

DEVELOPER CHALLENGE: MARVEL CHARACTER PROFILE

Due Date: Friday 15 April

URL resources:

<https://angular.io/>
<https://material.angular.io/>
<https://material.angular.io/cdk/drag-drop/overview>
<https://nodejs.org/en/>
<https://developer.marvel.com/>
<https://developer.marvel.com/docs>
<https://developer.marvel.com/documentation/generalinfo>

You must add the following to the Headers, to use the marvel API.

Referrer: <https://developer.marvel.com>

You will need to get an API key.

REQUIREMENTS

1. Create a single page web application using Angular 8, Angular Material, Angular Material CDK, and NodeJS (Express) to expose REST endpoints from **gateway.marvel.com** to display specific data from selected Marvel superheroes. You may not replicate the Marvel API locally, the data must be queried from source and displayed as per the structure provided. The application must dynamically render on desktop and mobile screens, resizing appropriately. No hardcoding of any values. Keep as much of the logic in NodeJS as possible.
2. The web application must run on localhost. When you open the web application, display a page with the following:
 - A button to return data from the Marvel gateway as per [#3]
 - A scrollable window where a list of characters must be listed as per [#4]. The scrollable window needs to be set up as a cdkDropList (drag and drop source list)
 - A smaller window in the top right that would be a drop zone (cdkDropList) for an item dragged from the 'List of Characters' window. Each item dropped must replace the previously dropped one so there is only ever one item in the 'Drag Characters Here' window. See [#5].
 - A window in the lower right side of the screen where the character dropped in the previous window is displayed. It needs to display the character name, character picture (where available), stories where the character is featured and comics where the character appeared. See [#6].

See the image on the next page for a layout design.

MARVEL CHARACTER PROFILES

List of Characters

GET CHARACTERS

Drag Character Here

Displayed Character Info

3. Create an endpoint in your NodeJS to request Characters from the Marvel gateway as `/api/marvel/characters`. Structure:

Characters return object

```
{
  "characters": [
    {
      "id": 1001870,
      "name": "Ajax"
    }
  ]
}
```

4. When you click the button in [#2], populate the box on the left with the list of character data returned, displaying “name” from [#3] as a draggable object.

MARVEL CHARACTER PROFILES

List of Characters

- * 3-D Man
- * 8-Ball
- * Aardwolf
- * Aberration
- * Abyss
- * Aegis
- * Aero
- * Aftershock
- * Agatha Harkness
- * Ahab
- * Airstrike
- * Ajax

GET CHARACTERS

Drag Character Here

Displayed Character Info

5. Dragging a name from 'List of Characters' to 'Drag Character Here' must work as follows:
- Only one name should exist within the list.
 - When a name is dropped in the target window, it needs to execute an event to obtain additional data as per [#6]

MARVEL CHARACTER PROFILES

List of Characters

* 3-D Man

* 8-Ball

* Aardwolf

* Aberration

* Abyss

* Aegis

* Aero

* Aftershock

* Agatha Harkness

* Ahab

* Airstrike

* Ajax

GET CHARACTERS

Drag Character Here

Ahab

Displayed Character Info

6. As per [#5], when a name is dropped in the window, an event will need to fetch the image of the character, the comics and stories where that character appeared in. See [#7] and [#8].

MARVEL CHARACTER PROFILES

List of Characters

- * 3-D Man
- * 8-Ball
- * Aardwolf
- * Aberration
- * Abyss
- * Aegis
- * Aero
- * Aftershock
- * Agatha Harkness
- * Ahab
- * Airstrike
- * Ajax


GET CHARACTERS

Drag Character Here

Ahab

Displayed Character Info

Ahab



STORIES
A beginning
Utility of Myth

COMICS
Avengers: The Initiative
Deadpool

- Expose Rest Endpoint on same node server to integrate to gateway.marvel.com to build a character profile. /api/marvel/character/profile

The Node End Point must return the following object, not the standard Marvel structure. You will make use of the various Marvel end-points in the middle layer to build up the full Hero profile.

Profile return object

```
{
  "characters": {
    "name": "Ajax",
    "imagePath": "",
    "stories": [
      {
        "name": "",
        "title": "",
        "description": ""
      }
    ],
    "comics": [
      {
        "title": "",
        "issueNumber": "",
        "description": ""
      }
    ]
  }
}
```

- Use standard medium as the image variant name. To build up the image for display:

```
{
  "thumbnail": {
    "path": "http://i.annihil.us/u/prod/marvel/i/mg/3/40/4bb4680432f73",
    "extension": "jpg"
  }
}
```

Example: http://i.annihil.us/u/prod/marvel/i/mg/3/40/4bb4680432f73/standard_medium.jpg

- Upload to Git and email the project link to:

sunette.lagrange@metropolitan.co.za

Tharbeat.Hendricks@metropolitan.co.za

james.fischer@mmltd.co.za