TheMovieDB API

Daniel Weibel

February 2017

The "movie" objects in the responses of the API calls:

- /movie/popular
- /movie/now_playing
- /movie/top_rated
- /movie/upcoming

are identical.

The fields of a movie object are:

```
poster_path (string or null)
adult (boolean)
overview (string)
release_date (string)
genre_ids (array[integer])
id (integer)
original_title (string)
original_language (string)
title (string)
backdrop_path (string or null)
popularity (number)
vote_count (integer)
vote_average (number)
```

The corresponding member variables of a Java class (e.g. for conversion betwen JSON objects and Java objects by Gson) are:

```java
String posterPath;
boolean adult;
String overview;
String releaseDate;
int[] genreIds;
int id;
String originalTitle;
String originalLanguage;
String title;
String backdropPath;
double popularity;
int voteCount;
double voteAverage;
```

Note: for a correct conversion between underscore-delimited and camel-case identifiers, the method:

```java
setFieldNamingPolicy(FieldNamingPolicy.LOWER_CASE_WITH_UNDERSCORES);
```

must be called on the `GsonBuilder` object.

The complete responses of the above API calls look like this:

```
{
    page:
    results: []
    total_pages:
    total_results:
}
```

The field `results` is an array of movie objects as described above.

Note: the API calls */movie/now_playing* and */movie/upcoming* in addition have the field `dates: {}`.