

# Backend Coding Challenge: Profile Enrichment Service

## Objective

The goal of this task is to build a backend service that receives user information via an API, enriches it by scraping data from an external webpage, and then returns the combined data.

## Technical Specifications

### 1. API Endpoint

You must create and expose a single API endpoint with the following properties:

- **Method:** POST
- **URL:** /users/enrich

### 2. Request Body

The endpoint must accept a JSON object with the following structure:

```
{  
  "username": "testuser",  
  "email": "test@example.com",  
  "profileUrl": "http://<URL_TO_BE_SCRAPED>"  
}
```

## Required Workflow & Logic

Your service must perform the following actions in sequence:

1. **Receive and Parse:** Accept the POST request and parse the incoming JSON body to extract the username, email, and profileUrl.
2. **Scrape External URL:** Implement a client-side scraping mechanism. Your service must make an HTTP GET request to the profileUrl provided in the request body to fetch its HTML content.
3. **Extract Data:** From the retrieved HTML, parse the document and extract the text content of the primary heading (<h1>) tag. This text will be considered the user's full name.
4. **Combine Data:** Create a new user profile object by combining the data from the initial request with the scraped data.
5. **Simulate Database Save:** For this exercise, a real database connection is not required. You can simulate this step by preparing the final user object.
6. **Return Response:** Send a response to the original caller with:
  - An HTTP status code of 201 Created.
  - A response body containing the final, enriched user profile as a JSON object.

---

## Expected Final Output

The JSON object returned by your API should conform to the following structure, using the data processed during the workflow and be suitable to be generated in a pdf or doc format.

```
{
  "username": "testuser",
  "email": "test@example.com",
  "fullName": "Test User Fullname", // This value is from the scraped <h1> tag
  "sourceProfile": "http://<URL_TO_BE_SCRAPED>"
}
```