《智能移动开发》实验报告

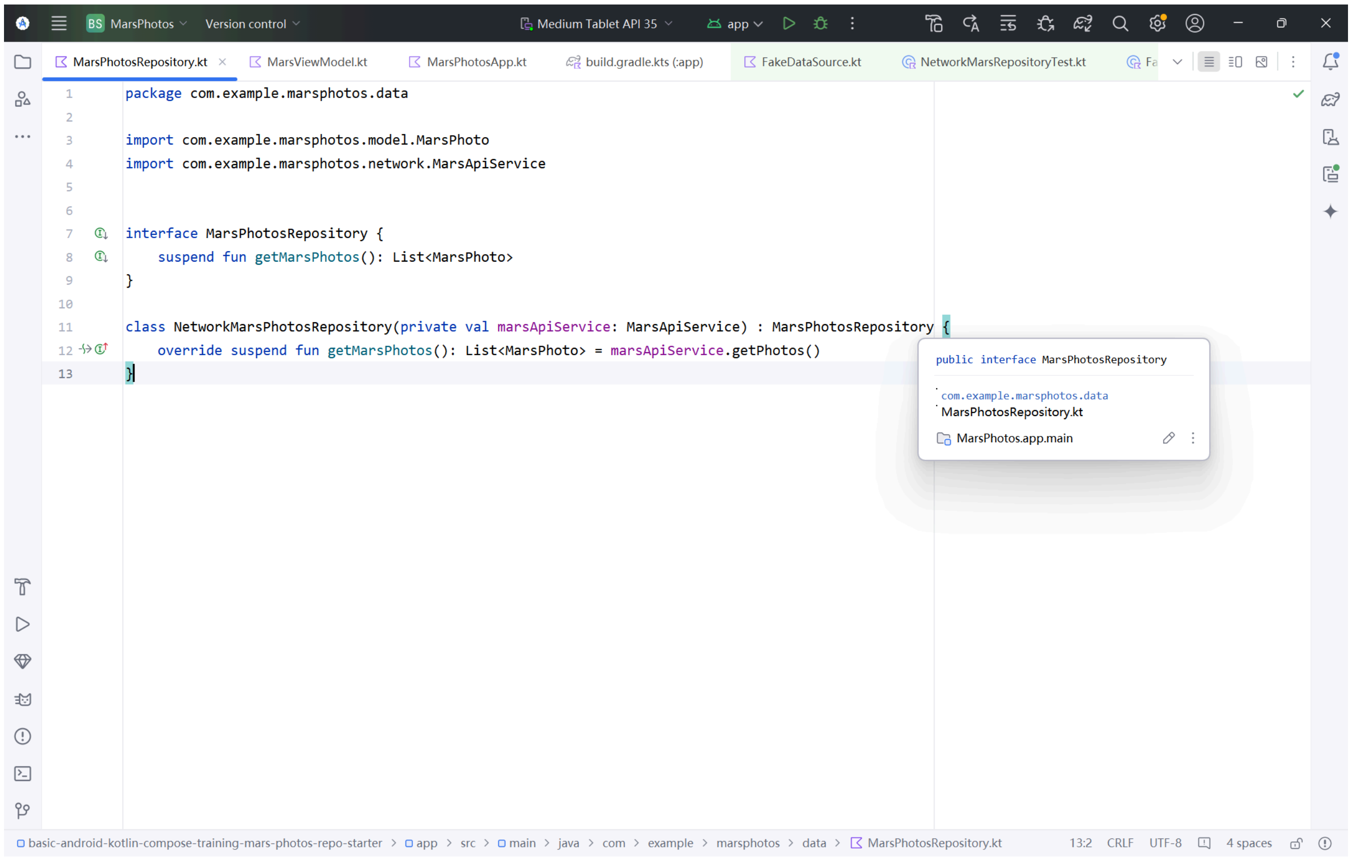
第十二次实验：从互联网加载和显示图片

学号：2212195

姓名：乔昊

添加存储库和手动依赖项注入

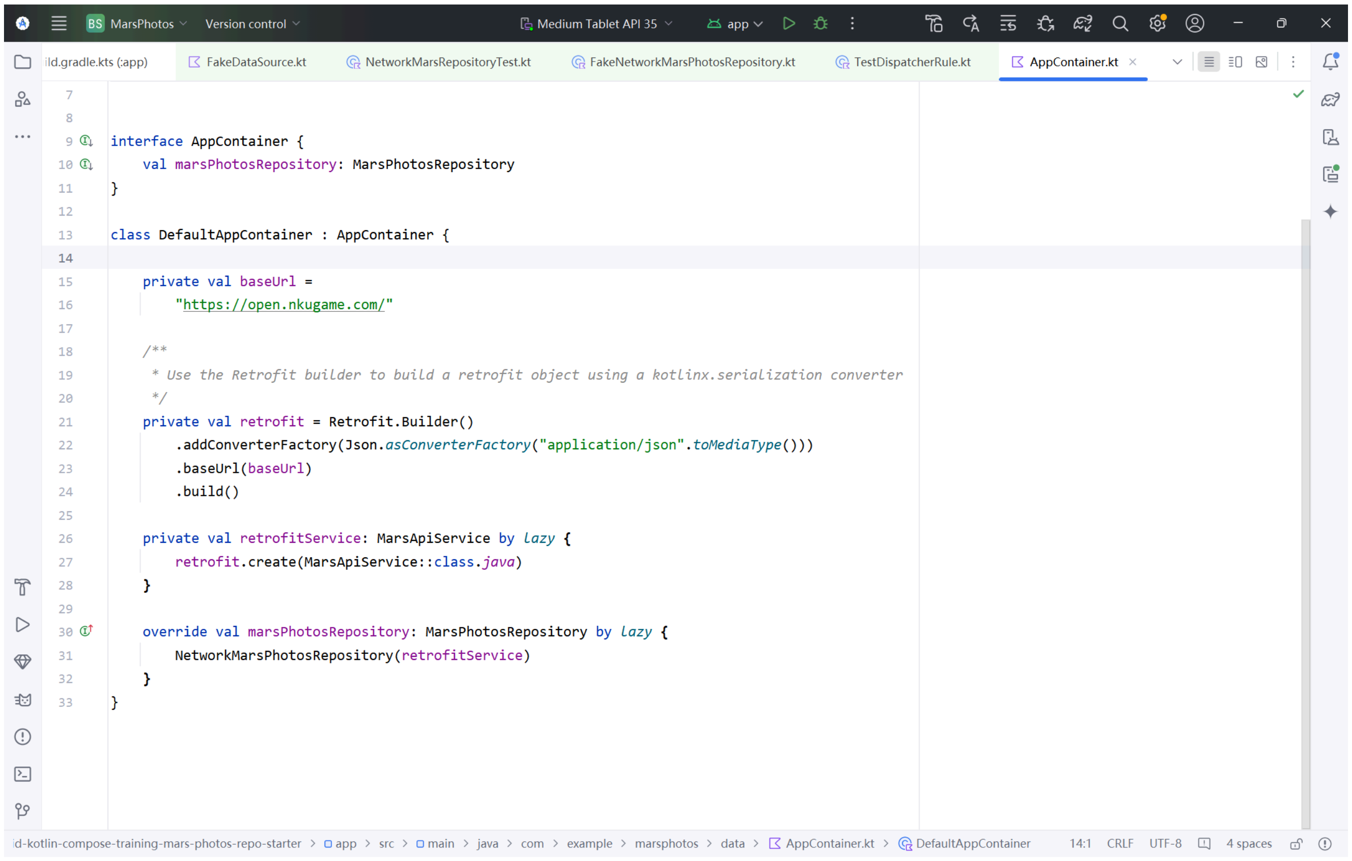
创建数据层



```
package com.example.marsphotos.data

import com.example.marsphotos.model.MarsPhoto
import com.example.marsphotos.network.MarsApiService


interface MarsPhotosRepository {
    suspend fun getMarsPhotos(): List<MarsPhoto>
}

class NetworkMarsPhotosRepository(private val marsApiService: MarsApiService) : MarsPhotosRepository {
    override suspend fun getMarsPhotos(): List<MarsPhoto> = marsApiService.getPhotos()
}
```

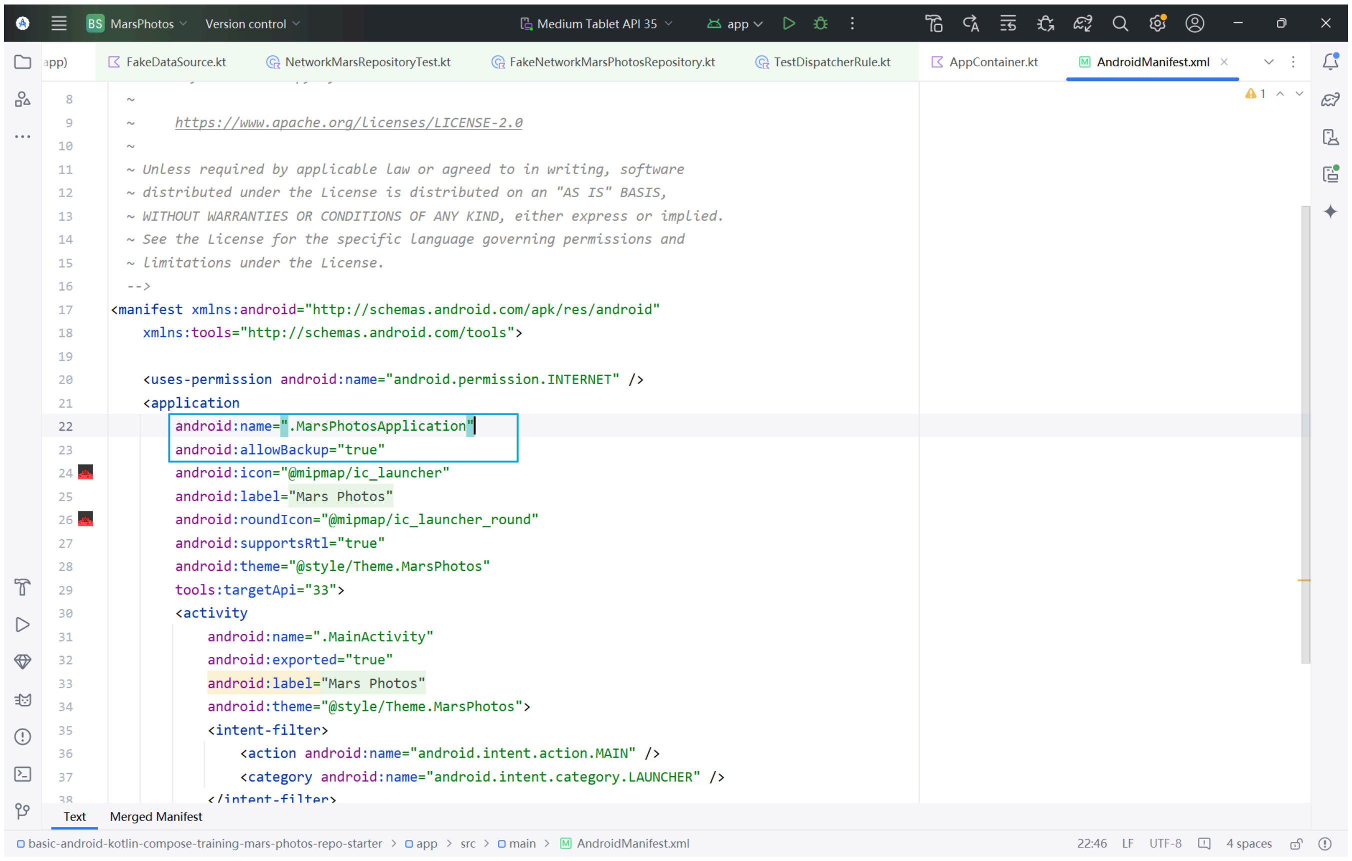
MarsPhotos
Version control
Medium Tablet API 35
app
MarsPhotosRepository.kt
MarsViewModel.kt
MarsPhotosApp.kt
build.gradle.kts (:app)
FakeDataSource.kt
NetworkMarsRepositoryTest.kt
public interface MarsPhotosRepository
com.example.marsphotos.data
MarsPhotosRepository.kt
MarsPhotos.app.main
basic-android-kotlin-compose-training-mars-photos-repo-starter > app > src > main > java > com > example > marsphotos > data > MarsPhotosRepository.kt
13:2 CRLF UTF-8 4 spaces

依赖项注入



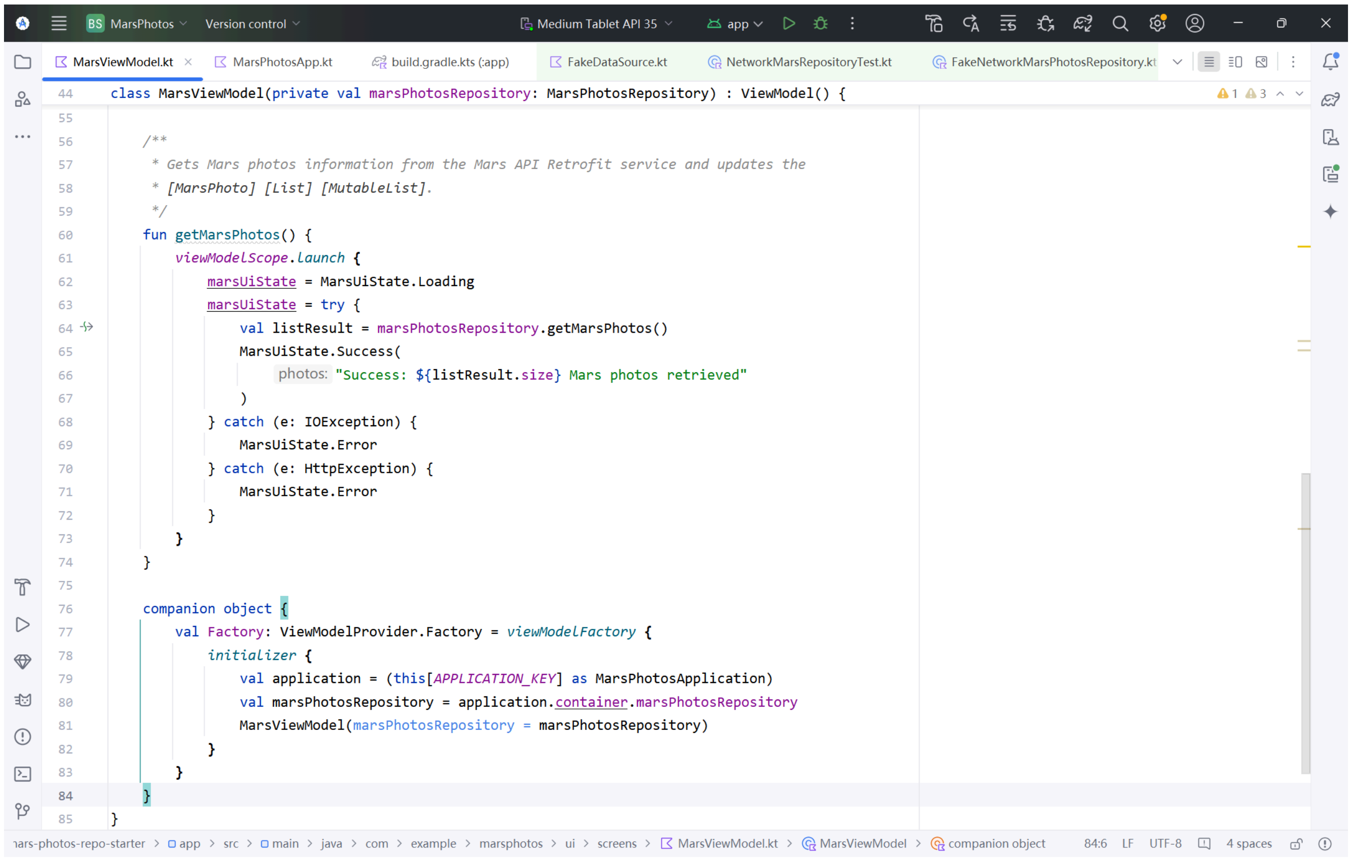
```kotlin


interface AppContainer {
    val marsPhotosRepository: MarsPhotosRepository
}

class DefaultAppContainer : AppContainer {

    private val baseUrl =
        "https://open.nkugame.com/"

    /**
     * Use the Retrofit builder to build a retrofit object using a kotlinx.serialization converter
     */
    private val retrofit = Retrofit.Builder()
        .addConverterFactory(Json.asConverterFactory("application/json".toMediaType()))
        .baseUrl(baseUrl)
        .build()

    private val retrofitService: MarsApiService by lazy {
        retrofit.create(MarsApiService::class.java)
    }

    override val marsPhotosRepository: MarsPhotosRepository by lazy {
        NetworkMarsPhotosRepository(retrofitService)
    }
}
```

添加容器至应用


MarsPhotos
Version control
Medium Tablet API 35
app
FakeDataSource.kt
NetworkMarsRepositoryTest.kt
FakeNetworkMarsPhotosRepository.kt
TestDispatcherRule.kt
AppContainer.kt
AndroidManifest.xml
Text
Merged Manifest
basic-android-kotlin-compose-training-mars-photos-repo-starter > app > src > main > AndroidManifest.xml
22:46 LF UTF-8 4 spaces

```
  ~
  ~     https://www.apache.org/licenses/LICENSE-2.0
  ~
  ~ Unless required by applicable law or agreed to in writing, software
  ~ distributed under the License is distributed on an "AS IS" BASIS,
  ~ WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  ~ See the License for the specific language governing permissions and
  ~ limitations under the License.
  -->
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <uses-permission android:name="android.permission.INTERNET" />
    <application
        android:name=".MarsPhotosApplication"
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Mars Photos"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.MarsPhotos"
        tools:targetApi="33">
        <activity
            android:name=".MainActivity"
            android:exported="true"
            android:label="Mars Photos"
            android:theme="@style/Theme.MarsPhotos">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
```

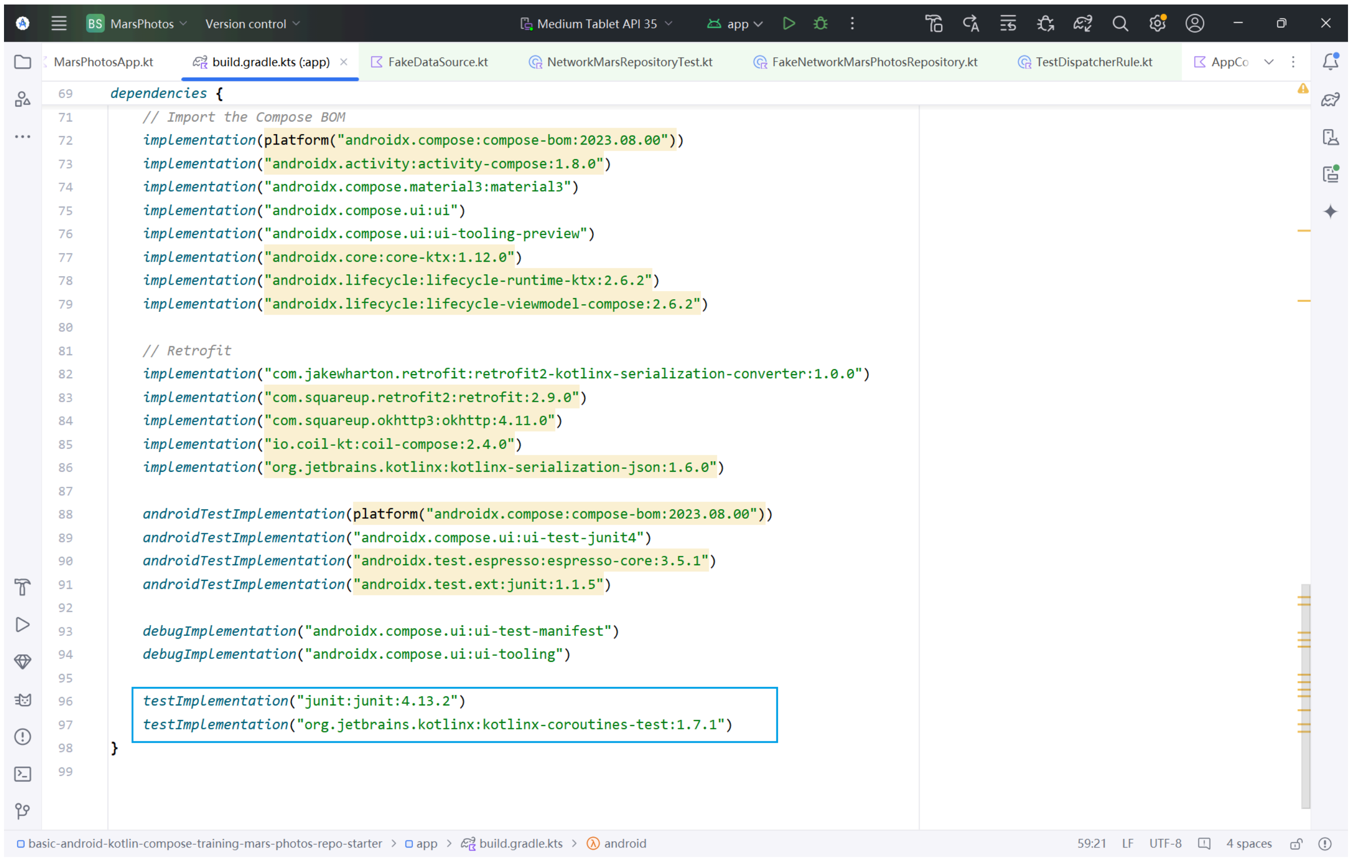
添加仓库至 ViewModel



```kotlin
class MarsViewModel(private val marsPhotosRepository: MarsPhotosRepository) : ViewModel() {

    /**
     * Gets Mars photos information from the Mars API Retrofit service and updates the
     * [MarsPhoto] [List] [MutableList].
     */
    fun getMarsPhotos() {
        viewModelScope.launch {
            marsUiState = MarsUiState.Loading
            marsUiState = try {
                val listResult = marsPhotosRepository.getMarsPhotos()
                MarsUiState.Success(
                     photos: "Success: ${listResult.size} Mars photos retrieved"
                )
            } catch (e: IOException) {
                MarsUiState.Error
            } catch (e: HttpException) {
                MarsUiState.Error
            }
        }
    }

    companion object {
        val Factory: ViewModelProvider.Factory = viewModelFactory {
            initializer {
                val application = (this[APPLICATION_KEY] as MarsPhotosApplication)
                val marsPhotosRepository = application.container.marsPhotosRepository
                MarsViewModel(marsPhotosRepository = marsPhotosRepository)
            }
        }
    }
}
```

MarsPhotos
Version control
Medium Tablet API 35
app
MarsViewModel.kt
MarsPhotosApp.kt
build.gradle.kts (:app)
FakeDataSource.kt
NetworkMarsRepositoryTest.kt
FakeNetworkMarsPhotosRepository.kt
ars-photos-repo-starter > app > src > main > java > com > example > marsphotos > ui > screens > MarsViewModel.kt > MarsViewModel > companion object
84:6 LF UTF-8 4 spaces

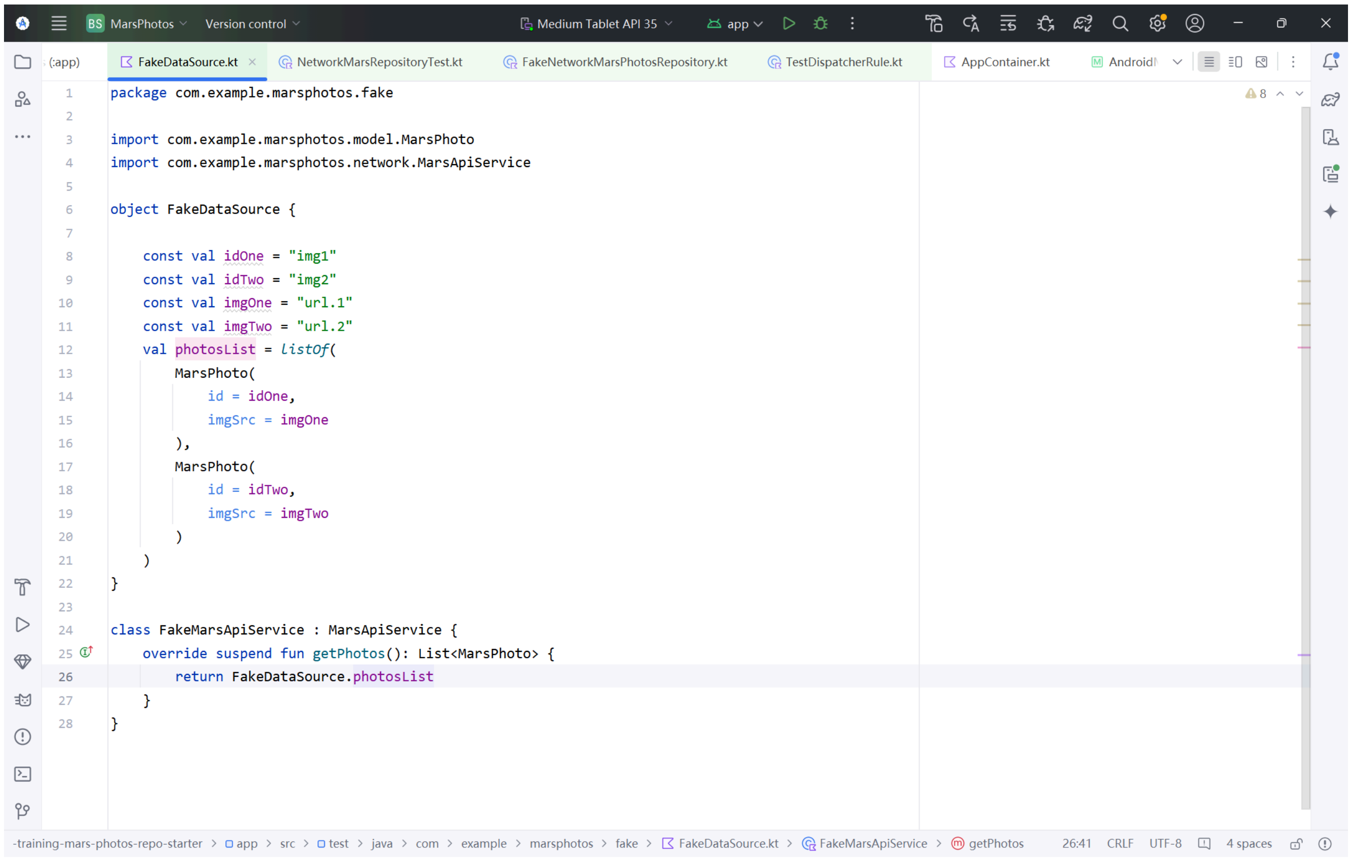
设置本地测试



```
dependencies {
    // Import the Compose BOM
    implementation(platform("androidx.compose:compose-bom:2023.08.00"))
    implementation("androidx.activity:activity-compose:1.8.0")
    implementation("androidx.compose.material3:material3")
    implementation("androidx.compose.ui:ui")
    implementation("androidx.compose.ui:ui-tooling-preview")
    implementation("androidx.core:core-ktx:1.12.0")
    implementation("androidx.lifecycle:lifecycle-runtime-ktx:2.6.2")
    implementation("androidx.lifecycle:lifecycle-viewmodel-compose:2.6.2")

    // Retrofit
    implementation("com.jakewharton.retrofit:retrofit2-kotlinx-serialization-converter:1.0.0")
    implementation("com.squareup.retrofit2:retrofit:2.9.0")
    implementation("com.squareup.okhttp3:okhttp:4.11.0")
    implementation("io.coil-kt:coil-compose:2.4.0")
    implementation("org.jetbrains.kotlinx:kotlinx-serialization-json:1.6.0")

    androidTestImplementation(platform("androidx.compose:compose-bom:2023.08.00"))
    androidTestImplementation("androidx.compose.ui:ui-test-junit4")
    androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")
    androidTestImplementation("androidx.test.ext:junit:1.1.5")

    debugImplementation("androidx.compose.ui:ui-test-manifest")
    debugImplementation("androidx.compose.ui:ui-tooling")

    testImplementation("junit:junit:4.13.2")
    testImplementation("org.jetbrains.kotlinx:kotlinx-coroutines-test:1.7.1")
}
```

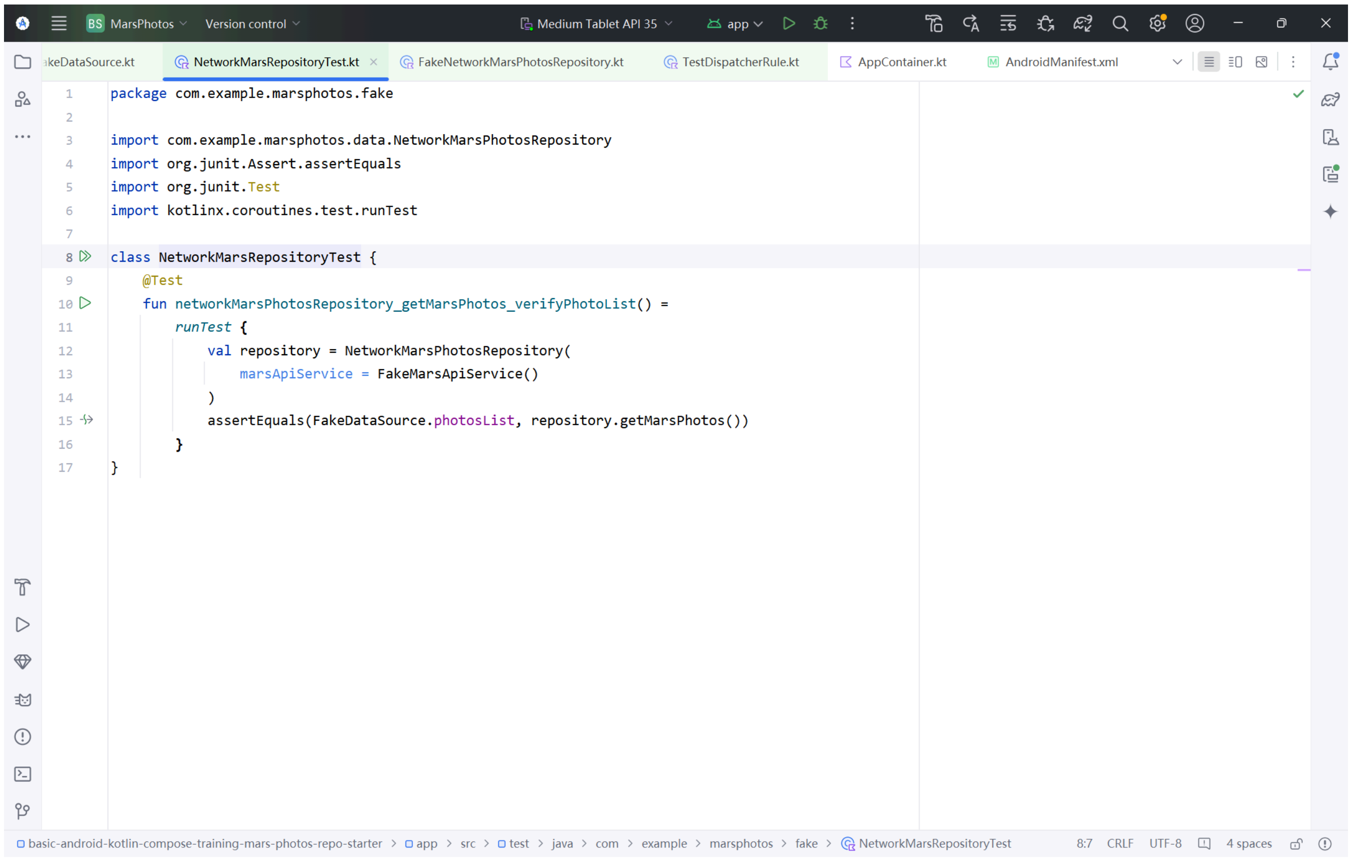
MarsPhotos
Version control
Medium Tablet API 35
app
MarsPhotosApp.kt
build.gradle.kts (:app)
FakeDataSource.kt
NetworkMarsRepositoryTest.kt
FakeNetworkMarsPhotosRepository.kt
TestDispatcherRule.kt
AppCo
basic-android-kotlin-compose-training-mars-photos-repo-starter > app > build.gradle.kts > android
59:21 LF UTF-8 4 spaces

创建虚构数据



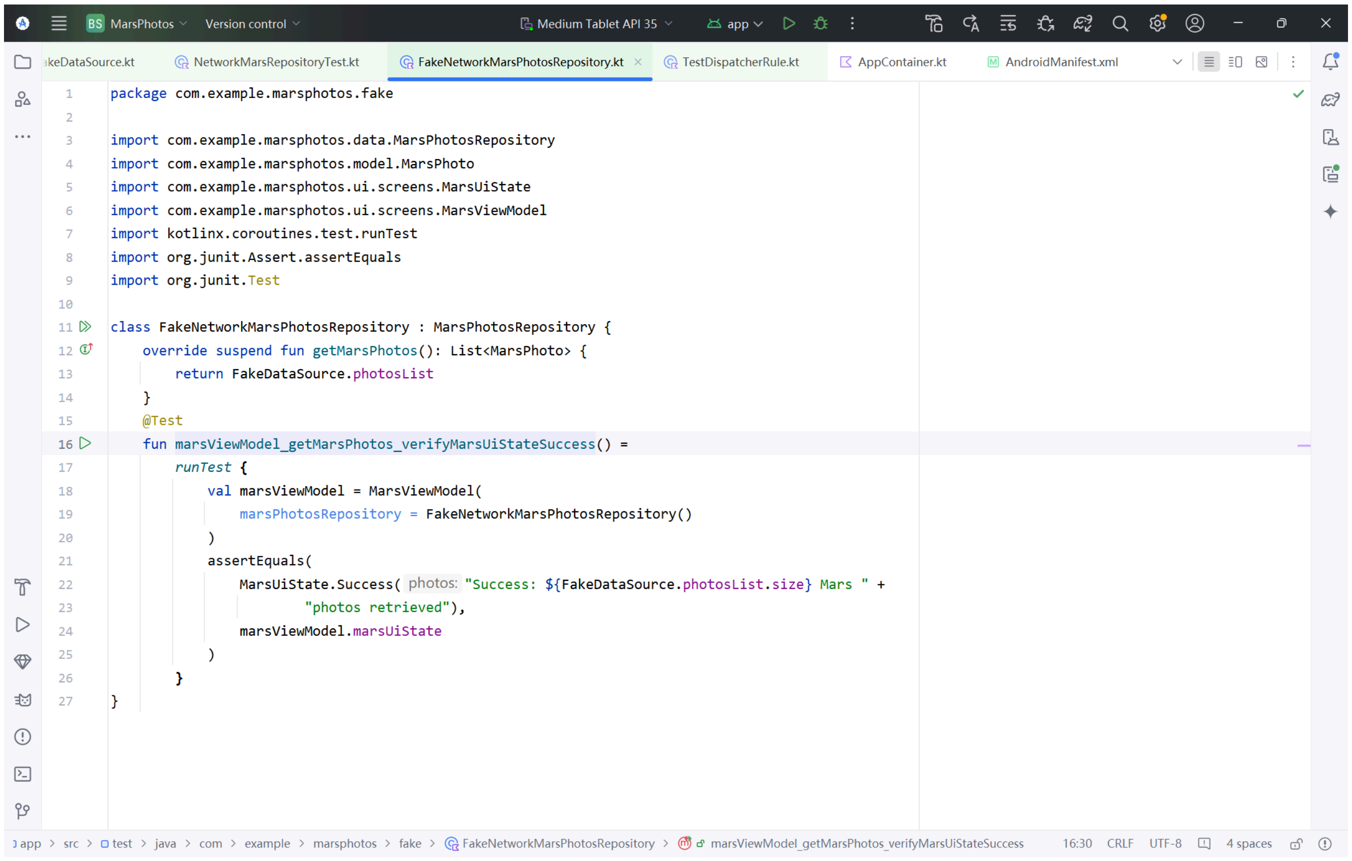
```kotlin
package com.example.marsphotos.fake

import com.example.marsphotos.model.MarsPhoto
import com.example.marsphotos.network.MarsApiService

object FakeDataSource {

    const val idOne = "img1"
    const val idTwo = "img2"
    const val imgOne = "url.1"
    const val imgTwo = "url.2"
    val photosList = listOf(
        MarsPhoto(
            id = idOne,
            imgSrc = imgOne
        ),
        MarsPhoto(
            id = idTwo,
            imgSrc = imgTwo
        )
    )
}

class FakeMarsApiService : MarsApiService {
    override suspend fun getPhotos(): List<MarsPhoto> {
        return FakeDataSource.photosList
    }
}
```

编写仓库测试


MarsPhotos
Version control
Medium Tablet API 35
app
keDataSource.kt
NetworkMarsRepositoryTest.kt
FakeNetworkMarsPhotosRepository.kt
TestDispatcherRule.kt
AppContainer.kt
AndroidManifest.xml
basic-android-kotlin-compose-training-mars-photos-repo-starter > app > src > test > java > com > example > marsphotos > fake > NetworkMarsRepositoryTest
8:7 CRLF UTF-8 4 spaces

```kotlin
package com.example.marsphotos.fake

import com.example.marsphotos.data.NetworkMarsPhotosRepository
import org.junit.Assert.assertEquals
import org.junit.Test
import kotlinx.coroutines.test.runTest

class NetworkMarsRepositoryTest {
    @Test
    fun networkMarsPhotosRepository_getMarsPhotos_verifyPhotoList() =
        runTest {
            val repository = NetworkMarsPhotosRepository(
                marsApiService = FakeMarsApiService()
            )
            assertEquals(FakeDataSource.photosList, repository.getMarsPhotos())
        }
}
```

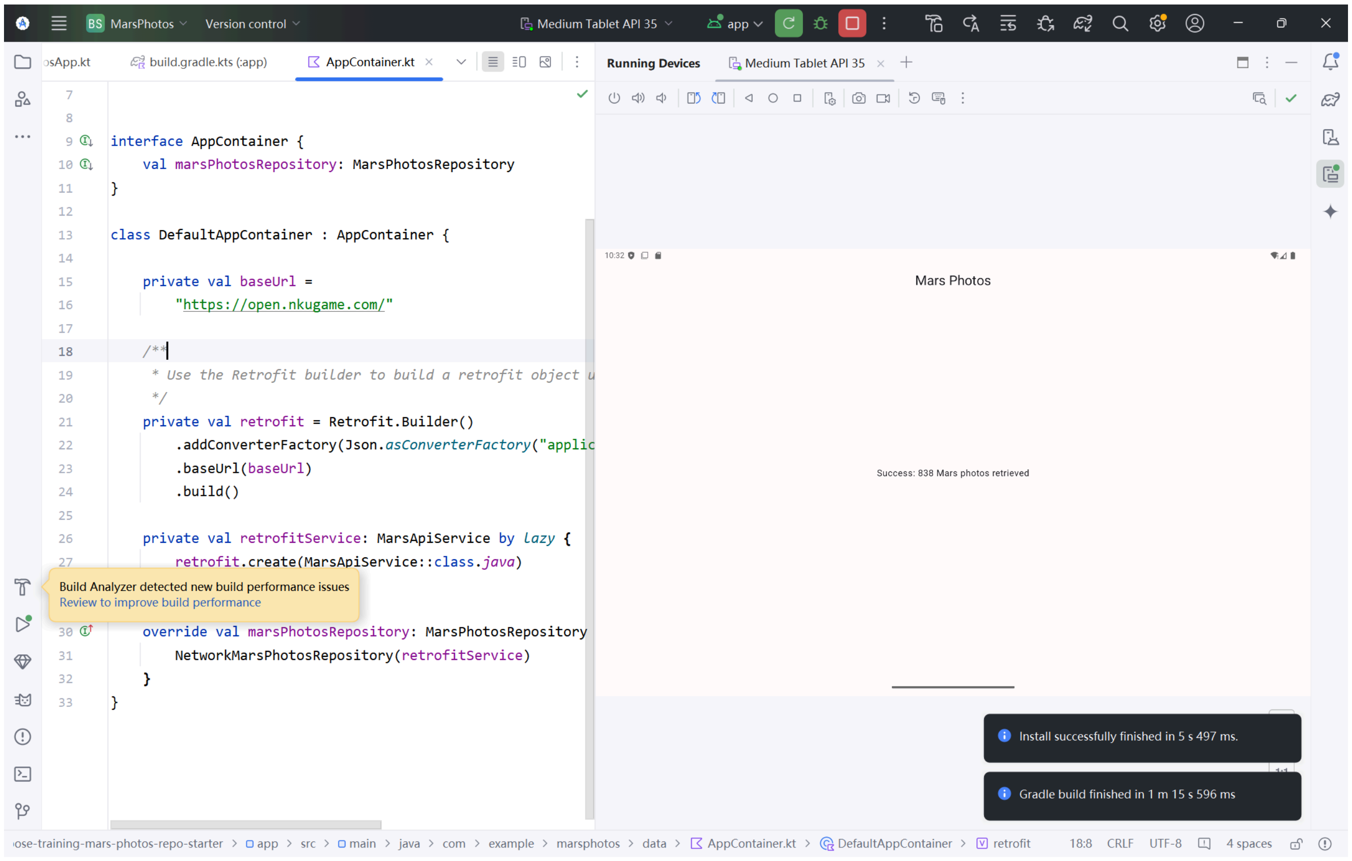
编写 ViewModel 测试



```kotlin
package com.example.marsphotos.fake

import com.example.marsphotos.data.MarsPhotosRepository
import com.example.marsphotos.model.MarsPhoto
import com.example.marsphotos.ui.screens.MarsUiState
import com.example.marsphotos.ui.screens.MarsViewModel
import kotlinx.coroutines.test.runTest
import org.junit.Assert.assertEquals
import org.junit.Test

class FakeNetworkMarsPhotosRepository : MarsPhotosRepository {
    override suspend fun getMarsPhotos(): List<MarsPhoto> {
        return FakeDataSource.photosList
    }
    @Test
    fun marsViewModel_getMarsPhotos_verifyMarsUiStateSuccess() =
        runTest {
            val marsViewModel = MarsViewModel(
                marsPhotosRepository = FakeNetworkMarsPhotosRepository()
            )
            assertEquals(
                MarsUiState.Success( photos: "Success: ${FakeDataSource.photosList.size} Mars " +
                        "photos retrieved"),
                marsViewModel.marsUiState
            )
        }
}
```

MarsPhotos
Version control
Medium Tablet API 35
app
keDataSource.kt
NetworkMarsPhotosRepositoryTest.kt
FakeNetworkMarsPhotosRepository.kt
TestDispatcherRule.kt
AppContainer.kt
AndroidManifest.xml
app > src > test > java > com > example > marsphotos > fake > FakeNetworkMarsPhotosRepository > marsViewModel_getMarsPhotos_verifyMarsUiStateSuccess
16:30 CRLF UTF-8 4 spaces

实现效果



```
interface AppContainer {
    val marsPhotosRepository: MarsPhotosRepository
}

class DefaultAppContainer : AppContainer {

    private val baseUrl =
        "https://open.nkugame.com/"

    /**
     * Use the Retrofit builder to build a retrofit object u
     */
    private val retrofit = Retrofit.Builder()
        .addConverterFactory(Json.asConverterFactory("applic
        .baseUrl(baseUrl)
        .build()

    private val retrofitService: MarsApiService by lazy {
        retrofit.create(MarsApiService::class.java)

    override val marsPhotosRepository: MarsPhotosRepository
        NetworkMarsPhotosRepository(retrofitService)
    }
}
```

Build Analyzer detected new build performance issues
Review to improve build performance

Mars Photos

Success: 838 Mars photos retrieved

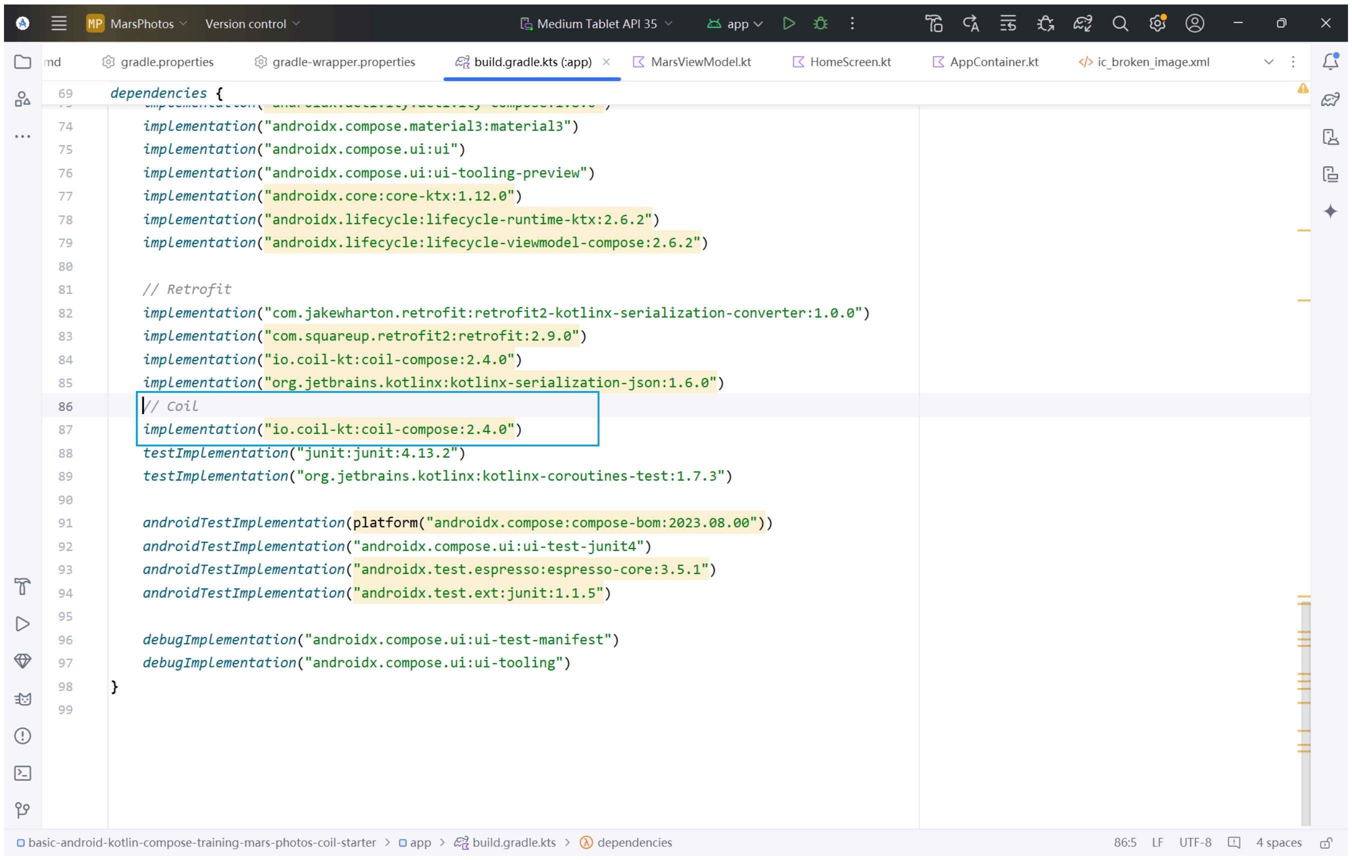
Install successfully finished in 5 s 497 ms.

Gradle build finished in 1 m 15 s 596 ms

MarsPhotos
Version control
Medium Tablet API 35
app
osApp.kt
build.gradle.kts (app)
AppContainer.kt
Running Devices
10:32
ose-training-mars-photos-repo-starter > app > src > main > java > com > example > marsphotos > data > AppContainer.kt > DefaultAppContainer > retrofit
18:8 CRLF UTF-8 4 spaces

从互联网加载和显示图片

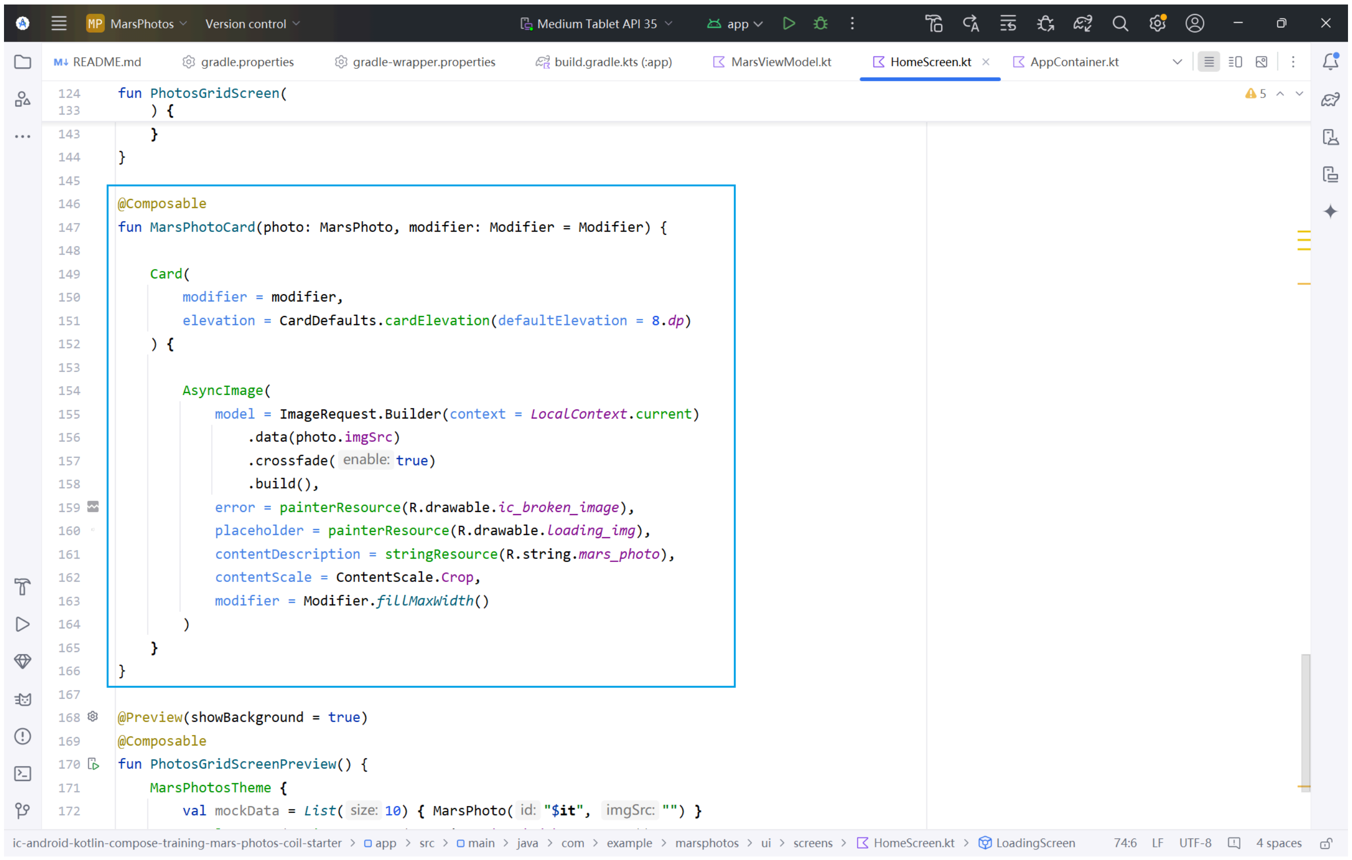
添加依赖项



```
dependencies {
    implementation("androidx.compose.material3:material3")
    implementation("androidx.compose.ui:ui")
    implementation("androidx.compose.ui:ui-tooling-preview")
    implementation("androidx.core:core-ktx:1.12.0")
    implementation("androidx.lifecycle:lifecycle-runtime-ktx:2.6.2")
    implementation("androidx.lifecycle:lifecycle-viewmodel-compose:2.6.2")

    // Retrofit
    implementation("com.jakewharton.retrofit:retrofit2-kotlinx-serialization-converter:1.0.0")
    implementation("com.squareup.retrofit2:retrofit:2.9.0")
    implementation("io.coil-kt:coil-compose:2.4.0")
    implementation("org.jetbrains.kotlinx:kotlinx-serialization-json:1.6.0")
    // Coil
    implementation("io.coil-kt:coil-compose:2.4.0")
    testImplementation("junit:junit:4.13.2")
    testImplementation("org.jetbrains.kotlinx:kotlinx-coroutines-test:1.7.3")

    androidTestImplementation(platform("androidx.compose:compose-bom:2023.08.00"))
    androidTestImplementation("androidx.compose.ui:ui-test-junit4")
    androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")
    androidTestImplementation("androidx.test.ext:junit:1.1.5")

    debugImplementation("androidx.compose.ui:ui-test-manifest")
    debugImplementation("androidx.compose.ui:ui-tooling")
}
```

显示现在图片

添加同步图像函数

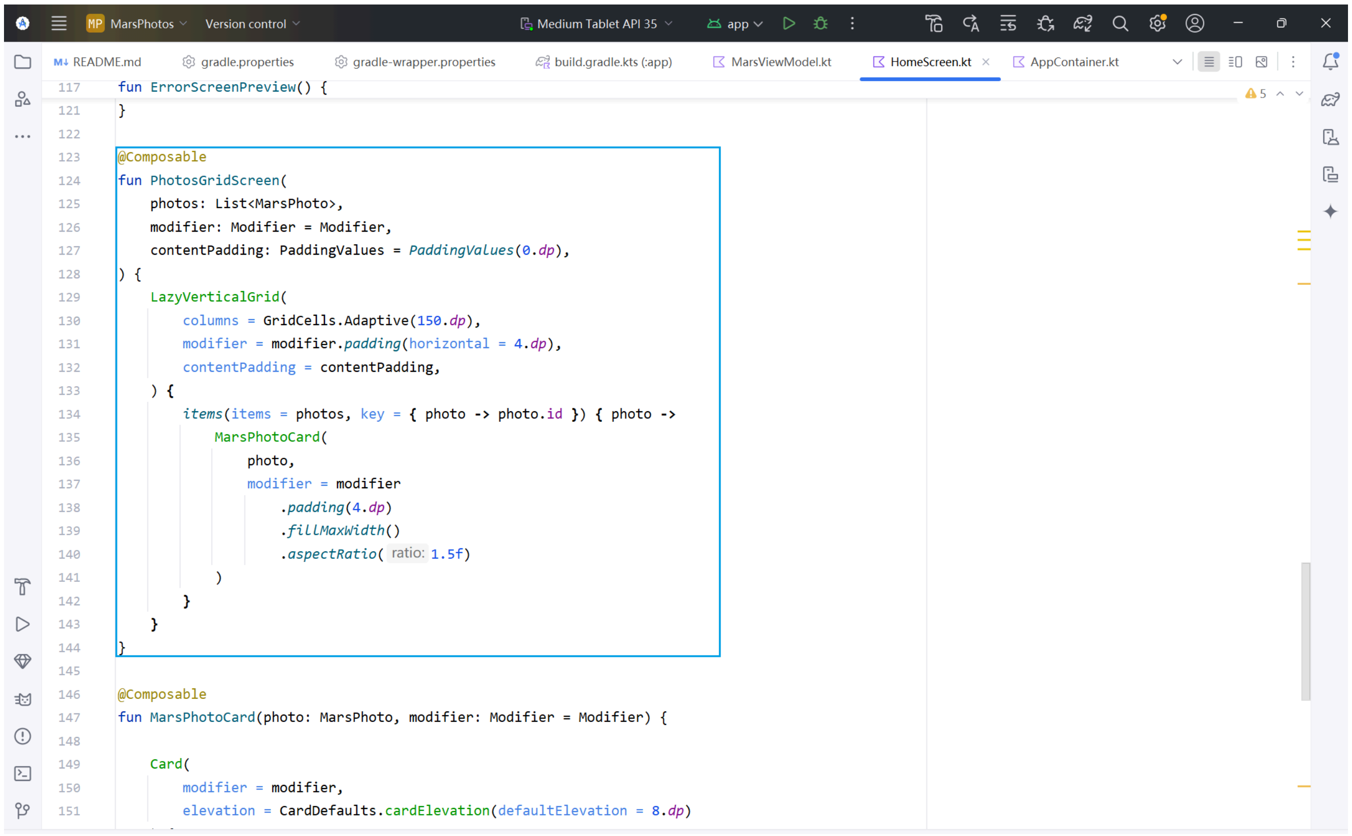

MarsPhotos   Version control   Medium Tablet API 35   app

README.md   gradle.properties   gradle-wrapper.properties   build.gradle.kts (app)   MarsViewModel.kt   HomeScreen.kt   AppContainer.kt

```kotlin
fun PhotosGridScreen(
    ) {
    }
}

@Composable
fun MarsPhotoCard(photo: MarsPhoto, modifier: Modifier = Modifier) {

    Card(
        modifier = modifier,
        elevation = CardDefaults.cardElevation(defaultElevation = 8.dp)
    ) {

        AsyncImage(
            model = ImageRequest.Builder(context = LocalContext.current)
                .data(photo.imgSrc)
                .crossfade( enable: true)
                .build(),
            error = painterResource(R.drawable.ic_broken_image),
            placeholder = painterResource(R.drawable.loading_img),
            contentDescription = stringResource(R.string.mars_photo),
            contentScale = ContentScale.Crop,
            modifier = Modifier.fillMaxWidth()
        )
    }
}

@Preview(showBackground = true)
@Composable
fun PhotosGridScreenPreview() {
    MarsPhotosTheme {
        val mockData = List( size: 10) { MarsPhoto( id: "$it",  imgSrc: "") }
```

ic-android-kotlin-compose-training-mars-photos-coil-starter > app > src > main > java > com > example > marsphotos > ui > screens > HomeScreen.kt > LoadingScreen   74:6   LF   UTF-8   4 spaces

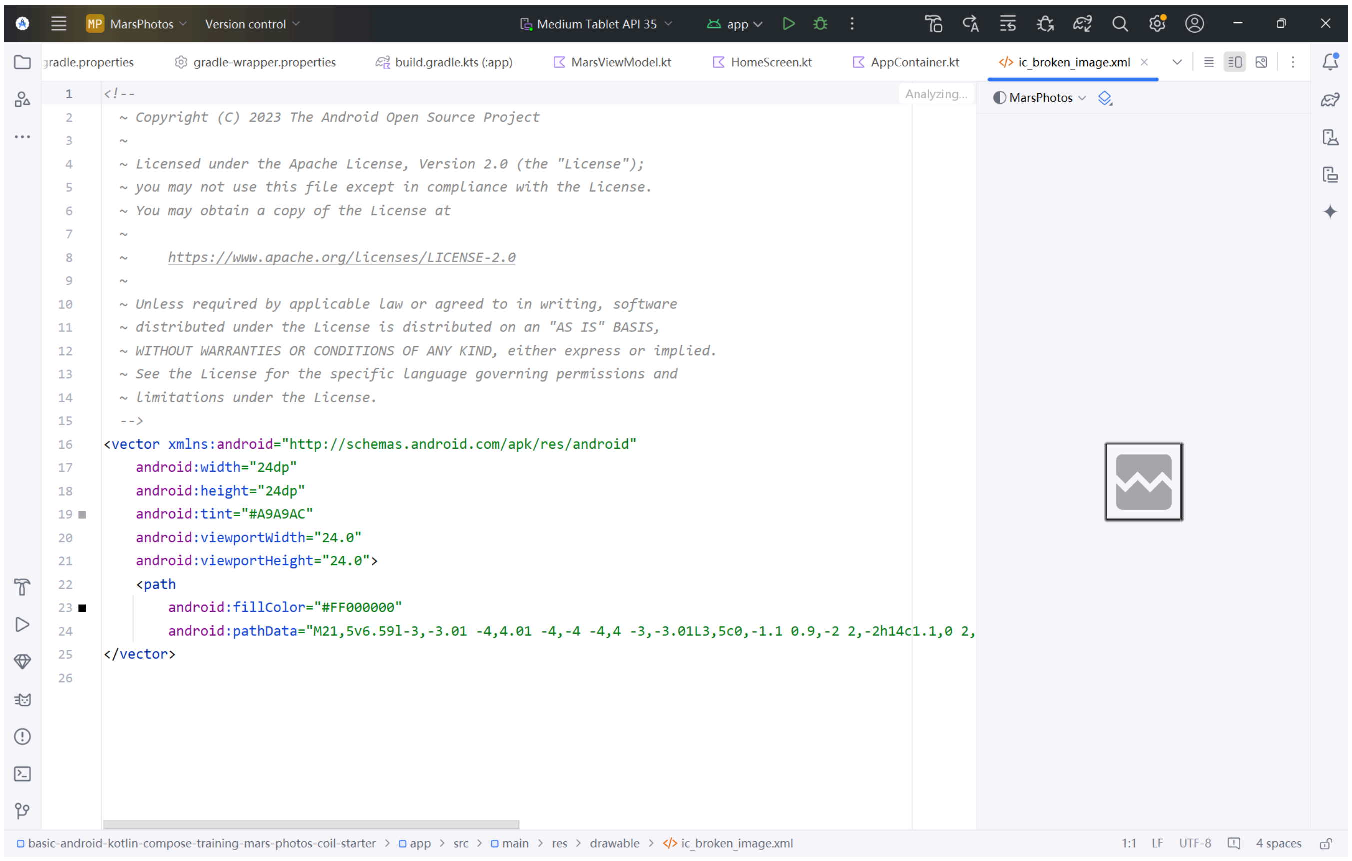
lazyVerticalGrid


MarsPhotos   Version control   Medium Tablet API 35   app

README.md   gradle.properties   gradle-wrapper.properties   build.gradle.kts (app)   MarsViewModel.kt   HomeScreen.kt   AppContainer.kt

```kotlin
fun ErrorScreenPreview() {
}

@Composable
fun PhotosGridScreen(
    photos: List<MarsPhoto>,
    modifier: Modifier = Modifier,
    contentPadding: PaddingValues = PaddingValues(0.dp),
) {
    LazyVerticalGrid(
        columns = GridCells.Adaptive(150.dp),
        modifier = modifier.padding(horizontal = 4.dp),
        contentPadding = contentPadding,
    ) {
        items(items = photos, key = { photo -> photo.id }) { photo ->
            MarsPhotoCard(
                photo,
                modifier = modifier
                    .padding(4.dp)
                    .fillMaxWidth()
                    .aspectRatio( ratio: 1.5f)
            )
        }
    }
}

@Composable
fun MarsPhotoCard(photo: MarsPhoto, modifier: Modifier = Modifier) {

    Card(
        modifier = modifier,
        elevation = CardDefaults.cardElevation(defaultElevation = 8.dp)
```

更新图片配置

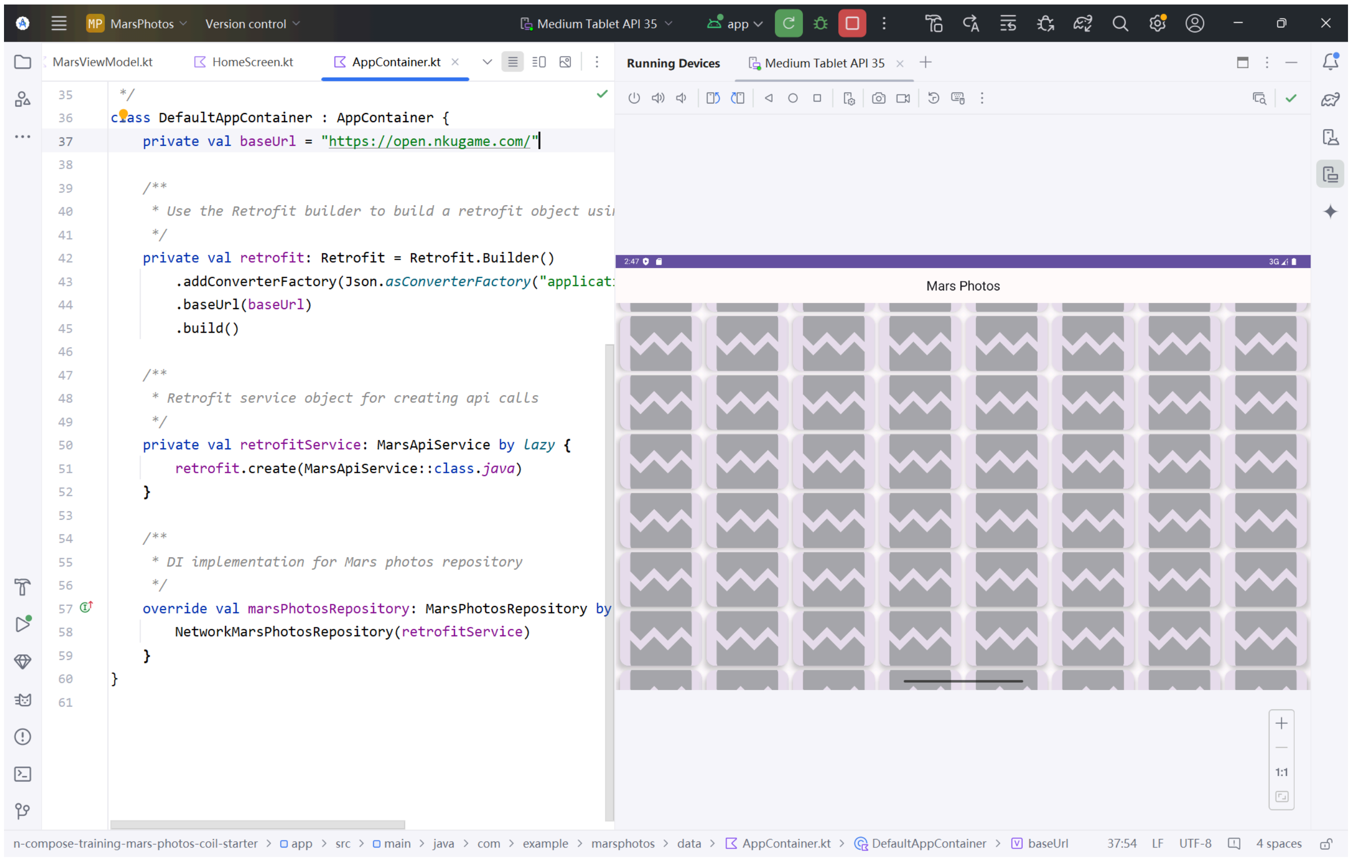

MarsPhotos　Version control　Medium Tablet API 35　app

gradle.properties　gradle-wrapper.properties　build.gradle.kts (:app)　MarsViewModel.kt　HomeScreen.kt　AppContainer.kt　ic_broken_image.xml

```xml
<!--
  ~ Copyright (C) 2023 The Android Open Source Project
  ~
  ~ Licensed under the Apache License, Version 2.0 (the "License");
  ~ you may not use this file except in compliance with the License.
  ~ You may obtain a copy of the License at
  ~
  ~     https://www.apache.org/licenses/LICENSE-2.0
  ~
  ~ Unless required by applicable law or agreed to in writing, software
  ~ distributed under the License is distributed on an "AS IS" BASIS,
  ~ WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  ~ See the License for the specific language governing permissions and
  ~ limitations under the License.
  -->
<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="24dp"
    android:height="24dp"
    android:tint="#A9A9AC"
    android:viewportWidth="24.0"
    android:viewportHeight="24.0">
    <path
        android:fillColor="#FF000000"
        android:pathData="M21,5v6.59l-3,-3.01 -4,4.01 -4,-4 -4,4 -3,-3.01L3,5c0,-1.1 0.9,-2 2,-2h14c1.1,0 2,
</vector>
```

basic-android-kotlin-compose-training-mars-photos-coil-starter > app > src > main > res > drawable > ic_broken_image.xml　1:1　LF　UTF-8　4 spaces

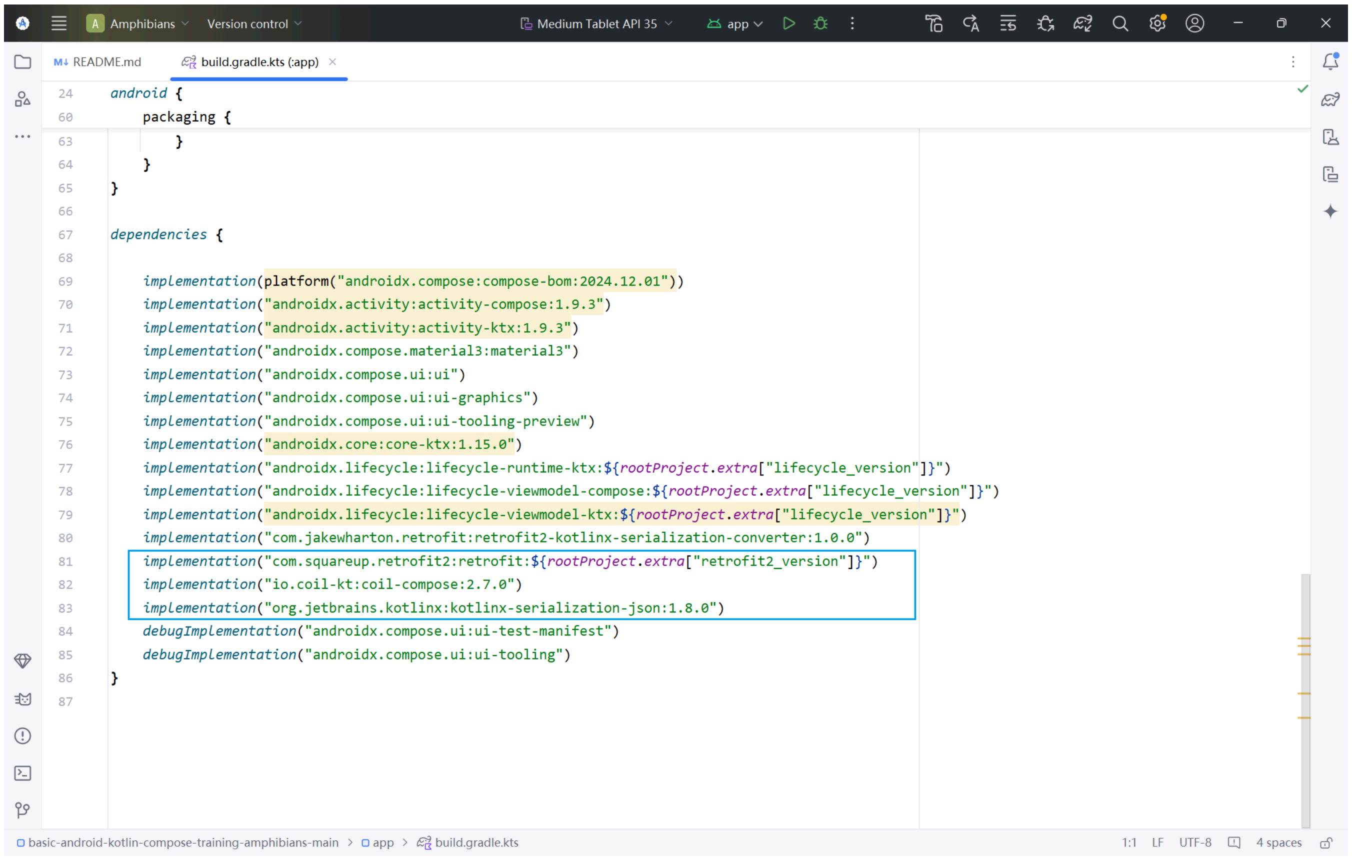
实现效果



```
    */
class DefaultAppContainer : AppContainer {
    private val baseUrl = "https://open.nkugame.com/"

    /**
     * Use the Retrofit builder to build a retrofit object usi
     */
    private val retrofit: Retrofit = Retrofit.Builder()
        .addConverterFactory(Json.asConverterFactory("applicat
        .baseUrl(baseUrl)
        .build()

    /**
     * Retrofit service object for creating api calls
     */
    private val retrofitService: MarsApiService by lazy {
        retrofit.create(MarsApiService::class.java)
    }

    /**
     * DI implementation for Mars photos repository
     */
    override val marsPhotosRepository: MarsPhotosRepository by
        NetworkMarsPhotosRepository(retrofitService)
    }
}
```

练习：构建两栖动物应用

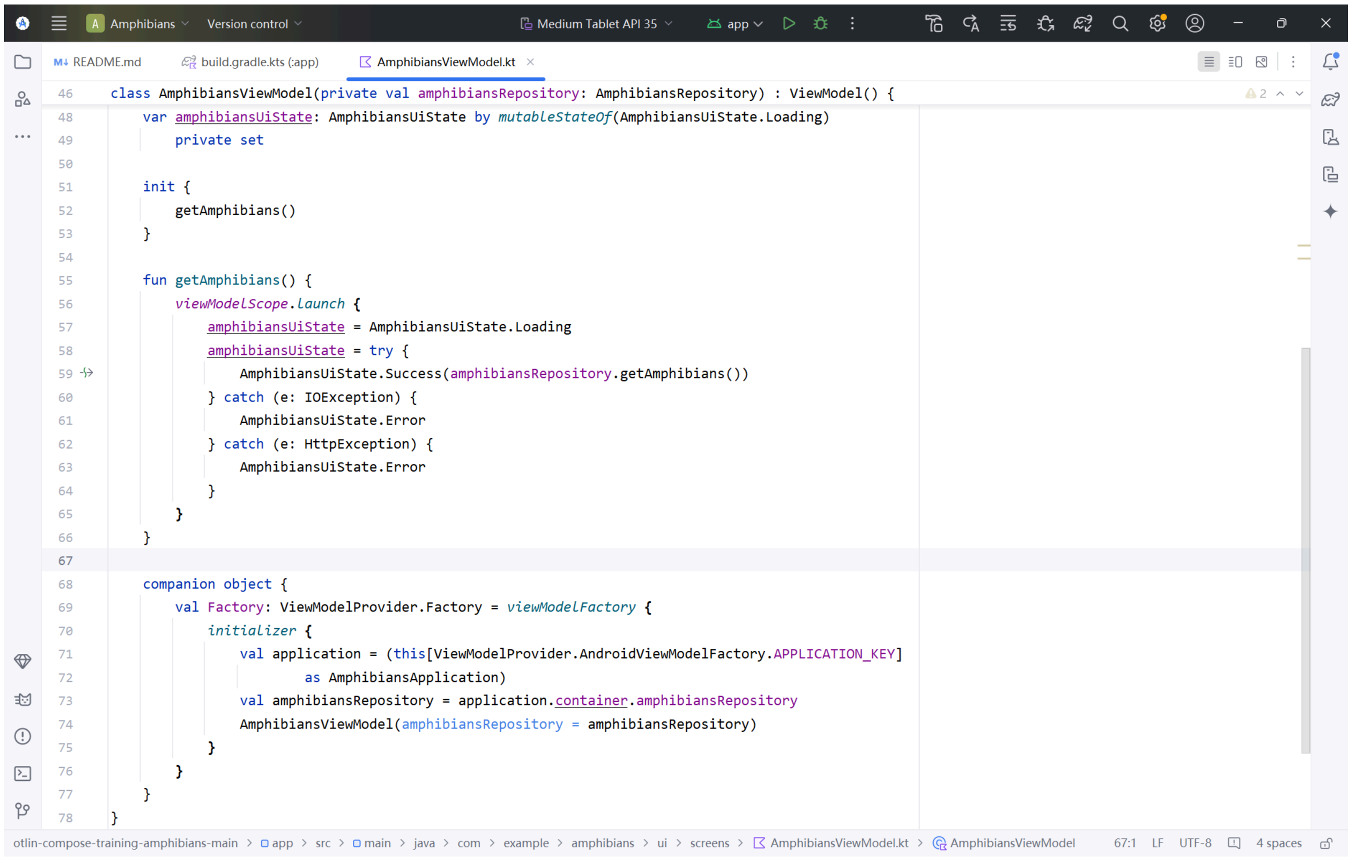
设置依赖项



```
android {
    packaging {
        }
    }
}

dependencies {

    implementation(platform("androidx.compose:compose-bom:2024.12.01"))
    implementation("androidx.activity:activity-compose:1.9.3")
    implementation("androidx.activity:activity-ktx:1.9.3")
    implementation("androidx.compose.material3:material3")
    implementation("androidx.compose.ui:ui")
    implementation("androidx.compose.ui:ui-graphics")
    implementation("androidx.compose.ui:ui-tooling-preview")
    implementation("androidx.core:core-ktx:1.15.0")
    implementation("androidx.lifecycle:lifecycle-runtime-ktx:${rootProject.extra["lifecycle_version"]}")
    implementation("androidx.lifecycle:lifecycle-viewmodel-compose:${rootProject.extra["lifecycle_version"]}")
    implementation("androidx.lifecycle:lifecycle-viewmodel-ktx:${rootProject.extra["lifecycle_version"]}")
    implementation("com.jakewharton.retrofit:retrofit2-kotlinx-serialization-converter:1.0.0")
    implementation("com.squareup.retrofit2:retrofit:${rootProject.extra["retrofit2_version"]}")
    implementation("io.coil-kt:coil-compose:2.7.0")
    implementation("org.jetbrains.kotlinx:kotlinx-serialization-json:1.8.0")
    debugImplementation("androidx.compose.ui:ui-test-manifest")
    debugImplementation("androidx.compose.ui:ui-tooling")
}
```

创建界面层

公开屏幕界面状态，处理界面层中的业务逻辑

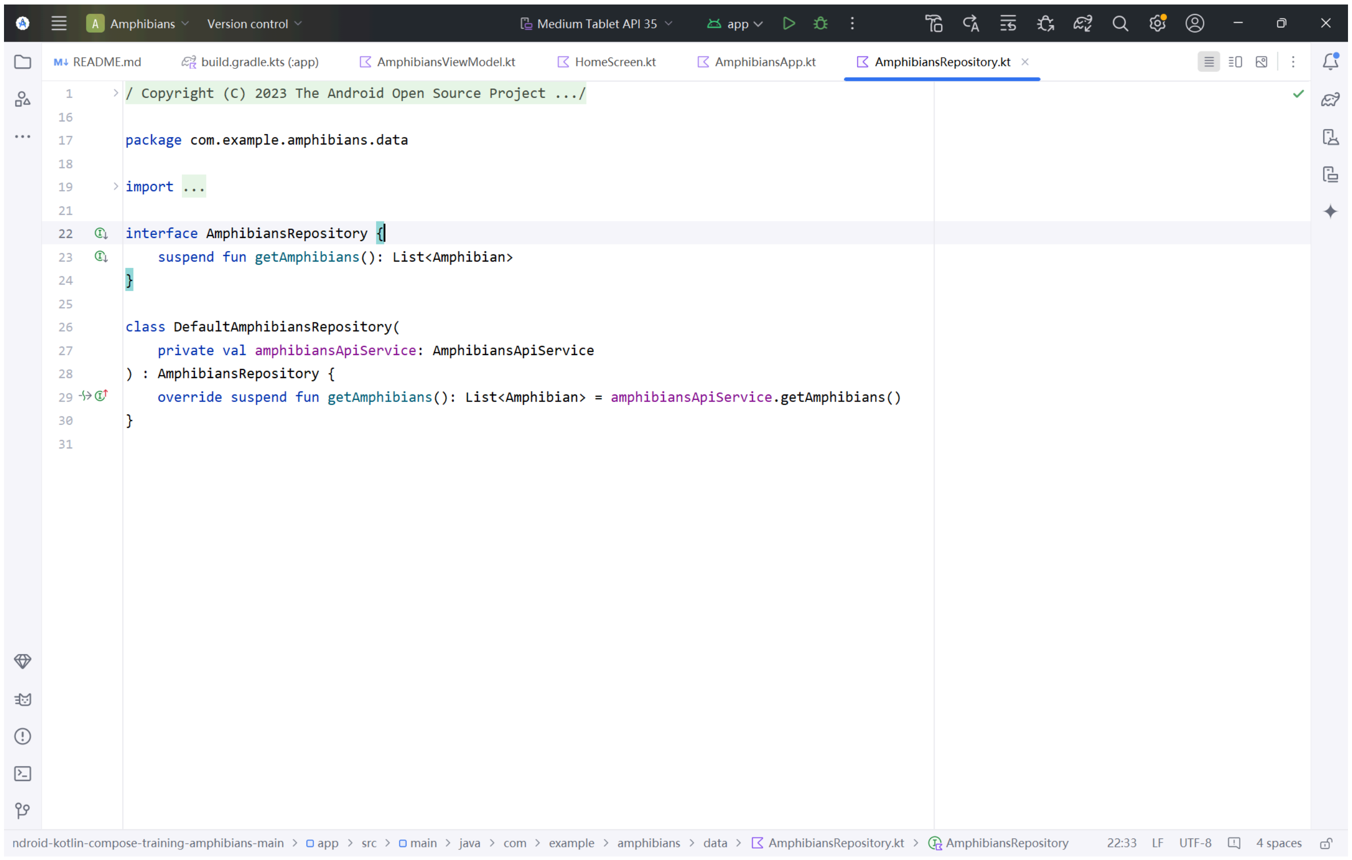


```kotlin
class AmphibiansViewModel(private val amphibiansRepository: AmphibiansRepository) : ViewModel() {
    var amphibiansUiState: AmphibiansUiState by mutableStateOf(AmphibiansUiState.Loading)
        private set

    init {
        getAmphibians()
    }

    fun getAmphibians() {
        viewModelScope.launch {
            amphibiansUiState = AmphibiansUiState.Loading
            amphibiansUiState = try {
                AmphibiansUiState.Success(amphibiansRepository.getAmphibians())
            } catch (e: IOException) {
                AmphibiansUiState.Error
            } catch (e: HttpException) {
                AmphibiansUiState.Error
            }
        }
    }

    companion object {
        val Factory: ViewModelProvider.Factory = viewModelFactory {
            initializer {
                val application = (this[ViewModelProvider.AndroidViewModelFactory.APPLICATION_KEY]
                        as AmphibiansApplication)
                val amphibiansRepository = application.container.amphibiansRepository
                AmphibiansViewModel(amphibiansRepository = amphibiansRepository)
            }
        }
    }
}
```

otlin-compose-training-amphibians-main > app > src > main > java > com > example > amphibians > ui > screens > AmphibiansViewModel.kt > AmphibiansViewModel
67:1 LF UTF-8 4 spaces

创建数据层

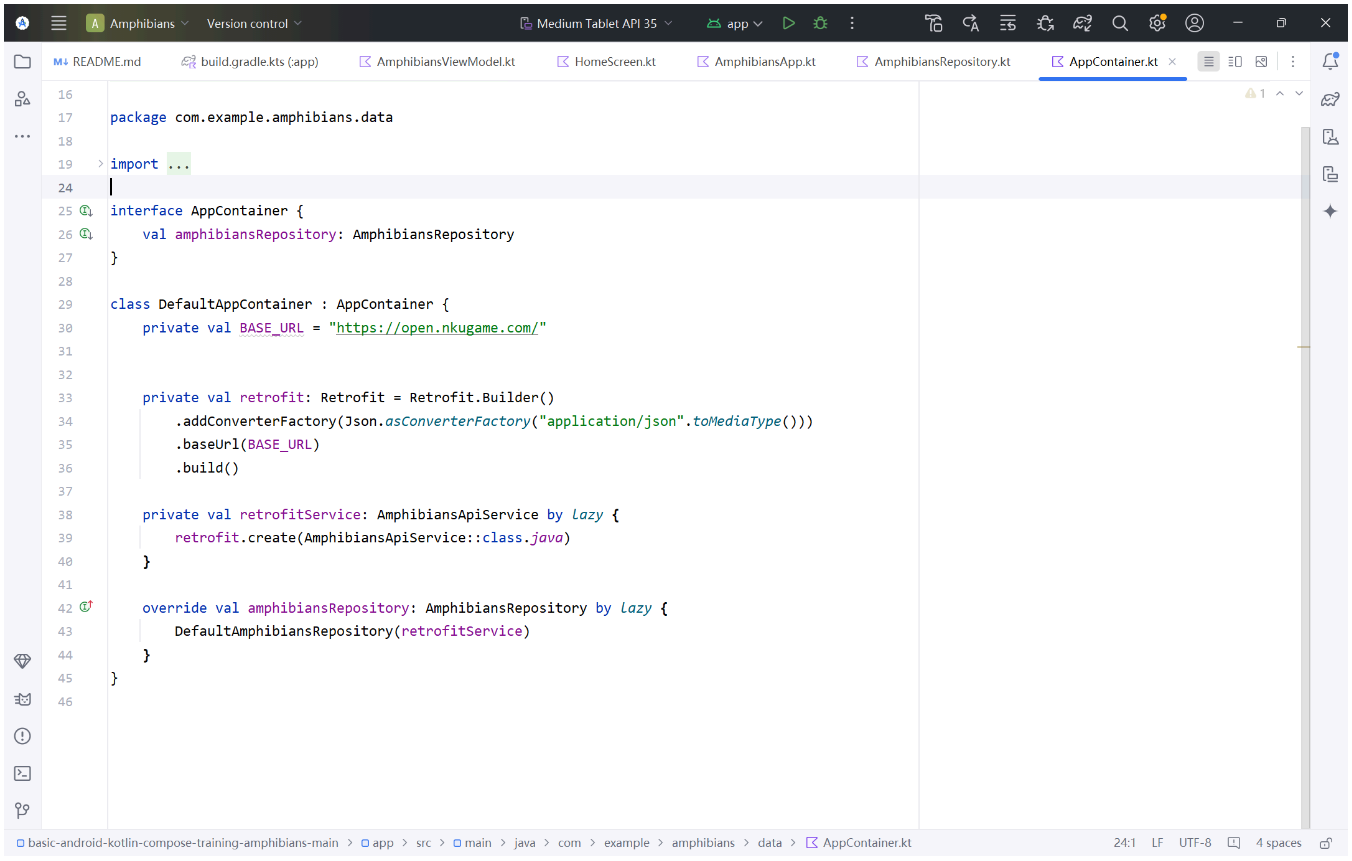
从互联网获取数据



README.md | build.gradle.kts (:app) | AmphibiansViewModel.kt | HomeScreen.kt | AmphibiansApp.kt | AmphibiansRepository.kt

```kotlin
/ Copyright (C) 2023 The Android Open Source Project .../

package com.example.amphibians.data

import ...

interface AmphibiansRepository {
    suspend fun getAmphibians(): List<Amphibian>
}

class DefaultAmphibiansRepository(
    private val amphibiansApiService: AmphibiansApiService
) : AmphibiansRepository {
    override suspend fun getAmphibians(): List<Amphibian> = amphibiansApiService.getAmphibians()
}
```

ndroid-kotlin-compose-training-amphibians-main > app > src > main > java > com > example > amphibians > data > AmphibiansRepository.kt > AmphibiansRepository   22:33 LF UTF-8 4 spaces

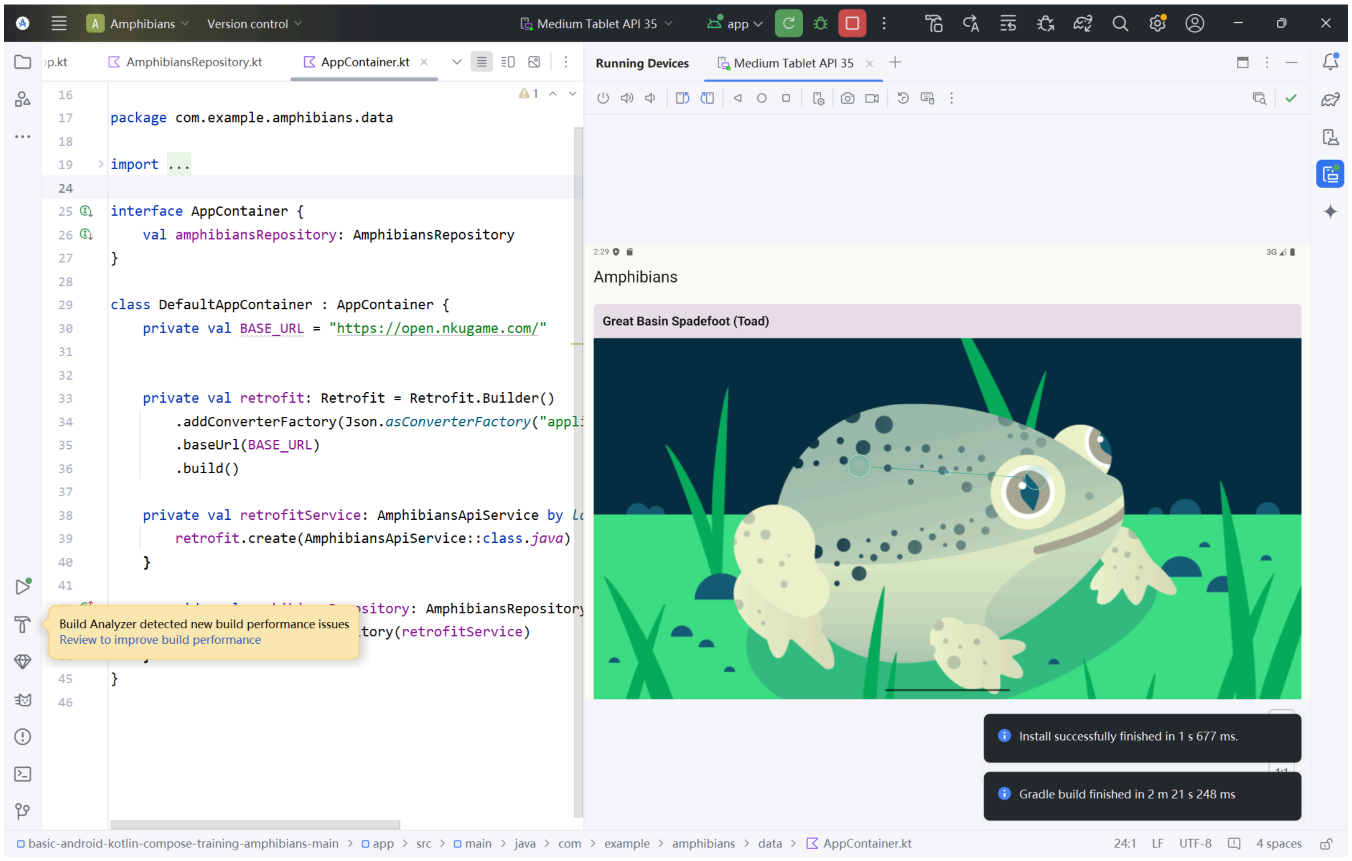


README.md | build.gradle.kts (:app) | AmphibiansViewModel.kt | HomeScreen.kt | AmphibiansApp.kt | AmphibiansRepository.kt | AppContainer.kt

```kotlin
package com.example.amphibians.data

import ...

interface AppContainer {
    val amphibiansRepository: AmphibiansRepository
}

class DefaultAppContainer : AppContainer {
    private val BASE_URL = "https://open.nkugame.com/"


    private val retrofit: Retrofit = Retrofit.Builder()
        .addConverterFactory(Json.asConverterFactory("application/json".toMediaType()))
        .baseUrl(BASE_URL)
        .build()

    private val retrofitService: AmphibiansApiService by lazy {
        retrofit.create(AmphibiansApiService::class.java)
    }

    override val amphibiansRepository: AmphibiansRepository by lazy {
        DefaultAmphibiansRepository(retrofitService)
    }
}
```

basic-android-kotlin-compose-training-amphibians-main > app > src > main > java > com > example > amphibians > data > AppContainer.kt   24:1 LF UTF-8 4 spaces

实现效果



```kotlin
package com.example.amphibians.data

import ...

interface AppContainer {
    val amphibiansRepository: AmphibiansRepository
}

class DefaultAppContainer : AppContainer {
    private val BASE_URL = "https://open.nkugame.com/"


    private val retrofit: Retrofit = Retrofit.Builder()
        .addConverterFactory(Json.asConverterFactory("appli
        .baseUrl(BASE_URL)
        .build()

    private val retrofitService: AmphibiansApiService by l
        retrofit.create(AmphibiansApiService::class.java)
    }

                          sitory: AmphibiansRepository
                         tory(retrofitService)

}
```

Build Analyzer detected new build performance issues
Review to improve build performance

Amphibians

Great Basin Spadefoot (Toad)

Install successfully finished in 1 s 677 ms.

Gradle build finished in 2 m 21 s 248 ms

basic-android-kotlin-compose-training-amphibians-main > app > src > main > java > com > example > amphibians > data > AppContainer.kt   24:1 LF UTF-8 4 spaces