

Please read these specifications carefully. The test requires you to follow all of them correctly.

Objective

The goal of this project is to have a simple web service that stores a list of movies and a set of favorite movies per user.

- Users will be able to see the list of movies and favorite movies from an Android app and an iOS app
- Admins will be able to see the list of movies and manage movies (add, edit, delete) from a web browser. They should be able to see what movies each of the existing users have favorited.

Your objective is to build the server, API and admin pages for this service

Server + Admin pages requirements

- All the code you create for this test has to be made available in a github public repository. It can be in your personal github account. Please give access to user <https://github.com/diegopau> to your repository.
- The server must be developed using Ruby on Rails framework. The database must be PostgreSQL or MySQL.
- A simple **fully RESTful** API will allow the frontend / iOS and Android app to download the list of movies and communicate to the server what are the favorite movies of a given user. Designing the API is one of your tasks
- You can host the server temporarily in Heroku or any other host of your preference. If you need us to provide you hosting we will provide you a Digital Ocean droplet with Ruby on Rails installed on it.
- For the admin pages we don't require any kind of specific design, it can be as simple as needed, the design won't be evaluated, we just need it to work.
- We don't need to setup a domain, the admin pages should be accessible at `http://xxx.xxx.xxx.xxx/admin` (xxx.xxx.xxx.xxx = IP address of the server) or `https://domain.com/admin`

- We have opened a Skype group chat for any doubts you may have, this is the place to ask any of your doubts: <https://join.skype.com/nQAaTJddpa5G>

Specifications

Data and functionality

1. We need to build a database of movies on our server. All the movies in the database will be introduced manually by the admin on the admins pages.
 - a. From the admin pages the movies can be created, edited and deleted. There will also be an admin page to list all the existing movies. There will also be pages to view the details of a movie and to edit the movie info.
 - b. From the apps the movies can only be viewed. We will be able to see:
 - i. The full list of movies from which movies can be favorited/unfavorited
 - ii. A page with details of a single movie
 - iii. A list of favorite movies. The movies favorited by the user

This is the data that a movie can have:

Name	Mandatory / Optional	Details
Name	Mandatory	80 characters limit, any unicode character is accepted
Genres	Mandatory	At least 1 genre is mandatory. New genres can't be added, there will be a set of predefined genres to choose from. Multiple genres can be selected.
Year	Mandatory	A year with 4 digit format (ex. 1982)
Director	Optional	60 characters limit, any unicode character is accepted
Main star	Optional	60 characters limit, any unicode character

		is accepted
Description	Optional	400 characters limit, any unicode character is accepted
Favorited	Mandatory	This is not a field the admins will fill up. It's a field that defaults to 0 and that shows the total number of users that favorited the movie

This is the list of possible genres:

- Action
- Adventure
- Animation
- Biography
- Comedy
- Crime
- Documentary
- Drama
- Fantasy
- History
- Horror
- Musical
- Mystery
- Romance
- Sci-Fi
- Thriller
- War
- Western

2. We also need a database of users and the list of favorite movies for each user. There will be a single field for each user:

username	mandatory	20 characters limit, only alphanumeric characters and the underscore “_” are allowed
----------	-----------	--

3. We won't have a full login/authentication system to identify users. It will be the most simple approach possible:
 - Only the username will be used to identify a user. No passwords will be used.
 - Users can be created only from the apps (or from postman, using the API endpoint)
 - First time the app is opened it will ask for a username. If the username exists then it will load that user's list of favorite movies. If the username does not exist it will create a user with that user name.
 - Each user will have a different set of favorited movies. Movies can only be favorited from the apps (or from postman, using the API endpoint)
 - All we do to authenticate the user is to check if the username already exists in the database, if the user does not exist it will be created and will send a response to the app acknowledging that a new user has been created, if the user exists then the server must send a different response to the app indicating that the user already exists.

Admin pages

1. Please allow access to the admin pages on this URL <http://xxx.xxx.xxx.xxx/admin> or <https://domain.com/admin>. We don't need several admin users, just one. Once all is setup and completed please deliver the credentials to access the admin pages to diego@scissorboy.com, together with anything else we need to know about the admin pages and the server implementation.

Please also deliver in that email a short documentation showing all the API calls available for the apps. Alternatively you can have this documentation in the Readme of your github repo.

2. We will need the following 2 main sections on the admin pages:
 - a. Movies
 - i. We need to be able to see a list of all existing movies in a table. The table only needs to show the name of the movie "Name", the year of the movie "Year" and the number of users that favorited the movie "# times favorited". Also, for each row of the table there has to be a "Delete" link that will remove the movie from the database.

- ii. At the top of the list there will be a button “Add Movie” that will open the form to create a new movie. The form will have these components:
 - A text field for the movie name
 - A dropdown or any of the type of field that allow multiple selection for the movie genres
 - A year selection field. Could be a calendar or a dropdown as long as allows to select a year ranging from at least 1900 to 2019
 - A text field for the Director
 - A text field for the Main star
 - A textarea field for the description

At the bottom of the page we should have 2 buttons: “Create movie” to create the movie record and “Back” to go back to the movie list without making changes.

- iii. The name of the movie in the list should be clickable and it will open the edit page. From the edit page all the fields of the movie can viewed and modified (except “Favorited”) which is a read-only field that doesn’t need to be shown on the edit page. There should be at the bottom of the edit page 2 buttons: “Update” to update the movie record with the changes done on the edit page and “Back” to go back to the movie list without making changes.

b. Users

- i. We need to be able to see a list of all the existing users. It will be a table with the username “username”, the number of favorited movies for that user “# favorited movies”, and a “Delete” link that will remove the user from the database.
- ii. There will not be a way of creating users from this page. Users are only created from the app
- iii. In the list of users the “username” can be clicked. It will open a new page that shows:
 - The user name
 - The list of movies favorited by the user, only the name of the movies must be shown.

- A “Back” button to go back to the user list page.

3. Please add a logout button to end the admin session

API endpoints

The API should cover the following actions from the app:

- Check if user exists in the database
- Get the list of all the movies in the database
- Get the details of a specific movie in the database, including the number of users that favorited it
- Get list of favorite movies for a specific user
- Favorite a specific movie (for a specific user)