//First I added collapsing toolbar layout for the app.

//then I used the shimmering for the text <https://github.com/RomainPiel/Shimmer-android>

***Doubt in***

***GETTING STARTED : Authentication***

***ACCOUNT – after you create login***

***AUTHENTICATION***

***CHANGES IN ID OF A MOVIE/TV CHANNEL,ETC***

***COLLECTION – Getiing a collection detail, collection of a image for a particular collection id***

***COMPANIES – Getting details of various companies***

***CONFIGURATION***

***CREDITS***

***TMDB API***

***TMDP API PROVIDES THE VALID ID OF VAROIUS MOVIES AND ALL ON A CERTAIN DATE***

| Path | Name |
| --- | --- |
| Movies | /p/exports | movie\_ids\_MM\_DD\_YYYY.json.gz |
| TV Series | /p/exports | tv\_series\_ids\_MM\_DD\_YYYY.json.gz |
| People | /p/exports | person\_ids\_MM\_DD\_YYYY.json.gz |
| Collections | /p/exports | collection\_ids\_MM\_DD\_YYYY.json.gz |
| TV Networks | /p/exports | tv\_network\_ids\_MM\_DD\_YYYY.json.gz |
| Keywords | /p/exports | keyword\_ids\_MM\_DD\_YYYY.json.gz |
| Production Companies | /p/exports | production\_company\_ids\_MM\_DD\_YYYY.json.gz |

If you were looking for a list of valid movie ids, the full download URL for the file published on April 28 2017 would be --

<http://files.tmdb.org/p/exports/movie_ids_04_28_2017.json.gz>

***LANGUAGE***

Language support on TMDb is based on the language query parameter you send along with your request. You can specify one or two values (separated by a dash) with this parameter. A simple example to request german information of a list items looks like –

https://api.themoviedb.org/3/movie/76341?api\_key=<<api\_key>>&language=de

***IMAGES***

In order to get a full image url here, we need 3 pieces of data –

1) base url

2)file\_path

3)file\_size

e.g – If you are looking for image with size w500, then the url will look like this –

https://image.tmdb.org/t/p/w500/kqjL17yufvn9OVLyXYpvtyrFfak.jpg

Company and network logos are available in two formats, SVG and PNG. All of the logo\_path fields will return a .png. When looking at the image methods there is a new field called file\_typethat will show you the original version of the asset that was uploaded. For SVG's, you should call the original image size since we don't resize them. If you prefer to grab PNG's, you can call any size you wish just like normal.

e.g. of Netflix logo if you would orginal you would get file in svg format which you can’t resize -

<https://image.tmdb.org/t/p/original/wwemzKWzjKYJFfCeiB57q3r4Bcm.svg>

But if you want to get a png file, you can get it in original size or you can even adjust its size. E.g –

<https://image.tmdb.org/t/p/original/wwemzKWzjKYJFfCeiB57q3r4Bcm.png>

<https://image.tmdb.org/t/p/w500/wwemzKWzjKYJFfCeiB57q3r4Bcm.png>

***IMAGES LANGUAGES***

https://api.themoviedb.org/3/movie/550/images?api\_key={api\_key}&language=en-US&include\_image\_language=en,null

Therefore in tha above url Notice the include\_image\_language parameter. This query (en,null) is looking for all images that match English **and** those that haven't been set yet (null) in the movies tab which has the id as 550. It will give us all the images stored in the tmdb api.

You can do the above in a more better way using the append\_to\_response query

https://api.themoviedb.org/3/movie/550?api\_key={api\_key}&language=en-US&append\_to\_response=images&include\_image\_language=en,null

This will not only provide us with the images of that particular

Movie but will also provide us with the other imformation require like backdrop image, movie release data , etc

***APPEND TO RESPONSE QUERY***

This makes it possible to make sub requests within the same namespace in a single HTTP request. E.g you want the movie details and video of a movie. Then you would request it in 2 different request like this –

https://api.themoviedb.org/3/movie/157336?api\_key={api\_key}

[https://api.themoviedb.org/3/movie/157336/videos?api\_key={api\_key}](https://api.themoviedb.org/3/movie/157336/videos?api_key=%7bapi_key%7d)

But if we want to make these request in a signle command , we should use Append to Response parameter

https://api.themoviedb.org/3/movie/157336?api\_key={api\_key}&append\_to\_response=videos

Like we also want movie details along with the image and videos of the movie we can do this.

https://api.themoviedb.org/3/movie/157336?api\_key={api\_key}&append\_to\_response=videos,images

***REGION PARAMETER IN THE QUERY***

For example if you are searching for a movie named Whiplash to show German release data , you can make a query like to

[https://api.themoviedb.org/3/search/movie?api\_key=<<api\_key>>&query=whiplash&language=de-DE&region=DE](https://api.themoviedb.org/3/search/movie?api_key=%3c%3capi_key%3e%3e&query=whiplash&language=de-DE&region=DE)

It will search for the movie whiplash and get all the details . In case we do not enter the region parameter or the value of the release data is not entered to our data for that region, then we would directed back to the default primary release

There is also a with\_release\_type filter which can work with the filter “region”. For example we are using the movie discover method to find all the movies which are in theatres in germany this week.

[https://api.themoviedb.org/3/discover/movie?api\_key=<<api\_key>>&language=de-DE&region=DE&release\_date.gte=2016-11-16&release\_date.lte=2016-12-02&with\_release\_type=2|3](https://api.themoviedb.org/3/discover/movie?api_key=%3c%3capi_key%3e%3e&language=de-DE&region=DE&release_date.gte=2016-11-16&release_date.lte=2016-12-02&with_release_type=2|3)

Note while using the region parameter on discoverit is necessary to use the release\_type parameter otherwise it won’t work

***SEARCH AND QUERY FOR DETAILS***

We will first search for a movie(or TV show etc) and then query for its details

1. Search : First we are going to issue a query for one of the query like this

[https://api.themoviedb.org/3/search/movie?api\_key={api\_key}&query](https://api.themoviedb.org/3/search/movie?api_key=%7bapi_key%7d&query) = Movie Name

1. From the above we will get details about the movie and through its id , we can query all its details using movie detail method

[https://api.themoviedb.org/3/movie/343611?api\_key={api\_key}&append\_to\_response=videos](https://api.themoviedb.org/3/movie/343611?api_key=%7bapi_key%7d&append_to_response=videos)

***POPULARITY***

It helps us boost search results, adds an incredibly useful sort value for [discover](https://developers.themoviedb.org/3/discover/movie-discover)

***TRENDING***

***This is a really good github project for view Pager animations***

<https://github.com/dipanshukr/Viewpager-Transformation>

***This is a very good site to learn how to imbed play youtube videos in your app. Just follow these rules and you will be all done***

<https://www.sitepoint.com/using-the-youtube-api-to-embed-video-in-an-android-app/>

***After going through the above one. Just look at this one too. This will help us to integrate some extra features to our youtubeView***

<http://android-coffee.com/tutorial-play-youtube-video/>.

This youtube api provides some methods which can be evoked to make it more user friendly like enable them to jump back and forth in the video by some seconds, or enable them to play the next or previous video in a playlist.

***We’ll first learn how to jump to a specified time in the video.***

Button seekToButton = (Button) findViewById(R.id.seek\_to\_button);

seekToButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

int skipToSecs = Integer.valueOf(seekToText.getText().toString());

player.seekToMillis(skipToSecs \* 1000);

}

});

I have used this libriary to make expandable text view . It is a very good one

<https://github.com/Manabu-GT/ExpandableTextView>