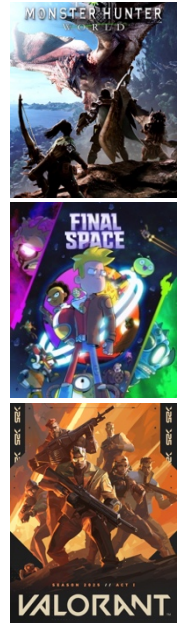# Expanding Your Angular Web App with More APIs

Integrate additional APIs into your existing Angular web app. You will retrieve data from new public APIs, process the information, and display it in a card-based layout, similar to what you've already built in the Rick and Morty section.

1. Use these **free public APIs** to integrate into your web app:
   - **Monster Hunter World API (MHW-DB)** – Retrieve a list of weapons from the videogame.
     - API Docs: https://mhw-db.com/docs
     - Endpoint: https://mhw-db.com/weapons?q={"id":{"$gte":1,"$lte":20}}
     - Returns information on **weapons**.
   - **Final Space API** – Retrieve character details from the animated series *Final Space*.
     - API Docs: https://finalspaceapi.com/docs/
     - Endpoint: https://finalspaceapi.com/api/v0/character
     - Returns a list of **characters** (names, species, images…)
   - **Valorant API** – Retrieve agent information from the *Valorant* videogame.
     - API Docs: https://valorant-api.com/
     - Endpoint: https://valorant-api.com/v1/agents
     - Returns a list of **agents**, including abilities, roles, and images.

2. Fetch API Data:
   - Use Angular's **HttpClient** service to retrieve data from the selected API(s).
   - Handle asynchronous data loading properly using **Observables** and **Subscriptions**.

3. Create New Components:
   - For each API you choose, create a new Angular **component** (e.g., mhw-weapon, final-space, valorant-agent).
   - Each component should be responsible for **fetching** and **displaying** the data.

4. Update the UI:
   - **Display** the retrieved data using Angular (*ngFor, property binding, etc.).
   - Use **CSS** to ensure consistency with the existing design.
   - Ensure each new API's data is presented as a **card**, but each one with a **different look and feel**.

5. Home and Navbar Component
   - Update both components to make the new information accessible:
     - **Navbar:** new text links
     - **Home:** new image links

6. Bonus Challenges (Optional):
   - Add a **search** or **filter** feature to refine the displayed API data.
   - Implement a **loading indicator** while the data is being fetched.

7. Delivery:
   - Compress the project without the **node_modules** folder and send it using the corresponding Moodle task.
   - Deadline: **16-02-2025 23:59**