

Backend Laravel Developer Challenge: News Aggregator API

Welcome to the take-home challenge for the Backend Laravel Developer position. We are excited to see your skills and experience in action. The challenge is to build a RESTful API for a news aggregator service that pulls articles from various sources and provides endpoints for a frontend application to consume.

Requirements:

1. User Authentication:

- Implement user registration and login endpoints using Laravel Sanctum for API token authentication.
- Include endpoints for user logout and password reset.

2. Article Management:

- Create endpoints to fetch articles with support for pagination.
- Implement search functionality to allow filtering articles by keyword, date, category, and source.
- Develop an endpoint for retrieving a single article's details.

3. User Preferences:

- Create endpoints to allow users to set and retrieve their preferred news sources, categories, and authors.
- Implement an endpoint to fetch a personalized news feed based on user preferences.

4. Data Aggregation:

- Develop a system to regularly fetch and store articles from at least 3 different news APIs (see the list of data sources below).

- Implement efficient data storage and indexing for optimized search and retrieval.

5. API Documentation:

- Provide comprehensive API documentation using tools like Swagger/OpenAPI.

Data Sources:

Choose at least 3 from the following list:

1. NewsAPI
2. OpenNews
3. NewsCred
4. The Guardian
5. New York Times
6. BBC News
7. [NewsAPI.org](https://newsapi.org)

Please note that some of these APIs might require registration and authentication keys to access the data.

Challenge Guidelines:

1. Backend Project:

- Develop a Laravel-based RESTful API that fulfills all the above requirements.
- Implement proper error handling and validation for all endpoints.

2. Database Design:

- Design an efficient database schema to store articles, user data, and preferences.
- Use Laravel migrations and seeders for database setup.

3. Docker:

- Dockerize the Laravel application.
- Provide a docker-compose.yml file for easy setup of the development environment.
- Include clear documentation on how to run the project using Docker.

4. Data Aggregation:

- Implement Laravel scheduled commands to regularly fetch and update articles from the chosen news APIs.
- Store the fetched data locally in the database.
- Ensure that all data filtering and searching operations are performed on the local data, not on live API data.

5. Best Practices:

- Adhere to Laravel best practices and coding standards.
- Incorporate software development principles such as DRY (Don't Repeat Yourself), KISS (Keep It Simple, Stupid), and SOLID.
- Write clean, maintainable, and well-documented code.

6. Testing:

- Write unit and feature tests for your API endpoints and core functionalities.
- Aim for good test coverage of your codebase.

7. Performance:

- Implement caching strategies where appropriate to optimize API performance.
- Consider implementing rate limiting for API endpoints.

8. Security:

- Ensure proper security measures are in place, including protection against common vulnerabilities (e.g., SQL injection, XSS).
- Implement proper authentication and authorization checks for all protected routes.

Submission Guidelines:

1. Create a GitHub repository for your project.
2. Include a README.md file with:
 - Setup instructions (including how to run the Docker environment)
 - API documentation or link to the hosted Swagger/OpenAPI docs
 - Any additional notes or explanations about your implementation
3. Ensure your commit history shows your development process.
4. Send us the link to your GitHub repository when you're ready to submit.

We look forward to seeing your implementation! Good luck!