

Project Overview

Web application lets users to upload and download photos/videos to their account. User should be able to register, login and manage their account and also can share photos/videos with other users.

Note: This is NOT a social network kind of application, users cannot view others profiles or search other users. They can only manage their own photos and videos and share them via email.

I will use “media” to refer to “Photos and Videos”

For Example:

- User “A” can signup/login to the app, upload photos/videos and manage them (edit, delete, upload etc.).
- User “A” can then select their photos and tap share, that will bring up a window with textbox to enter email address of user “B”.
- Simply enter the email of user “B” and press send.
- If user “B” exist then user will get the photos shared, else user “B” get email to join the app and see shared photos.

The actual project will be backend APIs in Nodejs. Also there will be a small test web project (preferably in Angular) that uses the APIs to show the functionality. This web project will not have access to database and should just call end points of API

Structure of APIs:

There will be 3 APIs involved:

1. **AuthAPI: Private** API that will expose endpoints for login/register related functionality to only third API (APIGateway, see 3rd point below)
2. **MediaAPI: Private** API that will expose endpoints for managing user media (photos and videos) and can only be accessible by third API (APIGateway, see 3rd point below)
3. **APIGateway: Public** API, that will have endpoints for running web app, this API will forward requests to the other two APIs above and return response to the web app

Important points to note:

1. Secure api covering all the edge cases (e.g. bad data passed), multiple attempts to login etc.
2. MySQL database to manage user data (NO Mongo DB)
3. Unit and integration tests to cover the code and all possible scenarios (both good and bad)

AuthAPI Endpoints

- /api/signup
- /api/verify
- /api/resetpassword (to request and confirm)
- /api/resentverification
- /api/forgotpassword
- /api/login

An example of test coverage for /api/signup

If signup form has fields for name, email, password and confirm password then the integration tests should cover:

- "name" not provided
- "name" is shorter than minimum length
- "name" is longer than maximum length
- "email" is not provided
- "email" is invalid, wrong format
- "email" is shorter than minimum length
- "email" is longer than maximum length
- "password" is not provided
- "password" is shorter than minimum length
- "password" is longer than maximum length
- "password" invalid criteria
- "confirmpassword" not provided
- "confirmpassword" did not match "password"
- valid data - success

MediaAPI Endpoints

- /api/upload/photo
- /api/upload/video
- /api/upload/photoandvideo (user should be able to upload 1 photo and 1 video together)
- /api/share/{photoids}
- /api/notifications (notifications of photos shared, that are not seen by user)
- /api/updateSeenNotifications

There would be other endpoints that may be needed as project progress. However, the basic functionality would be as described in Description section below

APIGateway Endpoints

These endpoints are the ones that would be visible to public and web app will consume them. This layer hides the above two APIs.

Generic endpoints

- /api/contact
- /api/terms (to return terms and conditions html)

Project Description

This project lets users to take or edit photos/videos (media) on their mobile and share with other users. There will be separate mobile app (already in progress) that will be using the above end points to upload and download media. The small web app will let users to manage the media uploaded or shared.

User Flow:

- User should be able to signup/login from web app (the same end points will be used in mobile apps later)
- User should be able to choose media from their computer and upload to server
- User can manage the media on web app (delete media, share with other users etc.)
- User picks (one or more media items) >> tap share >> enter email addresses of people to share with and send
 - o After pressing Send, the query goes to server and backend should check if the email addresses exist, if yes then those people will get a copy of the media shared. Else those email addresses will get emails to join the app and see their shared media
 - o Upon joining the app, users should automatically have shared media copied in their accounts

Note this project is first phase (or the bases) of other projects. That's why this needs to be modular and fully unit tested. Login API and Media API will be used for other projects as well.

Please let me know any questions or if anything that is not clear.

Thanks