     catch (Exception ex)
107     {
108         //
109     }
110 }
```

4. Testing

4.1 Unit testing

4.1.1 Testing Django framework

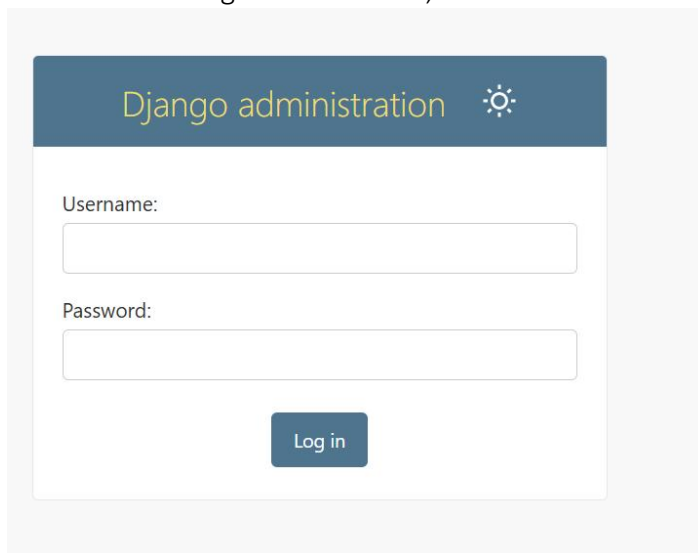
Start the debugger by clicking the [Start Debugging \(F5\)](#) button.



Open a browser and navigate to <http://127.0.0.1:8000/hello/VSCode>. Before the page renders, VS Code pauses the program at the breakpoint you set. The small yellow arrow on the breakpoint indicates that it's the next line of code to run.

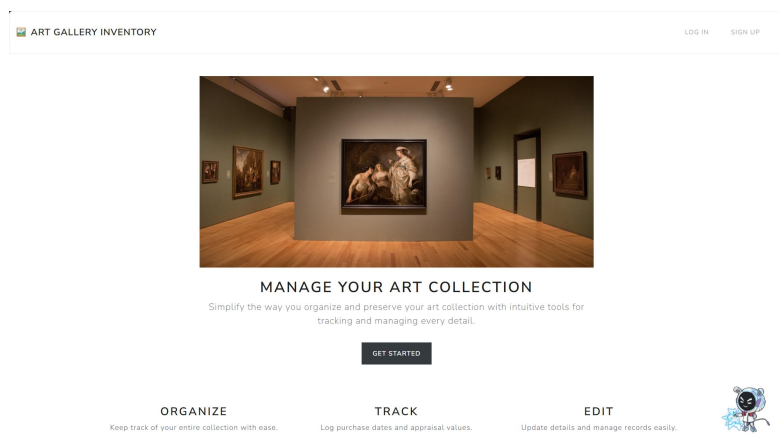
4.1.2 Testing administration

Enter <http://127.0.0.1:8000/admin/> in your browser. If the page displayed matches the image shown below, it indicates a successful setup.



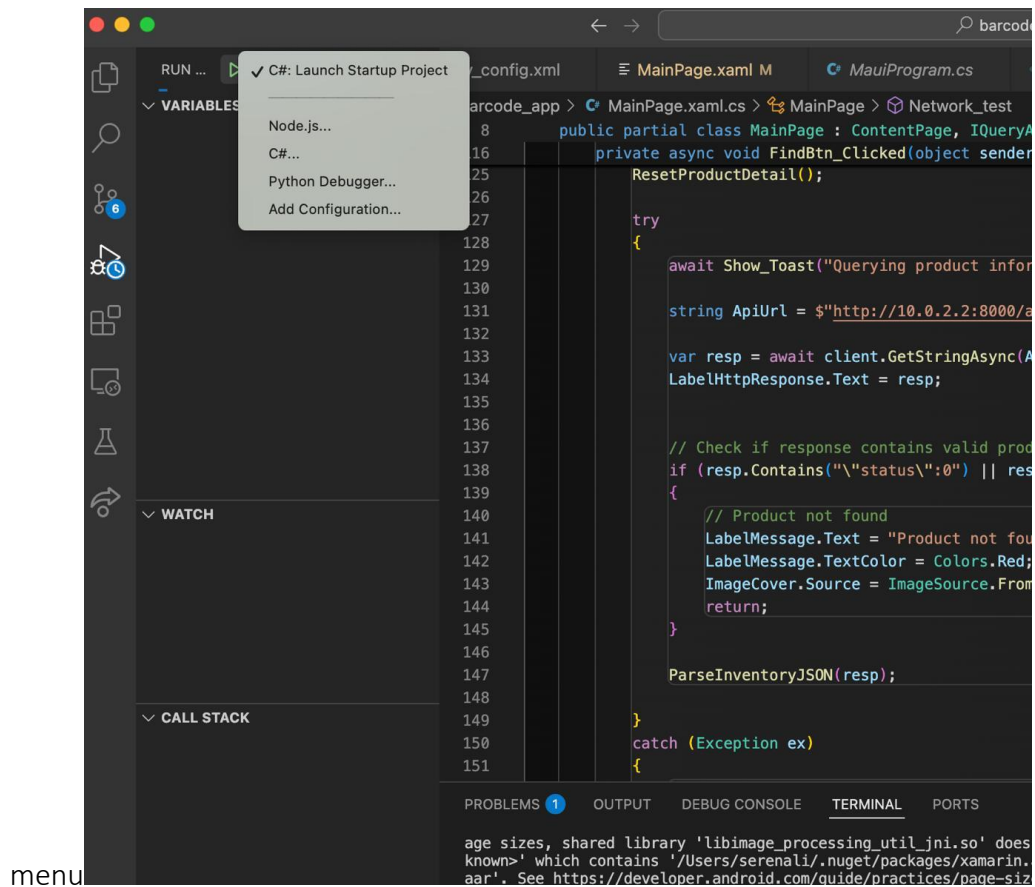
4.1.2 Testing website

Enter <http://127.0.0.1:8000/> in your browser. If the page displayed matches the image shown below, it indicates a successful setup

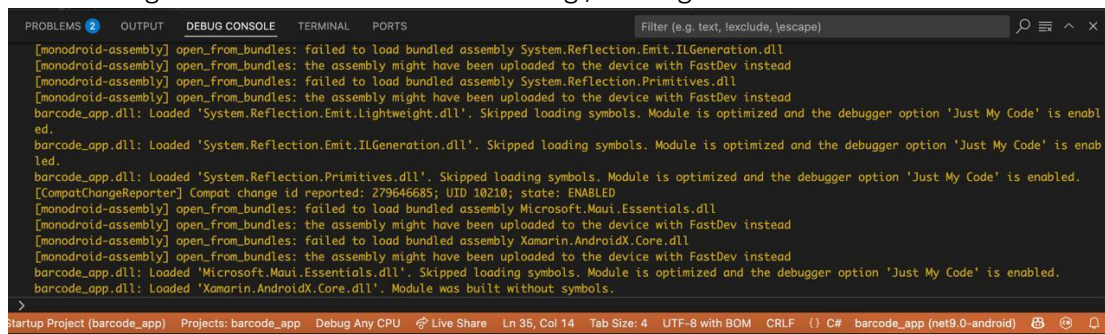


4.1.3 Testing android emulator running

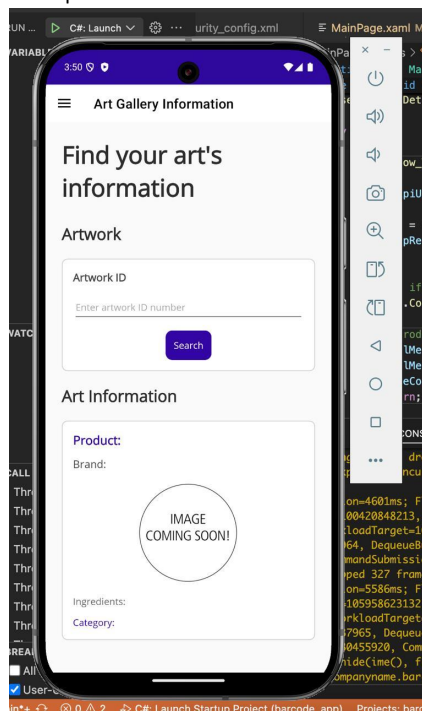
Run and debug android app on vs code go to 'Run and Debug' tab on the left-hand



Read debug console for errors when running / testing android emulator on vs code

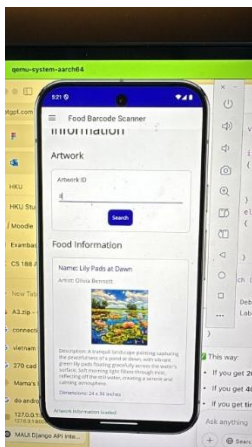


If the page displayed matches the image shown below, it indicates a successful setup.



4.2 System testing

Connect your Android phone to the laptop and run the app. If the connection and app operation succeed and match the image shown below, then it is considered a successful setup.



5. Deployment

5.1 Android deployment

APK formation

Type `dotnet publish -f:net9.0-android -c Release` to the terminal which form the APK to file. Then go to the file `-release` to find out the APK file

6. Maintenance and Updates

6.1 Troubleshooting and repair

6.1.1 Cannot login during the first time

LOG IN

- Please enter a correct username and password. Note that both fields may be case-sensitive.

Username*

admins

Password*

SIGN IN

Don't have an account? [Sign Up Here](#)

Type the command `python manage.py createsuperuser` to create a initial admin account.

```
PS C:\Users\31916\Desktop\asset_management-main> python manage.py createsuperuser
Username (leave blank to use '31916'): admins
Email address: 12@12.com
Password:
Password (again):
This password is too short. It must contain at least 8 characters.
This password is too common.
This password is entirely numeric.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
```

6.2.2 Cannot connect to Android phone

Running debug model on the Android phone