Globant Piscine

PokeAPI

Summary: Generate 'em all!

Version: 2

Contents

Chapter I

A word about this Project

Web development is an interesting career path because of the challenges and the plethora of options available to solve those problems. In this specific case, we'll focus on two of the most common libraries and frameworks to approach web development and then tackle a specific problem using AI generated images through an API.

This project is influenced by our previous experience with these types of academys where we used PokeAPI to create new applications on top of that API. The current document describes a new twist on that idea and allows us to explore new tools like AI generated images and content and include them in real-world applications.

Chapter II

Introduction

What this Project will show you:

- Usage of a WebFramework to build a Website.
- Communication with a remote REST API.
- Managing Front-End challenges like Accessibility or SPA.
- Using AI generated images in a real-world application.
- Using Oauth2 as an authentication method.
- Managing some basic security concerns.
- Setting up a complete development environment using using Docker and Docker Compose.
- Gaining experience in building, running, and deploying production-grade applications.

Chapter III

General instructions

Unless explicitely specified, the following rules will apply for every project of this Piscine.

- This subject is the one and only trustable source. Don't trust any rumor.
- This subject can be updated up to one hour before the turn-in deadline.
- The assignments in a subject must be done in the given order. Later assignments won't be rated unless all the previous ones are perfectly executed.
- Be careful about the access rights of your files and folders.
- Your assignments WON'T be evaluated by your Piscine peers.
- You must not leave in your turn-in your workspace any file other than the ones explicitly requested By the assignments. If the assignment don't precise them, put only the necessary ones to run your Project.
- Using some API Key or Token? Keep them for you! Do not push them on your repository.
- You have a question? Ask your left neighbor. Otherwise, try your luck with your right neighbor.
- Every technical answer you might need is available in the `man` or on the Internet.
- You must read the examples thoroughly. They can reveal requirements that are not obvious in the assignment's description.
- By Loki, by Freya! Use your brain!!!

Chapter IV

Glossary

This glossary is designed to help you quickly understand the main technologies and concepts you will work with in **PokeAPI**. In Chapter VII, you'll find extra resources and tutorials to get started confidently.

- **TypeScript**: A programming language that extends JavaScript, used mainly in frontend and backend web development for greater safety and scalability.
- **Tailwind**: A CSS framework for creating responsive and customizable user interfaces, commonly used in frontend design.
- **Framework**: A predefined structure of code (libraries, components, architectures) that provides a foundation for your application. It saves you from reinventing the basics and guides where and how to organize your code.
- **Docker**: A platform that allows you to create and run applications inside isolated containers.
- **Docker Compose**: A tool for defining and managing multiple Docker containers at once.
- **OAuth2**: A standard authorization protocol that provides secure access to resources on behalf of a user and helps protect user credentials.
- **SPA (Single Page Application)**: A web application that loads a single HTML page and dynamically updates its content without reloading the entire page.
- **JWT (JSON Web Token)**: A method for securely transmitting information between two parties in JSON format.
- **API (Application Programming Interface)**: An interface that allows two applications to communicate and share data.
- **Stable Diffusion**: An open-source platform offering free APIs to generate images, text, and audio from prompts.
- **Pollination.IA**: A generative AI service used to create images from text prompts.
- **WCAG 2.1 Level AA**: A set of web accessibility guidelines designed to make websites more usable for people with disabilities.

- **Insomnia / Postman**: API testing tools used during development and debugging.

Chapter V

Mandatory part

Globant	Exercise 00
PokeAPI	
Turn-in directory : *ex*00/	
Files to turn in : `All needed files to run your Project and nothing else`	
Allowed functions : `None`	

- You must include a **README.md** file in your repository explaining briefly the project and how to run it.
- **Docker** is mandatory for this project. You must provide a Dockerfile and a docker-compose.yml file to run your project.
- You will need to use Typescript, Vite and Tailwind to build your project.
- Using our project web application, users will be able to create new Pokemons from a list of available options to combine and share the results with their friends.
- Application should be developed using a **mobile-first** approach, supporting mobile users since the start of development.
- The application will be a **Single-Page Application (SPA)**. Users should be able to use back and forward buttons in the browser.
- Users will be able to create new Pokemons by selecting from a list of potential animal combinations and abilities.
- Users will be able to share the results as images generated by the application.
- Accessibility is very important for us. Your application should aim to support **WCAG 2.1 AA** level as much as possible. We will not enforce AA strict compliance but it's important to have an understanding of the subject and some experience with it's implementation at the end of the project.

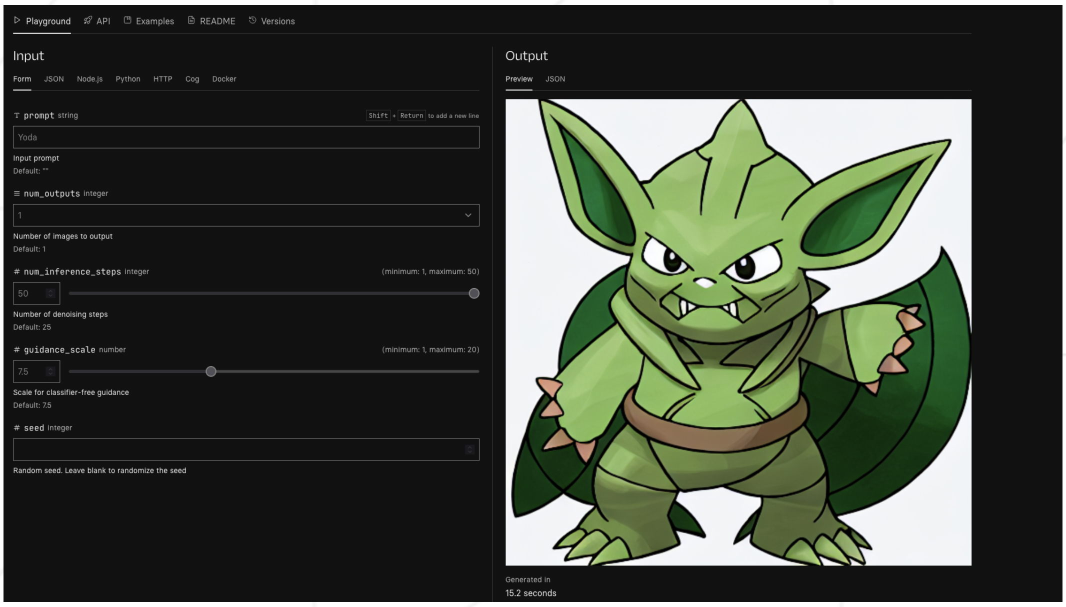
- Image generation using some 3rd party **REST API**.
- Use some good basic security measures, API keys, JWT tokens, etc.
- User authentication will be done using OAuth.

`Recommanded tools: Insomnia and Postman for API testing.`

`You may want to take a look at every technologies referenced in the project description before starting.`

`You can use the following API to generate images: http://stable-diffusion.42malaga.com:7860/, API Documentation, API Swagger Documentation`

`In any of the projects that require the consumption of an API you can leverage the use of a library to help you with it like swe or TanStack Query`



Input

prompt string — Shift + Return to add a new line

Yoda

Input prompt
Default: ""

num_outputs integer

1

Number of images to output
Default: 1

num_inference_steps integer (minimum: 1, maximum: 50)

50

Number of denoising steps
Default: 25

guidance_scale number (minimum: 1, maximum: 20)

7.5

Scale for classifier-free guidance
Default: 7.5

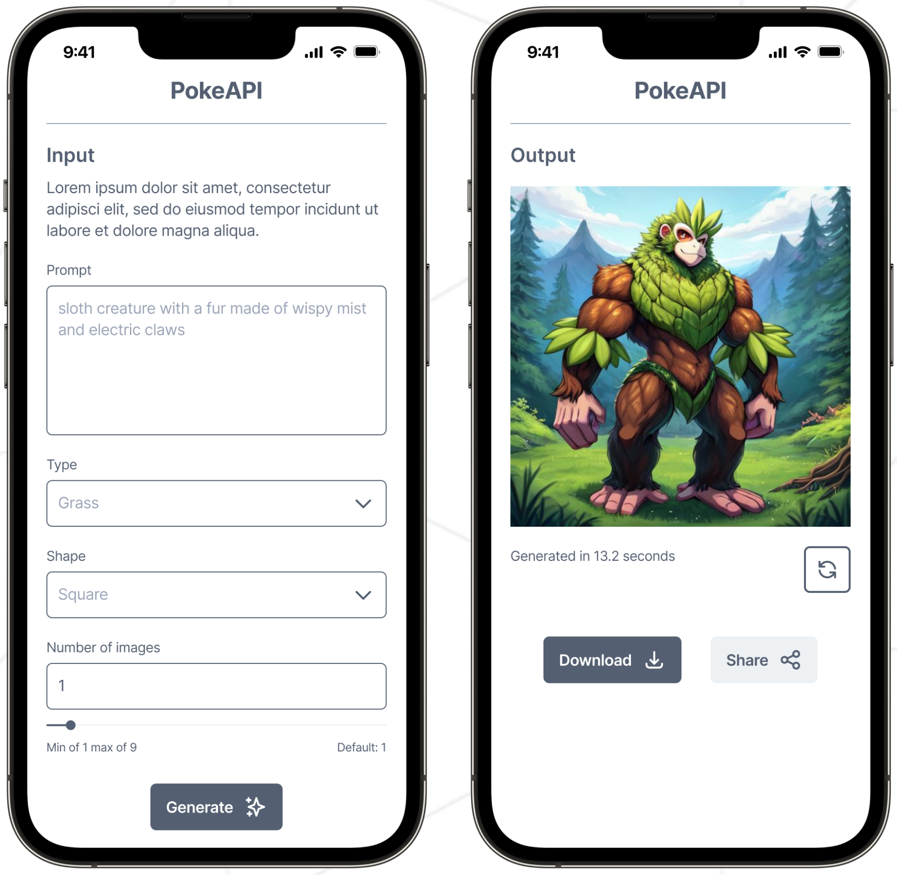
seed integer

Random seed. Leave blank to randomize the seed

Output

Generated in
15.2 seconds

Figure V.1: Example from to lambdal/text-to-pokemon – Run with an API on Replicate



Input

Lorem ipsum dolor sit amet, consectetur adipisci elit, sed do eiusmod tempor incidunt ut labore et dolore magna aliqua.

Prompt

sloth creature with a fur made of wispy mist and electric claws

Type

Grass

Shape

Square

Number of images

1

Min of 1 max of 9 — Default: 1

Output

Generated in 13.2 seconds

Figure V.2: Example UI for the project.

Chapter VI

Bonus Part

Once mandatory requirements are implemented in the application, you can add any additional features you see fit. Here a quick list of suggestions to give you ideas.

- Share can also include a URL to redirect your friends to your application or send by email for example. If you choose to implement these, some kind of unique identifier should be generated to avoid potential conflicts with the shared results.
- When creating new Pokemons, free text input could be supported.
- Image input could be added as well, providing another way to input the required information to generate new Pokemon.
- I18n support for internationalization and multiple languages.
- If you are super deep on AI learning, you could also train your own model and make it work with this project's frontend.

`Those are only examples. The sky's the limit so again, once mandatory functional requirements are met, we are open to any additional functionalities you see fit.`

Chapter VII

Resources

This chapter provides a collection of useful resources to help you get started with the technologies used in this project.

You'll find links to articles, tutorials, and videos — both in English and Spanish — as well as simple code examples to help you understand key concepts and start experimenting on your own.

These materials are meant to guide you a little if you get stuck, but we highly recommend that you keep exploring on your own.

- **Docker**
  - Article 1
  - Article 2
  - Article 3
  - Video
- **What is a framework**?
  - Article 1
  - Article 2
  - Article 3
- **Typescript**
  - Article
  - Video 1
  - Video 2
  - Video 3

We provide you an **example** below as a basic example to illustrate **React** + **TypeScript** usage. It does not include the full project setup. You still need to:

  - Create your index.html file.
  - Configure main.tsx to render your React app.
  - Set up tsconfig.json for TypeScript.
  - Install dependencies and configure Vite (vite.config.ts).

```
import React, { useState } from "react";

function App(): JSX.Element {
  const [count, setCount] = useState<number>(0);

  const handleClick = (): void => {
    setCount(count + 1);
    };

    return (
      <div style={{ textAlign: "center", marginTop: "2rem" }}>
      <h1>Hello World!</h1>
      <p>Okeeey, let’s goooo</p>

      <hr style={{ margin: "2rem 0" }} />

      <h2>Counter</h2>
      <p>You have clicked {count} times</p>
      <button onClick={handleClick} style={{ padding: "0.5rem 1rem", \
      fontSize: "1rem" }}>
      Click me
      </button>
      </div>
      );
      }

  export default App;
```

Chapter VIII

Submission

- Create a git repo (Github, Gitlab, Bitbucket, etc) and add your project files to it.
- Copy the link to your repository and paste it in the project submission form.
- Project submission form: TYPEFORM

Please note, no modifications made on the repo after the form is sent will be taken into account for the evaluation.

No Peer evaluation for this Piscine, but we strongly recommend reviewing and using the evaluation sheet we give you to check if your project meets all the requirements.