

FINAL PRACTICE

Digitalization of Delivery Notes

Description:

Develop an API using NodeJS and Express to manage delivery notes (hours worked or materials used) between clients and providers. The API should allow the following operations through its endpoints (you can refer to <https://bilyd-rpmaya.koyeb.app/api-docs/>):

OnBoarding (already done in the intermediate practice)

- **Create a user account:** POST /api/user/register
NOTE: Upon account creation (or when logging in later), you will receive a JWT token from the backend.
- **Validate email through a received code:** PUT /api/user/validation
NOTE: Since this request requires a session token, it must be sent with an Authorization Bearer header.
- **Login into the application (returns a session token):** POST /api/user/login
- **Enter personal user data:** PATCH or PUT /api/user
- **Enter company data:** PATCH /api/user/company

Additional Endpoints:

- Change password
- Recover password
- Invite another user to your company
- Hard delete (physically remove) and soft delete (logically remove) a user
- Send emails with verification codes

Clients:

- **Create a client:** POST /api/client
- **Update a client:** PUT (or PATCH) /api/client
(You must associate it with the user who created it and ensure it's not already created for that user or company.)

- **List all clients (if any):** GET /api/client
- **Retrieve a specific client:** GET /api/client/{id}
(You will only show the client's data for the logged-in user or the user's company.)
- **Archive (soft delete) and delete (hard delete) a client**
- **List archived clients and restore a specific one** (mark it as not deleted)

Projects:

- **Create a project associated with a user and a client:** POST /api/project
- **Update a project:** PUT /api/project/{projectId}
(Ensure it hasn't been previously created for that user or company.)
- **List all projects:** GET /api/project
- **Retrieve a specific project:** GET /api/project/{id}
(You will only show the projects for the logged-in user or their company.)
- **Archive (soft delete) and delete (hard delete) a project**
- **List archived projects and restore a specific one** (mark it as not deleted)

Delivery Notes:

- **Create a delivery note (hours or materials) for a specific project:** POST /api/deliverynote
 - A delivery note can be simple (one entry for hours worked by a person or a material)
 - A delivery note can contain multiple persons/hours and materials
- **List all delivery notes:** GET /api/deliverynote
- **Retrieve a specific delivery note:** GET /api/deliverynote/{id}
When fetching, also include user, client, and project data (use "ref" in the model and "populate" in the controller).
- **Generate and download a delivery note in PDF format:** GET /api/deliverynote/pdf/{id}
 - Use the `pdfKit` library (this will be covered in class)
 - You can use any template, but it must contain user, client, project, and delivery note (hours or materials) data, and the signature if signed.
 - Only the owner user or their guests can download delivery notes.

- **Sign a delivery note** (upload the signature image to IPFS or another cloud service)
- **After signing, upload the generated PDF to the cloud**, and prefer downloading it from there if available.
- **Delete a delivery note** (only if it is not signed).

Requirements:

- Create all necessary models and schemas (MongoDB or MySQL).
- Create all required routes.
- Implement all necessary validators.
- Implement all controllers.
- Except for registration, login, and password recovery, all endpoints must require a JWT token (so only a user can modify their data or their company's data if admin).
- Handle 4XX client-side errors appropriately.
- Log all 5XX HTTP errors (server-side) in a Slack channel (ideally, there shouldn't be any if error handling is correct).
- Document each endpoint using Swagger.
- Perform testing with JEST for all your endpoints.
- Make appropriate commits (do not upload all the code at once).
- Send emails for validation processes.
- Upload a `.envexample` file explaining the environment variables used.
- Upload examples of `.http` files for testing different endpoints.