

---

## 1 GOAL:

Create a simple application from scratch using Angular. We will use the Star Wars public API to make a web on this theme.

---

### 1.1 Considerations

#### Before Starting

We recommend reading the entire exercise first before starting to develop. Above all, we ask you to focus on the minimum objectives.

#### API to use

The API to be used will be the following:

- <https://swapi.dev/documentation#intro>

In the documentation you can find all the urls needed to perform the exercise.

#### How to deliver

To deliver the project you can use any of the following:

- Share the code using stackblitz
- Sharing the code using a public repository
- Send a zip file with the whole project

In all cases you must attach clear instructions on how to install and start the application.

#### Main considerations

- No application styling is required. Focus on well-crafted and clear functionality with good code quality.
  - It is not necessary to handle the possible errors that the API may give.
  - Is it not necessary manage loading behaviors.
  - No need to do more than what is proposed. Any extra development will not be considered or valued as more positive.
-

---

## 2 Exercise

---

### 2.1 First step

Note: It is an indispensable condition that the application has at least 2 modules.

The application will have a header in all the pages that it navigates. The following is the HTML of the header to add (also attached css to add a series of styles):

Header: See Appendix 3.1 and 3.2

At the top of the main screen, put in Links the resources available in the API. To get the resources you don't have to make a call, they have to be hardcoded.

<https://swapi.dev/documentation#root>.

If you need to do a search, check this link:

<https://swapi.dev/documentation#search>

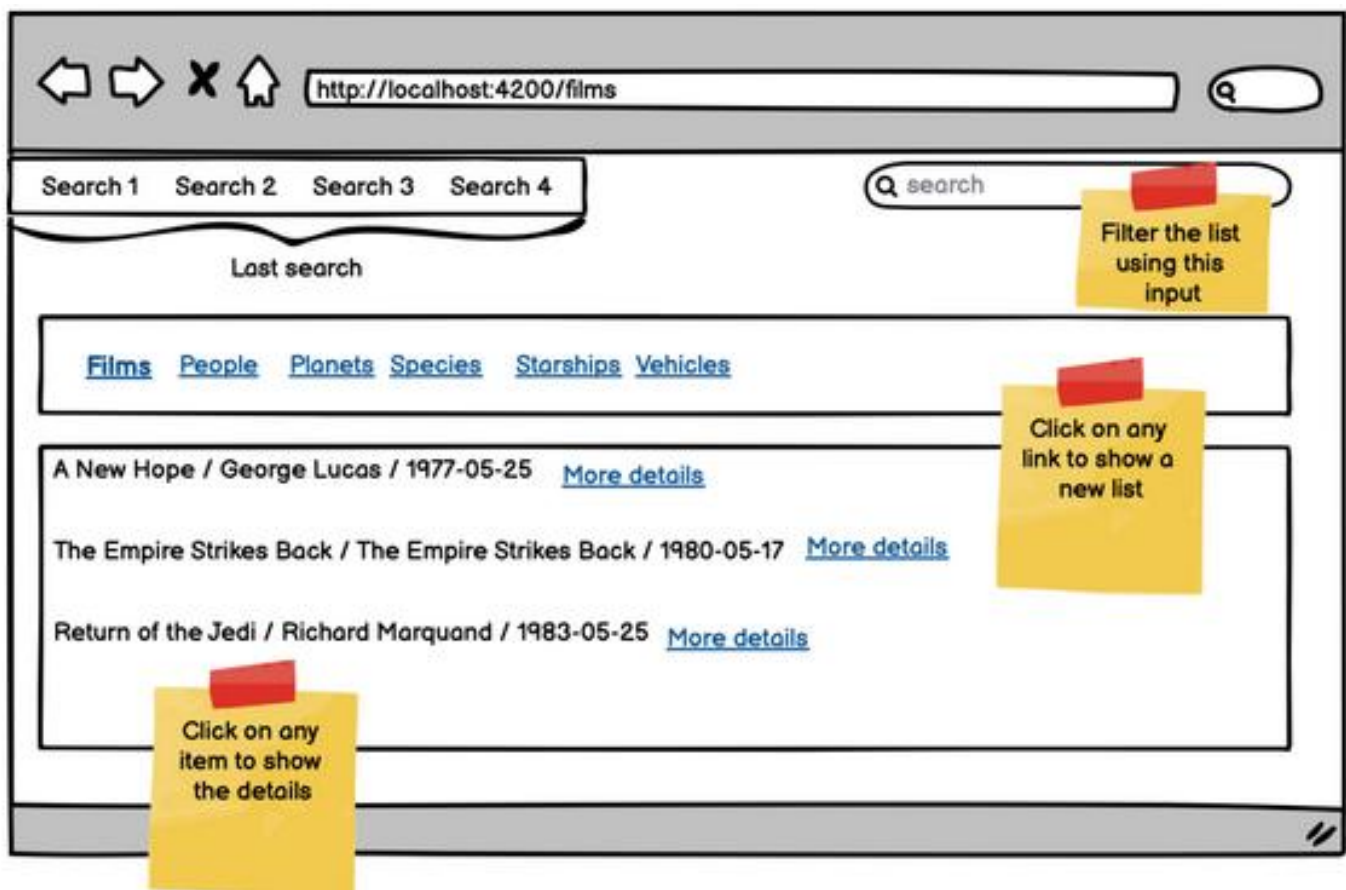
Each link should be clickable to load the information provided in a list. Additionally, 3 information fields should be displayed (**See example below**).

Note: Implement **people** link is mandatory, and one other of your choice. All other links should display a work in progress. This list will have to show by name the planet to which each person belongs ("homeworld").

- "films": No attributes to map.
- "people":
  - **height**: > 200 (high) // 100 - 200 (normal) // < 100 (low). Set these texts instead the number value. Height is mandatory field.
  - Resolve name of the planet of the property **homeworld**, is mandatory field.
- "planets": No attributes to map
- "species": No attributes to map
- "starships": **length**: > 1000 (large) // 100 - 1000 (normal) // < 100 (small). Set these texts instead the number value. Length is mandatory field.
- "vehicles": No attributes to map

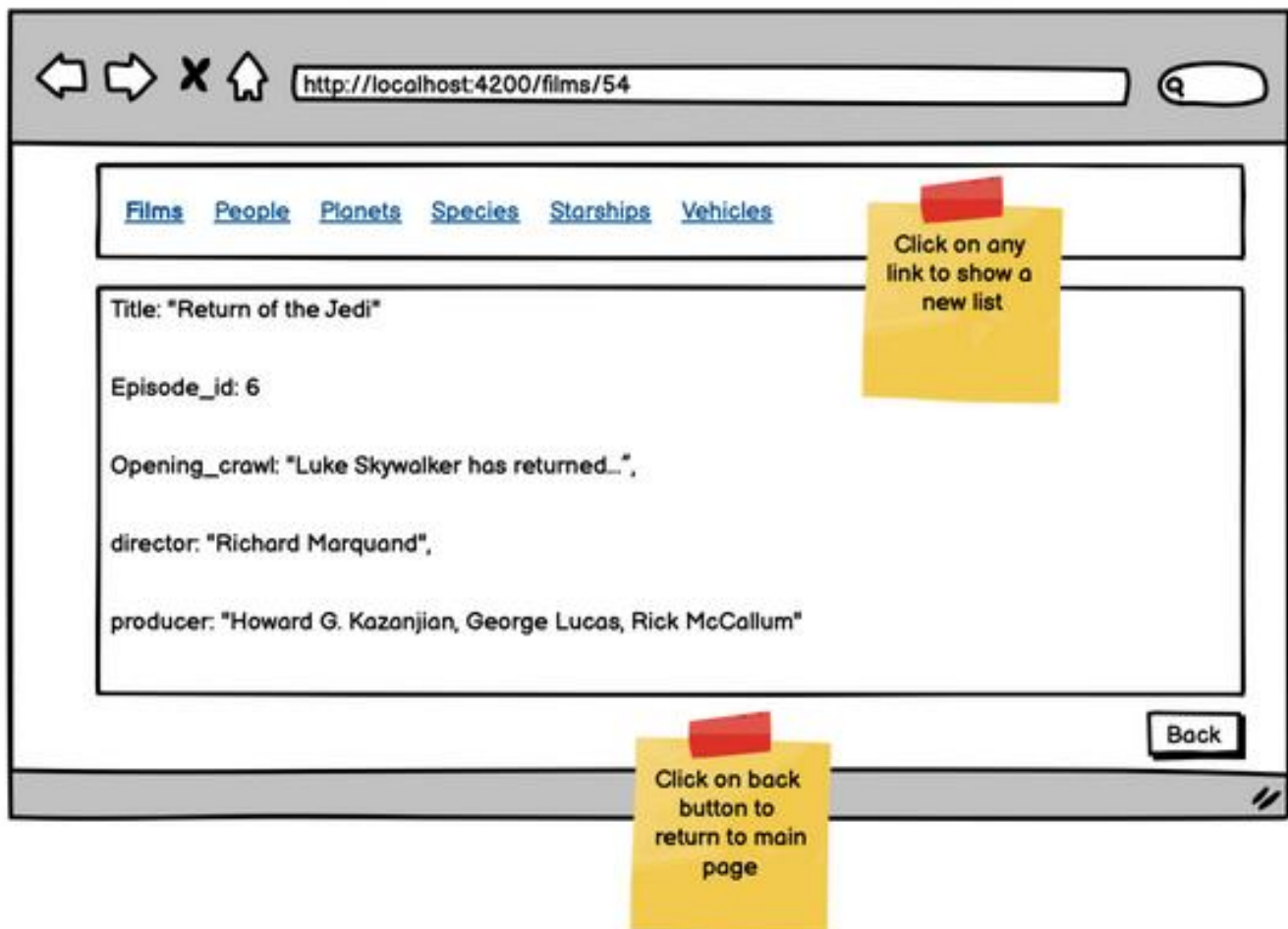
The list can be filtrable (in the header) and the last searched string (the last 4 minimum) must be saved in the storage and proposed to reuse these with a buttons or links

IMPORTANT: It is not necessary to implement pagination, displaying the first 10 elements at most is enough.



## 2.2 Second step

In the list, when we click on one of the items, we will change to a page where the details of the selected item are shown, with a button to return to the main page.



---

## 3 Appendix

---

### 3.1 MAIN CSS

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
```

```
*{box-sizing:border-box}body{margin:0;font-family:Arial,Helvetica,sans-serif}.topnav{overflow:hidden;background-color:#e9e9e9}.topnav a{float:left;display:block;color:#000;text-align:center;padding:14px 16px;text-decoration:none;font-size:17px}.topnav a:hover{background-color:#ddd;color:#000}.topnav a.active{background-color:#2196f3;color:#fff}.topnav .search-container{float:right}.topnav input[type=text]{padding:6px;margin-top:8px;font-size:17px;border:none}.topnav .search-container button{float:right;padding:6px 10px;margin-top:8px;margin-right:16px;background:#ddd;font-size:17px;border:none;cursor:pointer}.topnav .search-container button:hover{background:#ccc}@media screen and (max-width:600px){.topnav .search-container{float:none}.topnav .search-container button,.topnav a,.topnav input[type=text]{float:none;display:block;text-align:left;width:100%;margin:0;padding:14px}.topnav input[type=text]{border:1px solid #ccc}}
```

---

### 3.2 HEADER HTML

```
<div class="topnav">
<a>Search 1</a>
<a>Search 2</a>
<a>Search 3</a>
<div class="search-container">
<form>
<input type="text" placeholder="Search.." name="search">
<button><i class="fa fa-search"></i></button>
</form>
</div>
</div>
```