

Rust test

Rocketseed

Instructions

Create a Rust API service with a `/transform` POST endpoint. This endpoint should take two inputs in the body of a request:

1. **transform**: with one of the following string values
 - a. "uppercase"
 - b. "lowercase"
2. **html**: A string containing HTML text

The task is to take the HTML and transform all the text content of `<p>` elements according to the **transform** given in the request.

You are free to:

- Use any Rust framework (we use Actix internally)
- Use any Rust crates (kuchikiki might be helpful)
- Handle errors any way you want
- Use any project layout. Workspaces or a single crate does not matter. Any modules breakdown is fine too
- Have any kind of CI setup if you want

Your solution should be shared as a public GitHub repository containing all source code, tests, and documentation. Please provide us with the repository URL when submitting.

We expect this task to take approximately 2-4 hours to complete.

Examples

Basic Usage

Uppercase Transform

```
POST /transform
Content-Type: application/json

{
  "transform": "uppercase",
  "html": "<p>Hello world</p>"
}
```

Expected Response:

```
HTTP/1.1 200 OK
Content-Type: text/plain

<p>HELLO WORLD</p>
```

Lowercase Transform

```
POST /transform
Content-Type: application/json

{
  "transform": "lowercase",
  "html": "<p>Hello WORLD</p>"
}
```

Expected Response:

```
HTTP/1.1 200 OK
Content-Type: text/plain
```

```
<p>hello world</p>
```

Complex Usage

Multiple Paragraphs

```
POST /transform
Content-Type: application/json
```

```
{
  "transform": "uppercase",
  "html": "<div><p>First paragraph</p><span>Not a paragraph</span><p>Second
paragraph</p></div>"
}
```

Expected Response:

```
HTTP/1.1 200 OK
Content-Type: text/plain
```

```
<div><p>FIRST PARAGRAPH</p><span>Not a paragraph</span><p>SECOND
PARAGRAPH</p></div>
```

Nested Elements

```
POST /transform
Content-Type: application/json
```

```
{
  "transform": "uppercase",
  "html": "<p>Text with <strong>bold</strong> and <em>italic</em> elements</p>"
}
```

Expected Response:

HTTP/1.1 200 OK

Content-Type: text/plain

<p>TEXT WITH BOLD AND ITALIC ELEMENTS</p>