

Reel Viewer App

1. Fragment Lifecycle Management

In the `ReelFragment` class, the lifecycle of the `ExoPlayer` instance is managed to optimize resource usage and to ensure smooth playback of video reels. Here's how we handled each key lifecycle event:

- `onCreateView`: Here, we inflate the layout and set up the `PlayerView`. The `initializePlayer()` method is called within `onViewCreated`, which prepares and starts video playback if the network is available.
- `onStart`: Playback is resumed by setting `player.playWhenReady` to `true`. This ensures the video starts automatically when the fragment becomes visible.
- `onResume`: We again set `player.playWhenReady = true` to handle cases where the app may have been paused and is resuming, ensuring that playback continues smoothly.
- `onPause`: Playback is paused by setting `player.playWhenReady = false` when the fragment is no longer in the foreground, preventing video playback while the user might not be actively watching the screen.
- `onStop`: This method calls `player.playWhenReady = false` to stop playback when the fragment is no longer visible. This ensures that no resources are used for playback when the user is not viewing the video.
- `onDestroyView`: The `PlayerView` is detached from the `ExoPlayer` to prevent memory leaks and ensure a clean UI state.
- `onDestroy`: The `ExoPlayer` instance is released here to free up resources once the fragment is destroyed, as the player is no longer needed.

By implementing playback control across these lifecycle methods, the app efficiently manages resources while providing a seamless video viewing experience.

2. Error Handling and Network Check

To handle potential network connectivity issues, a check is performed in `initializePlayer()` using `isNetworkAvailable()`:

- If no network is detected, a Snackbar message notifies the user about connectivity issues.

- Additionally, we listen for `onPlayerError` in `ExoPlayer` to provide user feedback if a playback error occurs, displaying the error via a Snackbar.

3. Title Display for Each Video

The `ReelFragment` is designed to accept a video title as an argument alongside the URL. This title is displayed in the UI to provide context for each video reel. Titles are passed via `ReelPagerAdapter`, which maintains a list of titles corresponding to each video URL.

4. Additional Features

- **Automatic Orientation Adjustment:** The `PlayerView` has been configured with `resizeMode` to adjust video scaling based on screen orientation. This ensures the video scales appropriately for both portrait and landscape modes.
- **Network-Aware Playback:** The app checks for network availability before attempting playback, improving the user experience by preventing playback interruptions due to network loss.

Repository Link-

<https://github.com/Harsh03004/ReelViewerApp>