Serie 7

Content:

- Spring Todo-Backend and Vue.js client

(25) Spring Todo-Backend

Based upon the files given, extend the Todo-Backend *Spring Boot* project.

The goal is to attach multiple tags (e.g. 'Work', 'Social', 'Miscellaneous'...) to todos. There is a many-to-many relationship between todos and tags. You should be able to add tags and find todos linked to specific tag(s) directly from your REST API. A complete API definition for this project is available on SwaggerHub [2].

Most of the project is already implemented and you should add a few resources:

- *GET /tags/* Get the list of all tags
- *POST /tags/* Create a new tag
- *DELETE /tags/* Delete all tags
- *GET /tags/{id}* Get one tag
- *DELETE /tags/{id}* Get one tag
- *PATCH /tags/{id}* Update an existing tag
- *GET /tags/{id}/todos/* Get the list of todos associated with a tag

As an example, a compliant API is available on `http://swagger.thing.zone/`.

1. Add *TagController* functions to handle the above resources.
2. Your API should comply with our specs which are available online [1]. As an example, you can test a compliant API using `http://todospecs.thing.zone/?http://todo.thing.zone`.
3. Due to CORS issues, the tests performed with Chrome will not work. We therefore suggest you to use another browser such as Firefox for example.

(26) Vue.js todotag-frontend

Based upon the files given, extend the todotag-frontend *Vue.js* project. This exercice is optional.

The frontend communicates with the Todo-Backend REST API in order to interact with todos and tags. By default, the client points to `http://todo.thing.zone`. Ultimately, you should be able to change the url to your localhost, where your backend is running.

The web client consists in two "routes" corresponding to two different views: todos and tags. the *Todos* page is already fully implemented. The goal of this exercice is to finish the implementation of the *Tags* page in order to add, edit and remove tags from the backend.

The base structure of the files is already present, you need to add javascript code and complete Vue.js templates. All the code can be transformed from the *todos* route after you understand it.

In the *src/assets/js/service.js* file:

- add the functions *create*, *update* and create in the *tag* property of the service

In the *src/components/Tags.vue* file:

- Complete the form in order to allow the creation of new tags. A title is the only input of the form.
- Display the list of tags under the form.
- Create a snackbar in order to display information about tag operations.
- Create the *saved*, *deleted*, *saveNewTag* and *refresh* methods.
- In the *mounted* method, get all tags and sort them for display.
- Add all necessary data in the file and link them together.

In the *src/components/Tag.vue* file:

- Create the different elements in order to display a tag, save it and remove it.
- Create the *prepareTag*, *save* and *remove* methods.
- Add all necessary data in the file and link them together.

References

[1] Reference specs for todo-backend. http://todospecs.thing.zone/ (accessed Apr 13, 2020).

[2] Todo-tag backend api. http://petstore.swagger.io/?url=https://api.swaggerhub.com/apis/DurandA/todo-tag/0.0.3 (accessed Apr 13, 2020).

[3] Jacques Pasquier. Génie logiciel I, 2025. https://moodle.unifr.ch/course/view.php?id=284879 (accessed May 06, 2025).