



TEST PROJECT

2020/02

GW-3

GO2TR.COM
Amirreza Nasiri

INTRODUCTION

This document contains a server-side test project. The targeted programming language is PHP, along with the Laravel framework. Your task is to fulfill the document requirements and features within x hours from receiving it.

The project's expected scope is server-side only; thus, keep UI and frontend assets as minimal as possible. We'll evaluate your test project based on different factors, such as the project's readability and maintainability, performance, and security.

DESCRIPTION OF THE PROJECT

The overall application should suit the following requirements:

Admins:

Authentication

Laravel Sanctum should be used for API authentication for simplicity.

Image Management

Images are models that contain a URI, a path to valid file, image dimensions (width/height) and their comprehensive `alt` and `title` attributes (They're part of `` tag in HTML).

The project also needs to provide a few APIs to upload a new image, update existing image attributes, delete, list, and get individual details. These images may be used by the operator in the posts' body.

Uploaded JPEG/PNG images should also be converted to 50% of their size as another new image.

Post Management

Posts are models with a title, an HTML content body, and a status.

The project needs to provide a few APIs to insert, update, delete, list, and get individual post models. We also need an API to either enable or disable a specific post status.

The API related to getting an individual post model should also contain the following values:

- `content_list`: Post content list (determined by `<h1>` tags)
- `processed_content`: Processed content that:
 - Adds alt, title, width, and height attributes from image models for any `` tag that uses hosted images, if not explicitly specified.
 - Lazy loads images and iframes by adding `loading="lazy"` attribute, if not explicitly specified.

Users:

Authentication

Visitors on the website can sign-up/in as users with their phone number, email, or both. If the user is new, we also need to get their name. If they want to sign-up/in with a phone number, we need to send a verification SMS that's valid for 5 hours and can be requested every 10 minutes.

If they want to sign-up/in by email, we need to send a verification code to their email. The same restriction for SMS applies here too.

Use this example API as the SMS endpoint:

Request URL: <https://example.com/send-sms>

Request Method: POST

Request Data:

- receiver: Receiver phone number
- text: Text to send in SMS

Workflow:

You need to have a GitHub account. Create a new repository and start working on the test project. We'll evaluate your repository.

Questions:

Some parts of the project (such as API responses, model naming, etc.) are not included in the document since we put it all up to you. However, in the case of having a question, contact your coordinator.