

Challenge Idea

Make a simple API to make a profile for users with Articles in multi-languages.

Construction

• Users

- Name (json) => multi-language ex:{"en": "ahmed", "ar": "أحمد"}.
- Phone (number) Unique.
- Password (string) => Hashed.
- Email (string) => Unique.
- Verified email at (datetime).
- Verified phone at (datetime).

• Articles

- Title (json) => multi-language.
- Content (json) => multi-language.

Api Pages

• Register

- Email or Phone is required.
- Inputs example:

```
{
  "name": {"en": "ahmed", "ar": "أحمد"},
  "phone": "010123456789",
  "password": "123456",
  "email": "test@test.com"
}
```

• Verify account

- Can verify with email.
- Can Verify with phone (Create a fake SMS Api for now to verify the phone and accept immediatly).

• Login

- User can login with Email or Phone number.
- password.
- User must be verified.

• Create/Update Articles

- Token must be provided.
- All languages are required.
- Input example:

```
{
  "title": {
```

```

    "en": "Article title",
    "ar": "مقاله عنوان "
  },
  "content": {
    "en": "Article content (HTML not allowed)",
    "ar": "مقاله محتوا "
  }
}

```

• Delete Article by ID and token

• View Profile with articles

- Must provide Token.
- View profile with selected language only ex: "?lang=en" to select english.
- Password must be not visible.
- Output example:

```

{
  "errors": false,
  "results": {
    "name": "name (Show name only in Selected language)",
    "phone": "010123456789",
    "email": "test@test.com",
    "created_at": "2021-08-21T07:37:43.000000Z",
    "articles": [
      {
        "title": "Article 1 in English"
        "content": "Article Content 2 in English"
        "created_at": "2021-08-21T07:37:43.000000Z"
      },
      {
        "title": "Article 2 in English"
        "content": "Article Content 2 in English"
        "created_at": "2021-08-21T07:37:43.000000Z"
      }
    ]
  }
}

```

Acceptance Criteria

- Provided solutions should be implemented in Lumen ^8.*.
- List of languages must be according to a Config (ex: config/locales.php) file also can change it from .env file.
- Database must be include two tables only (Profiles, Articles).
- All response exceptions must have a return with appropriate status number.
- Take care about pages validation in high level.
- database must be in migration.
- All queries must be using ORM (No direct queries allowed).

Nice to be included

- Use docker compose to run this example.
- Use scopes on ORM.
- Unit tests coverage.

The Evaluation:

TASK WILL BE EVALUATED BASED ON:

- Code quality.
- Application performance in reading large amount of records.
- Code scalability : ability to change (add/remove languages) must cover missing languages in view profile.
- No exceptions.