**Test 1: Confirm Pokémon number 812 is Rillaboom**

**Assertion: Ensure that the response contains the name "Rillaboom" and the correct Pokémon ID.**

**Objective: Ensure that the PokéAPI correctly identifies Pokémon number 812 as "Rillaboom."**

**Implementation: HTTP Request: GET /pokemon/812**

**TYPESCRIPT-** In TypeScript we will use the axios library for making HTTP requests, and it includes asynchronous functions to handle promises. The test function, **testPokemonDetails**, performs assertions on the response **status code, Pokémon name, and ID**, throwing an error if any assertion fails.  
import axios from 'axios';

// Function to make an API request

async function makeApiRequest(endpoint: string): Promise<AxiosResponse> {

const apiUrl = '<https://pokeapi.co/api/v2>';

const url = `${apiUrl}${endpoint}`;

return await axios.get(url);

}

// Test code

async function testPokemonDetails() {

try {

// Make API request to get Pokémon details for ID 812

const response = await makeApiRequest('/pokemon/812');

// Assert the HTTP response status code is 200

if (response.status !== 200) {

throw new Error(`Unexpected status code: ${response.status}`);

}

// Parse JSON data from the response

const data = response.data;

// Assert Pokémon name is 'rillaboom' and ID is 812

if (data.name !== 'rillaboom') {

throw new Error(`Unexpected Pokémon name: ${data.name}`);

}

if (data.id !== 812) {

throw new Error(`Unexpected Pokémon ID: ${data.id}`);

}

console.log('Test passed successfully!');

} catch (error) {

console.error(`Test failed: ${error}`);

}

}

// Run the test

testPokemonDetails();

**Test 2: Check that the 5th generation of games is set in Unova, and there are two resources in the version groups array**

* Objective: Verify the correctness of information related to the 5th generation of Pokémon games in the PokéAPI.
* Request: GET /generation/5
* Assertion: Ensure that the response contains "Unova" as the region and has exactly two resources in the version groups array.

**Typescript Code**

async function makeApiRequest(endpoint: string): Promise<AxiosResponse> {

const apiUrl = '<https://pokeapi.co/api/v2>';

const url = `${apiUrl}${endpoint}`;

return await axios.get(url);

}

(async () => {

try {

const response = await makeApiRequest('/generation/5');

const data = response.data;

// Assertions

if (response.status === 200) {

if (data.main\_region.name === 'unova' && data.version\_groups.length === 2) {

console.log('Test passed successfully.');

} else {

console.error('Test failed: Incorrect data in the response.');

}

} else {

console.error(`Test failed: Unexpected status code ${response.status}.`);

}

} catch (error) {

console.error('Error making API request:', error.message);

}

})();

**Test 3: Check that you can search for all the Pokémon in the Kanto dex**

* Objective: Validate the ability to retrieve information about Pokémon in the Kanto region
* Request: GET /pokedex/kanto
* Assertion: Ensure that the response contains all Pokémon IDs from 1 to 151 and their corresponding names.  
   response = make\_api\_request('/pokedex/kanto')

import axios, { AxiosResponse } from 'axios';

async function makeApiRequest(endpoint: string): Promise<AxiosResponse> {

const apiUrl = '<https://pokeapi.co/api/v2>';

const url = `${apiUrl}${endpoint}`;

return await axios.get(url);

}

(async () => {

try {

const response = await makeApiRequest('/pokedex/kanto');

const data = response.data;

// Assertions

if (response.status === 200) {

const kantoPokemonIds = Array.from({ length: 151 }, (\_, i) => i + 1);

for (const pokemonEntry of data.pokemon\_entries) {

// Assert entry number is in Kanto Pokemon IDs

if (kantoPokemonIds.includes(pokemonEntry.entry\_number)) {

// Assert Pokemon species name is a string

if (typeof pokemonEntry.pokemon\_species.name === 'string') {

console.log(`Test passed for entry number ${pokemonEntry.entry\_number}`);

} else {

console.error(`Test failed: Pokemon species name is not a string for entry number ${pokemonEntry.entry\_number}`);

}

} else {

console.error(`Test failed: Entry number ${pokemonEntry.entry\_number} not in Kanto Pokemon IDs`);

}

}

} else {

console.error(`Test failed: Unexpected status code ${response.status}.`);

}

} catch (error) {

console.error('Error making API request:', error.message);

}

})();