Bandyer Backend selection home assignment

Premises

- **Back-end**: Use NodeJS (the version is not relevant) + any fancy plugins you find suitable for the task at hand.
- Use your Github or bitbucket or similar service to present your solution.

Assignment description

The goal of this project is to assess development skills suitable for a backend developer position. It tests knowledge and proficiency with software design, unit testing and web technologies like HTTP and REST.

The assignment is a small backend for a mobile app where users can use the mobile app to create *instants* uploading their photos. The mobile app uses a REST API to create *instants* from photo and upload them to the system. Each instant has also some other attributes. After a photo of an instant has been uploaded a job is queued by the REST API to resize the photo into a size more suitable for the mobile app.

The mobile app also has a feed that shows all instants. For every instant the app requests and displays also the associated attributes.

Task: Build the REST API and background photo resizing described above. Write unit tests for each component.

Requirements:

1. Design your API in a RESTful way and respond with JSON.
2. Make sure your code has tests. Write the code and design your system to be as realistic and production-ready as possible. Follow best-practices and focus on quality.
3. An instant should have the following attributes: name of the user, name of the photo, weight, length, latitude, longitude and a timestamp.

4. Add 3 endpoints:

    - **Create an instant**:

        - Request: attributes for an instant including the photo as an HTTP multipart/form-data upload.
        - Should parse input including file upload, save to database, enqueue background job and send response.
        - To keep things simple, no authentication needs to be performed. Instead the name of the user can be sent in the request as a string.

- **Get all photos**:
  - Should return all photos in the system ordered by date (newest first).
- **Get resized photo for a catch**
  - Background job to resize photo: all photos should be resized to be max 140x140px

Describe your solution in a README and how to run it.

Guidelines

Please commit often and with good commit messages. This will allow us to see how you've approached the problem. Don't worry about changing things around often.

Don’t hesitate to ask any questions if you’re uncertain about the task or anything else is unclear.

Some hints:

- For background processing you could use something like RabbitMQ (https://www.rabbitmq.com/) or rsmq (https://github.com/smrchy/rsmq).
- To resize images you can use sharp (https://github.com/lovell/sharp) or Jimp (https://github.com/oliver-moran/jimp).

REMEMBER --> git commit -m "Done!" && git push && leave the building!