

In this player we'll be working with exoplayer

We need to import the library need it, this we found it in the gitHub

We implemented it with just tipping SEPV in the xml and Android Studio will create de rest, and letter just create the instance in the activity

ExoPlayer is amazing because just with add the instance and media source we are ready to go, of course we need to set the media but nothing else

We learn to initializePlayer here one of the basic things to remember is to set the current window in 0 that way it going to start from the beginning the same with playback position, and for this example with are using a simple exo player so we need to create an instance of SimpleExoPlayer

We need to create the media because we need the ExoPlayer but we need to run something inside and this can happen just using the URI for set the location the instance MediaSource for create it wit the URI and letter just preparer to the player

As another's component than we need to register and unregister, we need to initializer the player, we will set this in onStart and in onResume depend of the API the same for releasePlayer in onPause and onStop

This codeLab show us how to set system Ui visibility something new for us

In the releasePlayer we learn to apply the set of the property for know where the media stop before going to foreground, that way when we resume the media will be in the same track

So how exoPlayer is so amazing, you just need to change your URL for a video and this will be played just with that, and if we want to play both we can use inside buildMediaSource the method ConcatenatingMediaSource and here we send the Audio and Video source, and one more time exoPlayer will take care

We Exoplayer we learned than we can play videos like netflix changing the resolution if the internet is very good or very bad and for that we use BANDWIDTH_METER and in the initializePlayer we add another property with the name AdaptiveTrackSelection.Factory but the big thing here is the resource need to be a dash url, because this is who is going to have the different quality of the source

In this codelab we learn about playbackState this one has four state and with this we can know if we already created an instance of the player but without media Source (ExoPlayer.STATE_IDLE), another state for the moment when he doesn't have enough for play (ExoPlayer.STATE_BUFFERING), the next one we used a lot and has a boolean value for let it know to the system if is playing or in pause (ExoPlayer.STATE_READY), and the last one is when it finished playing the source (ExoPlayer.STATE_ENDED)

ExoPlayer use `PlayerControlView` and this one is very customisable, one of the easy way is creating a new layout and paste the xml of the `playerView` from the repo and just change what we want because he look his views by the id, and letter in the main layout just add `controller_layout_id` with the id of the new layout