

Assignment 3

Rayana Sri Sai Santhosh – 20BIT0185

Video Player

The Code:

1. Video API

```
package santhosh_20bit0185.data.remote.api

import retrofit2.http.GET
import retrofit2.http.Query
import xyz.teamgravity.onlinevideoplayer.data.remote.dto.VideoDto

interface VideoApi {

    companion object {
        const val BASE_URL = "https://picsum.photos"
        const val PAGE = 1
        const val PER_PAGE = 10
        const val MAX_SIZE = 60
        const val VIDEO_URL =
            "https://firebasestorage.googleapis.com/v0/b/spotify-clone-702f3.appspot.com/o/video%2Falone.mp4?alt=media&token=73be6ebd-3f5f-4d2b-a3e5-1c9458e4734f"
    }

    @GET("/v2/list")
    suspend fun getPopularVideos(
        @Query("page") page: Int,
        @Query("limit") perPage: Int,
    ): List<VideoDto>
}
```

2. Video Paging Source

```
package santhosh_20bit0185.data.remote.datasource

import androidx.paging.PagingSource
import androidx.paging.PagingState
import timber.log.Timber
import santhosh_20bit0185.data.remote.api.VideoApi
import santhosh_20bit0185.data.remote.dto.VideoDto

class VideoPagingSource(
    private val api: VideoApi
)
```

```

) : PagingSource<Int, VideoDto>() {

    override fun getRefreshKey(state: PagingState<Int,
VideoDto>): Int? {
        return state.anchorPosition
    }

    override suspend fun load(params: LoadParams<Int>):
LoadResult<Int, VideoDto> {
        val page = params.key ?: VideoApi.PAGE

        return try {
            val videos = api.getPopularVideos(
                page = page,
                perPage = params.loadSize
            )

            Timber.d("Just api request")

            LoadResult.Page(
                data = videos,
                prevKey = if (page == VideoApi.PAGE) null else
page - 1,
                nextKey = if (videos.isEmpty()) null else page +
1
            )
        } catch (e: Exception) {
            Timber.d("Exception: ${e.message}")
            LoadResult.Error(e)
        }
    }
}

```

3. Get Videos

```

package santhosh_20bit0185.domain.usecase

import androidx.paging.PagingData
import kotlinx.coroutines.flow.Flow
import santhosh_20bit0185.domain.model.VideoModel
import santhosh_20bit0185.domain.repository.VideoRepository

class GetVideos(
    private val repository: VideoRepository
) {

    operator fun invoke(): Flow<PagingData<VideoModel>> {
        return repository.getVideos()
    }
}

```

4. Main Page UI

```
package santhosh_20bit0185.presentation.activity

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Surface
import androidx.compose.ui.Modifier
import dagger.hilt.android.AndroidEntryPoint
import santhosh_20bit0185.presentation.navigation.Navigation
import
santhosh_20bit0185.presentation.theme.OnlineVideoPlayerTheme

@AndroidEntryPoint
class Main : ComponentActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            OnlineVideoPlayerTheme {
                Surface(
                    color = MaterialTheme.colorScheme.background,
                    modifier = Modifier.fillMaxSize()
                ) {
                    Navigation()
                }
            }
        }
    }
}
```

5. Video List View

```
package santhosh_20bit0185.presentation.viewmodel

import androidx.compose.runtime.getValue
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.setValue
import androidx.lifecycle.SavedStateHandle
import androidx.lifecycle.ViewModel
import androidx.lifecycle.viewModelScope
import androidx.paging.PagingData
import androidx.paging.cachedIn
import dagger.hilt.android.lifecycle.HiltViewModel
import kotlinx.coroutines.flow.Flow
import santhosh_20bit0185.domain.model.VideoModel
import santhosh_20bit0185.domain.usecase.GetVideos
import javax.inject.Inject

@HiltViewModel
class VideoListViewModel @Inject constructor(
```

```

        getVideos: GetVideos,
        private val handle: SavedStateHandle,
    ) : ViewModel() {

        companion object {
            private const val HEART_CHECKED = "heartChecked"
            private const val HEART_CHECKED_DEFAULT = false
        }

        val videos: Flow<PagingData<VideoModel>> =
            getVideos().cachedIn(viewModelScope)

        var heartChecked: Boolean by
            mutableStateOf(handle.get<Boolean>(HEART_CHECKED) ?:
                HEART_CHECKED_DEFAULT)
            private set

        fun onHeartCheckedChange() {
            handle[HEART_CHECKED] = !heartChecked
            heartChecked = !heartChecked
        }
    }
}

```

6. Main

```

package santhosh_20bit0185.presentation.activity;

import android.content.Context;
import androidx.activity.ComponentActivity;
import androidx.activity.contextaware.OnContextAvailableListener;
import androidx.lifecycle.ViewModelProvider;
import dagger.hilt.android.internal.lifecycle.DefaultViewModelFactories;
import dagger.hilt.android.internal.managers.ActivityComponentManager;
import dagger.hilt.internal.GeneratedComponentManagerHolder;
import dagger.hilt.internal.UnsafeCast;
import java.lang.Object;
import java.lang.Override;

/**
 * A generated base class to be extended by the
 * @dagger.hilt.android.AndroidEntryPoint annotated class. If using
 * the Gradle plugin, this is swapped as the base class via bytecode
 * transformation.
 */
public abstract class Hilt_Main extends ComponentActivity
    implements GeneratedComponentManagerHolder {
    private volatile ActivityComponentManager componentManager;

    private final Object componentManagerLock = new Object();

    private boolean injected = false;

```

```

Hilt_Main() {
    super();
    _initHiltInternal();
}

Hilt_Main(int contentLayoutId) {
    super(contentLayoutId);
    _initHiltInternal();
}

private void _initHiltInternal() {
    addOnContextAvailableListener(new
OnContextAvailableListener() {
        @Override
        public void onContextAvailable(Context context) {
            inject();
        }
    });
}

@Override
public final Object generatedComponent() {
    return this.componentManager().generatedComponent();
}

protected ActivityComponentManager createComponentManager() {
    return new ActivityComponentManager(this);
}

@Override
public final ActivityComponentManager componentManager() {
    if (componentManager == null) {
        synchronized (componentManagerLock) {
            if (componentManager == null) {
                componentManager = createComponentManager();
            }
        }
    }
    return componentManager;
}

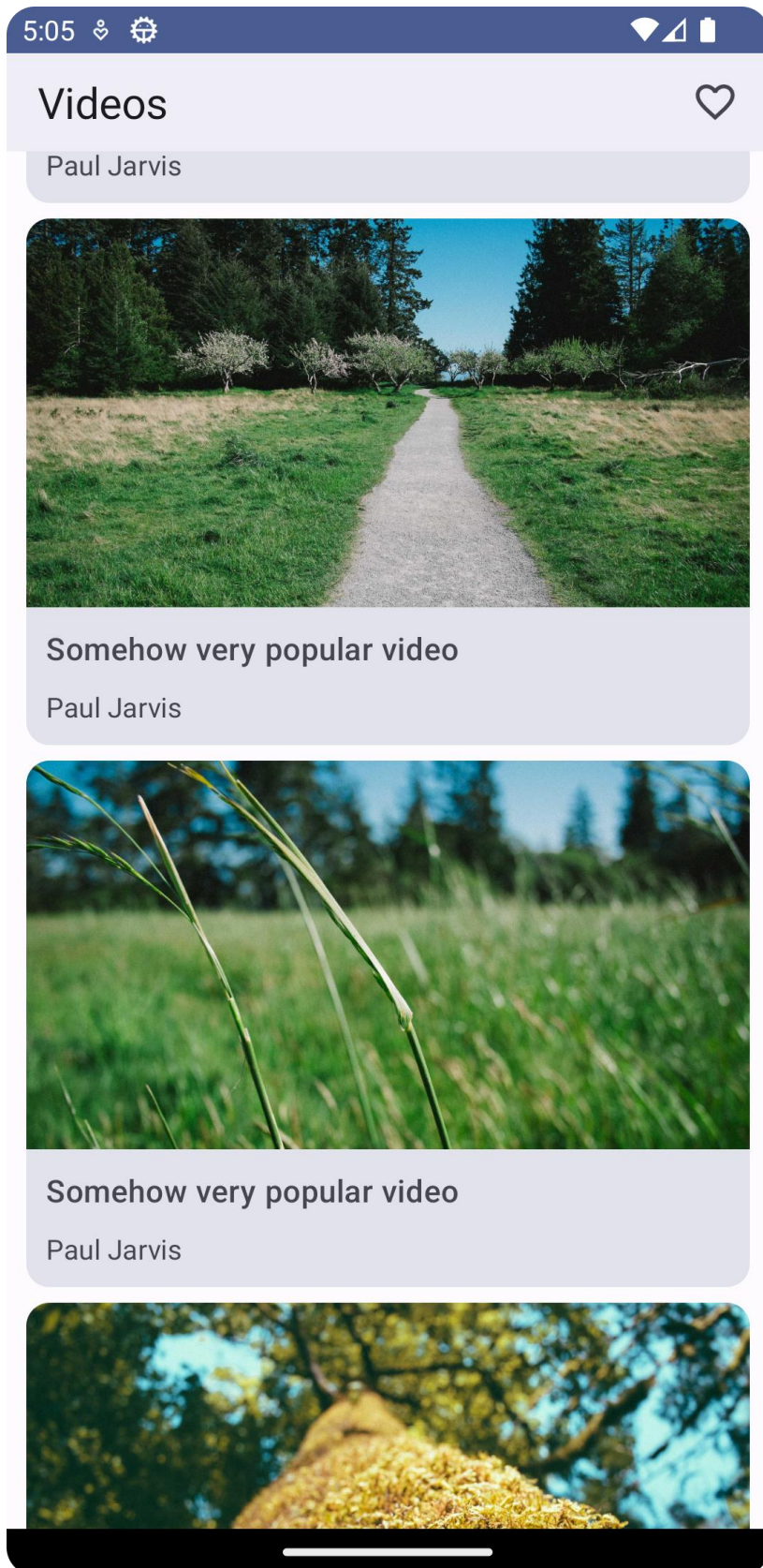
protected void inject() {
    if (!injected) {
        injected = true;
        ((Main_GeneratedInjector)
this.generatedComponent()).injectMain(UnsafeCasts.<Main>unsafeCast(this));
    }
}

@Override
public ViewModelProvider.Factory
getDefaultViewModelProviderFactory() {
    return DefaultViewModelFactories.getActivityFactory(this,
super.getDefaultViewModelProviderFactory());
}

```

```
}  
}
```

The Screenshots:



Videos



Somehow very popular video

Alejandro Escamilla



Somehow very popular video

Alejandro Escamilla





Somehow very popular video

Christopher Sardegna



Somehow very popular video

Luke Chesser

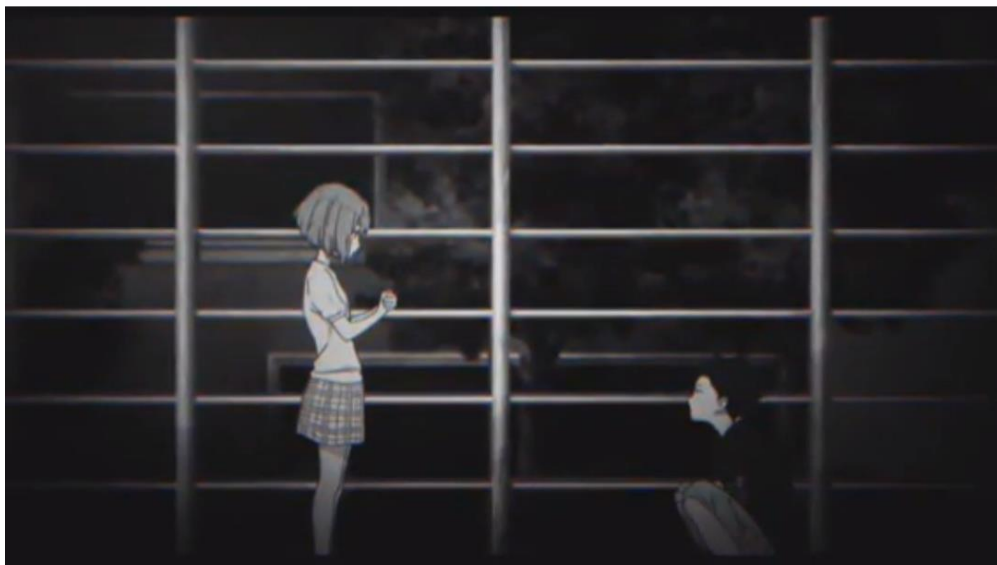


Somehow very popular video

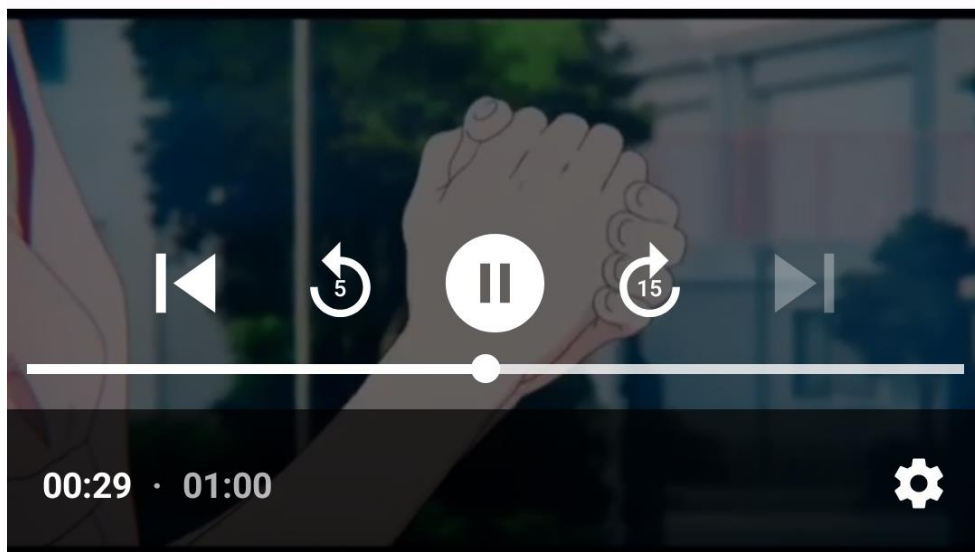
Margaret Barley



← Watch video



← Watch video



← Watch video

