Technical Documentation - Finerio Connect Assessment

1. Introduction

1.1. Project Overview

This document serves as a technical guide for the project “finerio-assessment”. This project mainly utilizes the Strategy Pattern to implement flexible web automation capabilities, specifically for interacting with different web pages and handling different captcha types.

1.2. Audience

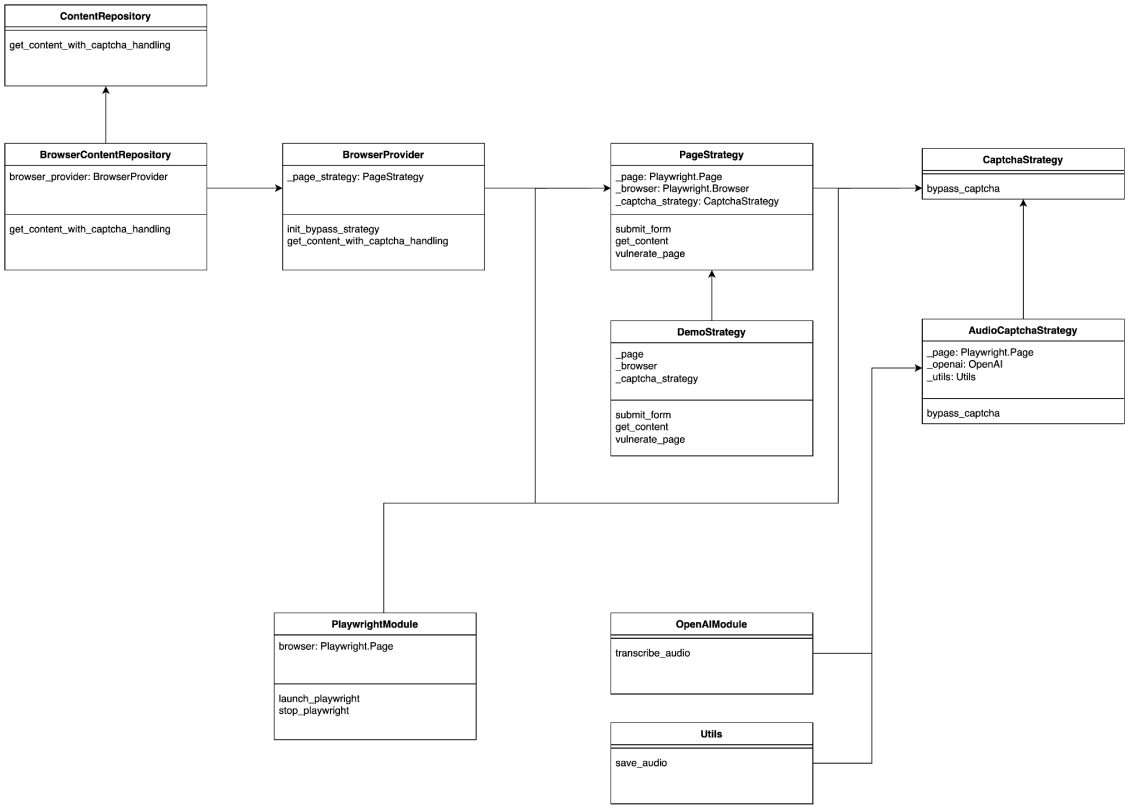
This documentation is intended for developers familiar with Python and web automation concepts.

2. Architecture Overview

2.1. UML Diagram

The diagram below shows the next classes:

- BrowserContentRepository: Separates browser provider with business logic and REST controller
- BrowserProvider: This class choose the PageStrategy with the CaptchaStrategy
- PageStrategy: Abstract class that contains the logic behind a specific page, pattern Strategy was used since different providers would have a different access method, in this particular case DemoStrategy is defined since the solution only applies for demo pages
- CaptchaStrategy: Abstract class that contains the logic behind different Captcha options, for example in this particular case AudioCaptcha Strategy was developed, this class interacts with OpenAIModule in order to get the transcription of the audio



2.2. Data flow

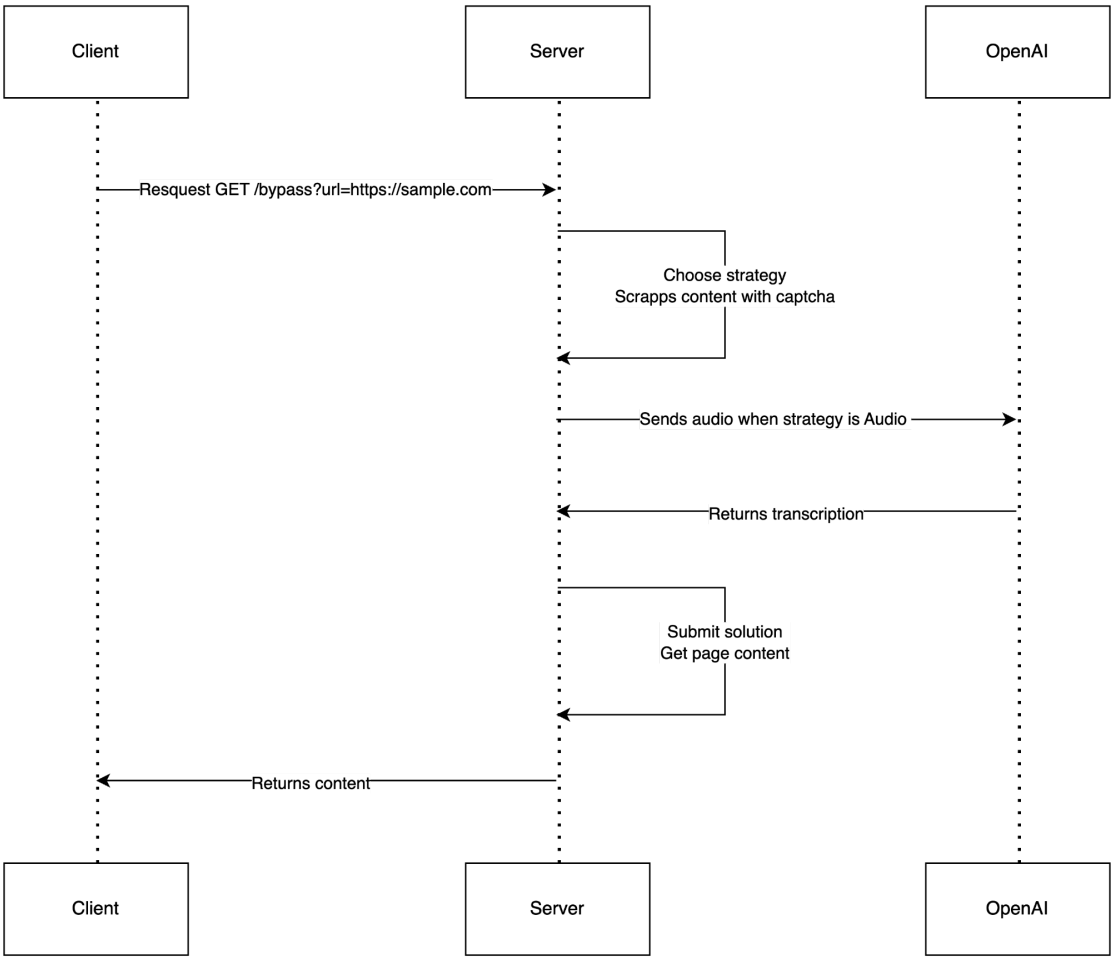
The data flow involves the next steps:

- The main application initiates the automation process when /bypass?url=https://sample.com is requested
- The browser provider selects the appropriate page strategy and captcha strategy

- The page strategy manager analyzes the context and bypasses Captcha challenge with the given strategy
- The page strategy interacts with the web browser though Playwright the execute the actions required
- The captcha strategy interacts with the provider to solve the challenge, in this case for example, OpenAI API was used to transcribe the audio challenge
- If Captcha challenge is successfully solved, then the content is returned

2.3. Sequence diagram

The sequence diagram of the solution is shown below, in the specific case with the integration with OpenAI, just to get the transcription of the audio using Whisper-1



2.4. Technology Stack

This project utilizes the following technologies:

- Programming Language: Python 3.8
- Web Automation Framework: Playwright

- Web Framework: FastAPI
- Strategy Pattern: A design pattern for flexible code structure
- Testing Frameworks: pytest, unittest
- Transcription provider: OpenAI - Whisper-1
- Dependency Injection: Lagom

3. Setup and installation

For further information check README.md file on repository

4. Usage

For further information check README.md file on repository

5. Configuration

The .env.sample file on repository requires only the next variables

- OPENAI_API_KEY: OpenAI api key with Whisper-1 permissions to handle transcription