



Interview Assignment

Create a frontend web application that uses the Eleox Interview API (documented below).

URL: <https://eleox-interview-api-7n5su.ondigitalocean.app/>

It must:

- Build and run successfully (if a build process is involved in your frontend application)
- Successfully login when provided valid credentials:
 - username: "interview"
 - password: "eleox"
- Display a list of people and render the "avatar" of each person in the list, if it exists.
- Include a README file that explains your decision making process to the best of your ability, as well as any build/run instructions.
- Display comments for a person, on some kind of "detailed view". *Hint*: For comments: you must make an additional API call

Bonus Points:

- A clickable "delete" button on a person with a pop-up confirmation for deletion.
- A clickable "create" button that isn't a part of the people list. Clicking this button should bring up a way to create a person (I.E. a Create person modal)
- Form validation
- Error handling
- Protected routes

Time? The "it must" requirements can take anywhere from 2-3 hours to finish. We don't expect more than a few hours of work on this, as we value your time. Show us your best.

Above and beyond? Showing off? Bonus points. This will take longer than a few hours.

We will be reviewing your code together so we'll need one of the following:

- A zip file of the repo
- A private github repo

Please write readable, re-usable code. Commit often.

eleox Interview API Documentation

Authorization: Bearer Token

Please pass in the access token returned from the **POST /login** after a successful login.

```
{
  "headers": {
    "Authorization": "Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJkbWJyYXdlIiwiaWF0IjoxNjA5NDM1Njc1LCJleHAiOiI1NTYxMjA0ODYsImF2ZCI6Ind3dy51bWJyYXdlLmNvbSIsInN1YiI6IndhbHQva2lkbmV7QHVtYnJhZ2UuY29tIiwiaWZmlzc3RuYW1lIjoiaS2lkbmV5IiwiZW1haWwiOiJ3YWx0LmtpZG5leUB1bWJyYXdlLmNvbSIsImpvYlRpdGxliQ3JhZnQ6IFNvZnR3YXJlIEVuZ2luZWVyaW5nIn0.1h55DREei28Jx6LyWSkZ46cjBA5IsHJb3nT-7MIKsSA"
  }
}
```

- Note1: The access from from **POST /login** will expire in 24 hours.
- Note2: The above access token is an example and will not work.

POST /login - returns access_token on successful login

- **username** required, must be valid email and valid credential
- **password** required, must be valid credential

Example Payload (with valid credentials)

```
{
  "username": "int@elelox.com",
  "password": "eleox"
}
```

Example Response:

```
{
  "access_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJkbWJyYXdlIiwiaWF0IjoxNjA5NDM1Njc1LCJleHAiOiI1NTYxMjA0ODYsImF2ZCI6Ind3dy51bWJyYXdlLmNvbSIsInN1YiI6IndhbHQva2lkbmV7QHVtYnJhZ2UuY29tIiwiaWZmlzc3RuYW1lIjoiaS2lkbmV5IiwiZW1haWwiOiJ3YWx0LmtpZG5leUB1bWJyYXdlLmNvbSIsImpvYlRpdGxliQ3JhZnQ6IFNvZnR3YXJlIEVuZ2luZWVyaW5nIn0.1h55DREei28Jx6LyWSkZ46cjBA5IsHJb3nT-7MIKsSA"
}
```

GET /people - returns all people *Requires Authentication

Example Response:

```
{
  "people": [
    {
      "id": "cd7e01d3-5001-4657-99e7-1471e7bd9642",
      "first_name": "Johathan",
      "last_name": "Mraz",
      "email": "Johathan.Mraz@eleox.com",
      "job_title": "Craft: Identity",
      "avatar": "https://picsum.photos/120"
    },
    {
      "id": "6462d35f-47ba-41ea-a385-abe0d32f55c2",
      "first_name": "Myrtle",
      "last_name": "Lakin",
      "email": "Myrtle.Lakin@eleox.com",
      "job_title": "Craft: Functionality",
      "avatar": "https://picsum.photos/120"
    }
  ]
}
```

GET /people/:id - returns one person, with comments *Requires Authentication

Note: Comments are randomized. You should get back at least one.

Example Response:

```
{
  "person": {
    "id": "cd7e01d3-5001-4657-99e7-1471e7bd9642",
    "first_name": "Orval",
    "last_name": "Streich",
    "email": "Orval.Streich@eleox.com",
    "job_title": "Craft: Group",
    "avatar": "https://picsum.photos/120",
    "comments": [
      {
        "id": "2df4878a-ba0c-4e35-8b85-9d4205116f75",
        "comment": "Voluptatum qui occaecati voluptas debitis  
tempore consequatur."
      },
      {
        "id": "3ea66677-c7ce-4e4e-a0c2-a70735fa179d",
        "comment": "Voluptas nihil in eum necessitatibus corrupti  
accusamus."
      }
    ]
  }
}
```

POST /people - returns one person *Requires Authentication

The actual data written to the database will be changed to avoid saving any sensitive information. Validations will still be checked.

- `first_name` required
- `last_name` required
- `email` required, must be valid
- `job_title` required
- `avatar` is optional

Example Payload

```
{
  "first_name": "Walt",
  "last_name": "Womble",
  "email": "walt.womble@eleox.com",
  "job_title": "Craft: Software Engineering"
}
```

Example Response:

```
{
  "person": {
    "id": "cd7e01d3-5001-4657-99e7-1471e7bd9642",
    "first_name": "Orval",
    "last_name": "Streich",
    "email": "Orval.Streich@eleox.com",
    "job_title": "Craft: Group",
    "avatar": null,
  }
}
```

DELETE /people/:id - deletes one person, given an id *Requires Authentication

Please don't delete all the people. Thanks!

Example Response:

```
{
  "deletedPerson": "cd7e01d3-5001-4657-99e7-1471e7bd9642"
}
```

