Installation

Angular blog can be obtained from https://github.com/ea234/AngularBlog

git clone https://github.com/ea234/AngularBlog.git

After the git clone is done, the Angular components have to be installed. This is done with the command "npm install," which must be run within the AngularBlog directory.

```
ea234@MsiZ370:~$ cd angular
ea234@MsiZ370:~/angular$ git clone https://github.com/ea234/AngularBlog.git
Klone nach &apos;AngularBlog&apos; …
remote: Enumerating objects: 776, done.
remote: Counting objects: 100% (776/776), done.
remote: Compressing objects: 100% (358/358), done.
remote: Total 776 (delta 567), reused 611 (delta 402), pack-reused 0 (from 0)
Empfange Objekte: 100% (776/776), 1.08 MiB | 12.76 MiB/s, fertig.
Löse Unterschiede auf: 100% (567/567), fertig.
ea234@MsiZ370:~/angular$ cd AngularBlog/
ea234@MsiZ370:~/angular/AngularBlog$ npm install
npm warn deprecated inflight@1.0.6: This module is not supported, and leaks
memory. Do not use it. Check out lru-cache if you want a good and tested way to
coalesce async requests by a key value, which is much more comprehensive and
powerful.
npm warn deprecated rimraf@3.0.2: Rimraf versions prior to v4 are no longer
supported
npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer
supported

added 1003 packages, and audited 1004 packages in 7s

176 packages are looking for funding
  run `npm fund` for details

5 vulnerabilities (3 low, 2 high)

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
ea234@MsiZ370:~/angular/AngularBlog$
```

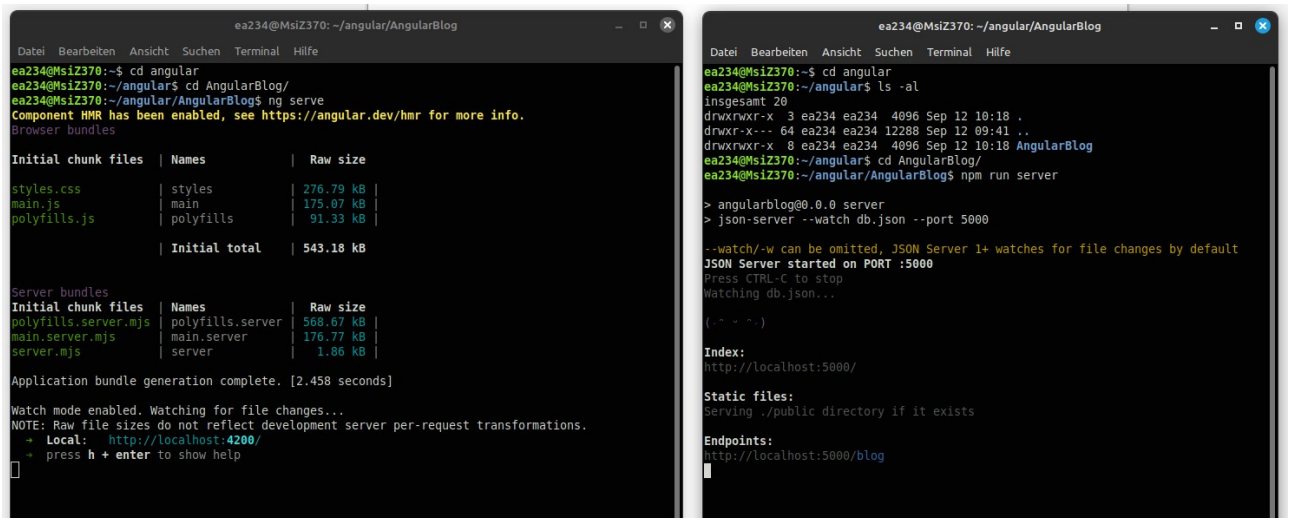
Application Start

To start the application you need 2 terminals.

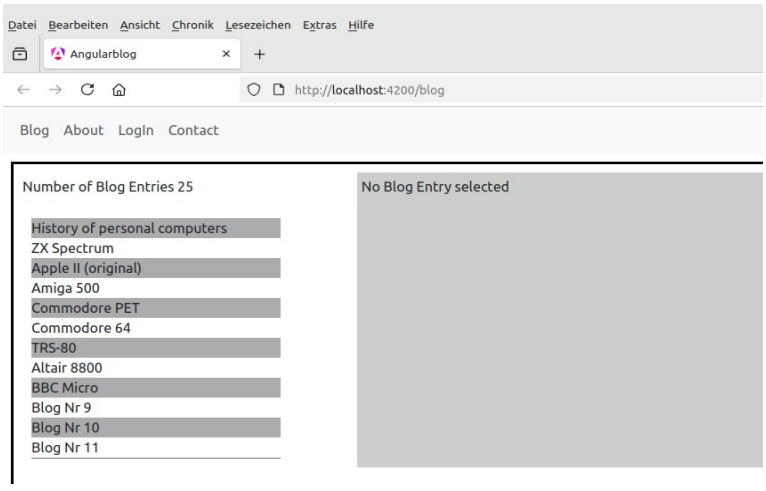
In one terminal you have to start the Json-Server, with „npm run server“

In the other terminal you have to start the Application with „ng serve“.

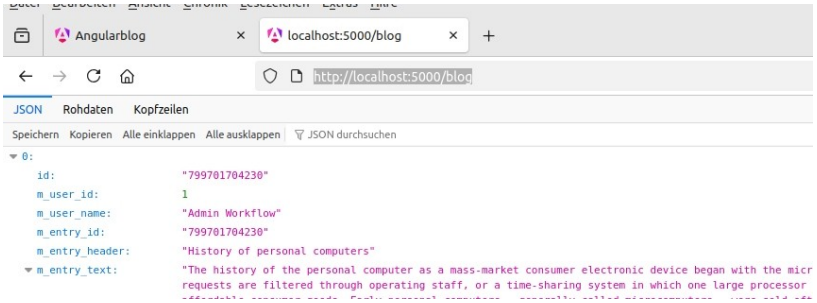
Both comands must be run within the application root directory.



The application is now accessable with **http://localhost:4200/blog**



The json-Server ist accessable with „http://localhost:5000/blog“



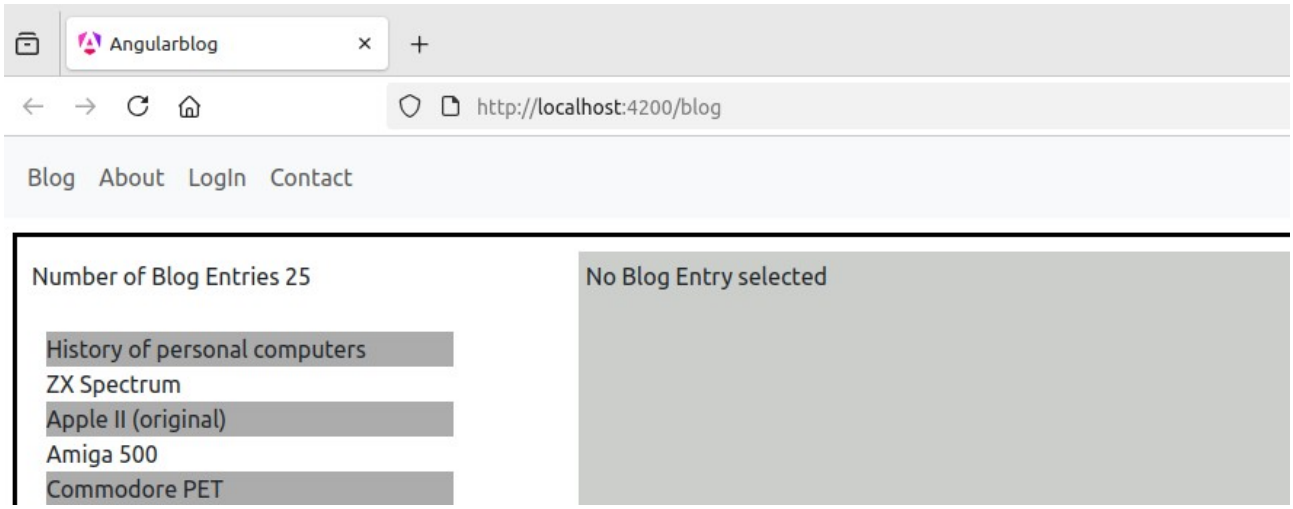
Startpage

The startpage shows a list of the blog-entries from the json server.

It also shows the number of blog-entries.

Initially there ist no blog-entry selected, to test the „BlogDetailEmptyComponent“.

The BlogDetailEmptyComponent is a placeholder, wich only displays a Message.



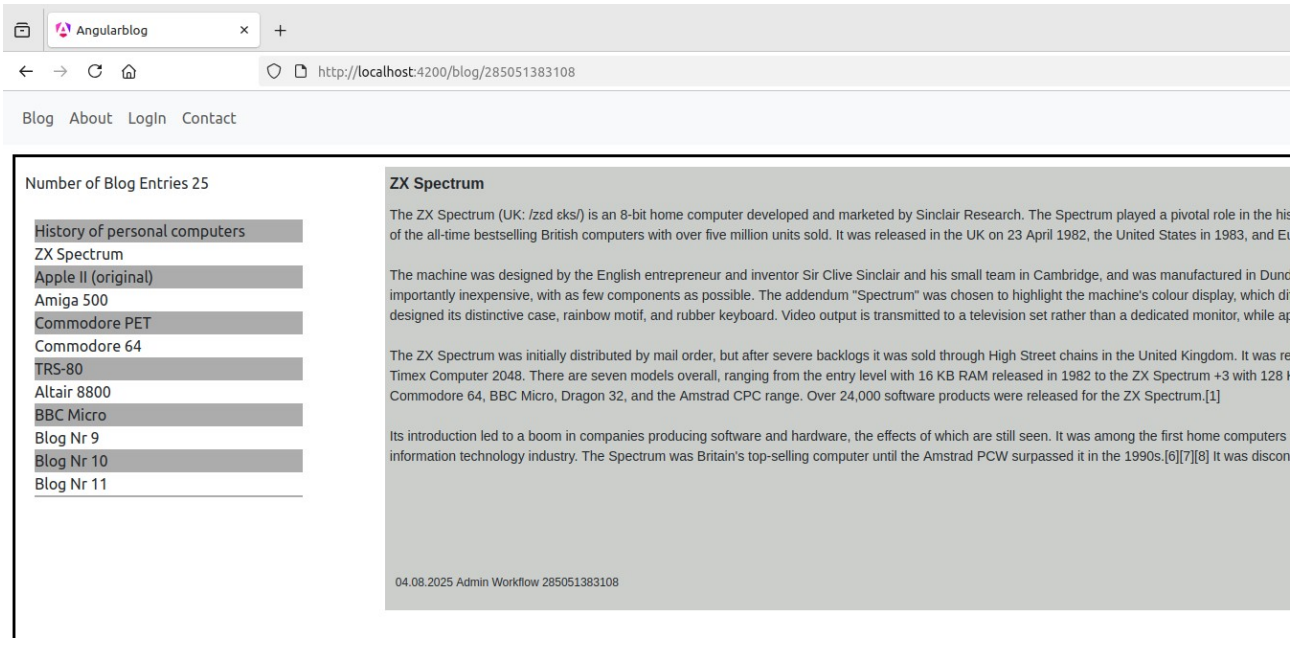
If an blog-entry is selected, it will be shown with the „BlogDetailLongComponent“.

This component is started with a route, which has the blog-entry-id as parameter.

This parameter is used to get the blog-entry from the server.

The Component uses the BlogJsonserverService, to communicate with the json-server.

If the blog-entry is not found, the component will show an alternate text.



Log-In and BlogUser

This test application uses 2 predefined users, which are defined in the Service „BlogUserService“.

One user with the status „logged out“, the user id 0 and the name „Guest“.

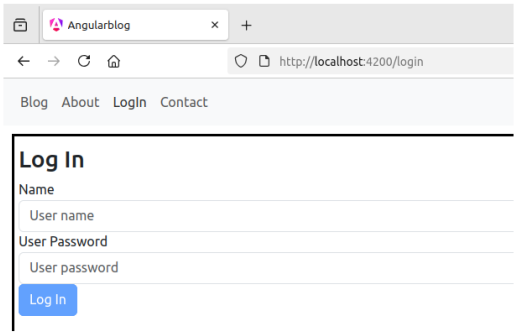
One user with the status „logged in“, the user id 1 and the name „admin“.

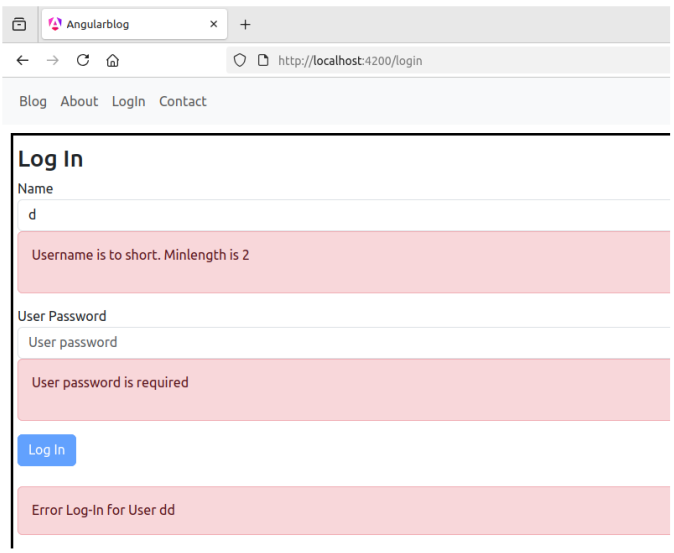
All other users will have the user id 2, with the name from the login-form.

The login-form consist out of the 2 inputs „user name“ and „user password“.

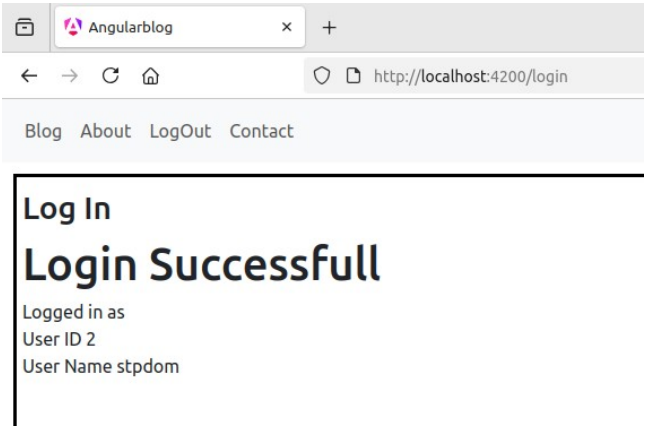
Since there isn’t a real user-database behind it, the password for the login is „login“.

Errors will be displayed.





If the login is successfull, the following message is displayed.



In the header component, there is now a „LogOut“ link displayed.

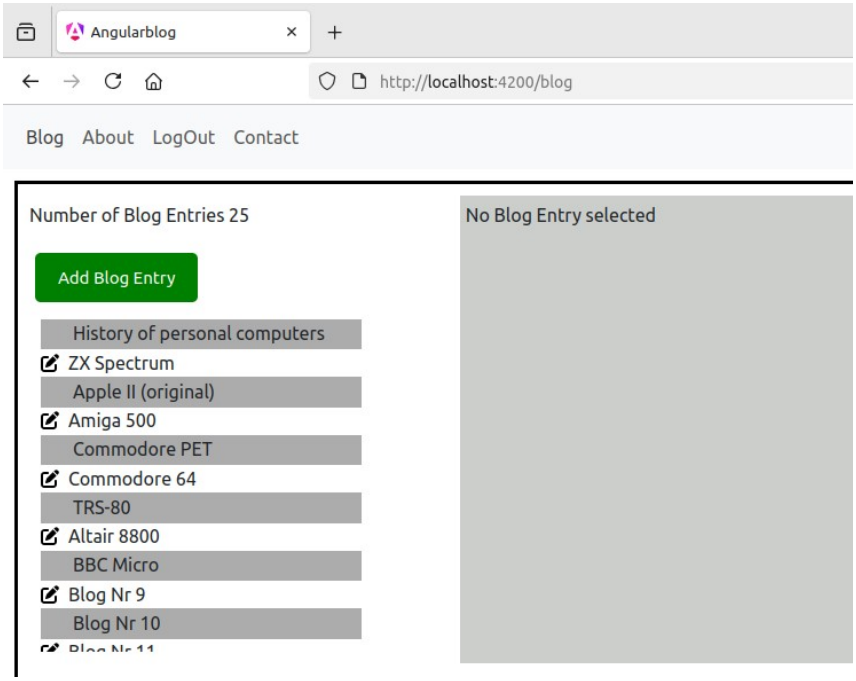
After a log in, you can navigate again to the blog section.

Blogview for logged in Users

If a user is logged in, a button for new blog entries is shown.

In the blog list, there is a link to edit the blog entry.

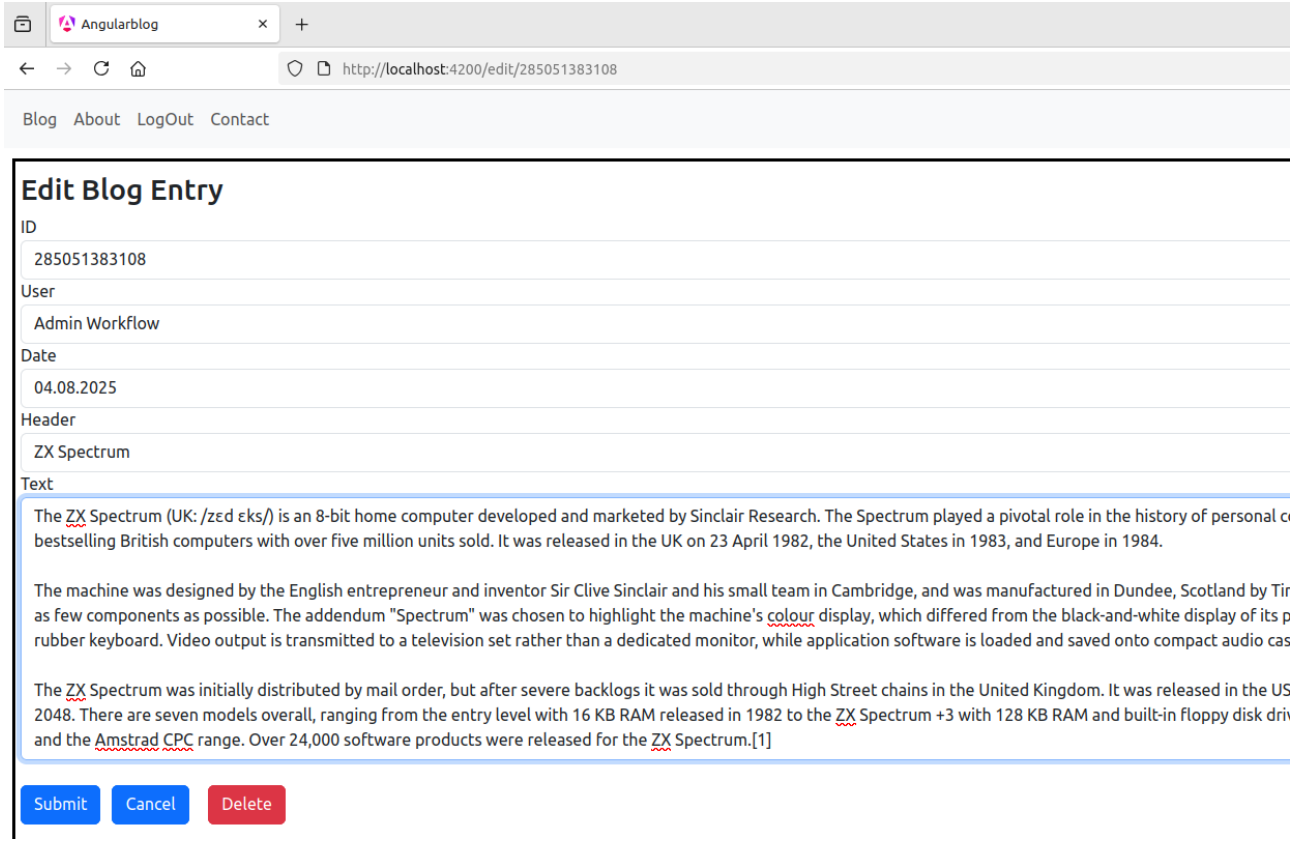
This link is only shown, when the entry-user-id matches with the logged in user-id.



Edit Blog Entry

If the blog-edit link is clicked, the edit form will be shown.

The component is „BlogEditFormComponent“



The routerlink contains the entry-id.

The blog entry is loaded from the database via the „BlogJsonserverService“.

If submit is clicked, the form will be saved via the „BlogJsonserverService“.

The user will be routed back to the blog-route.

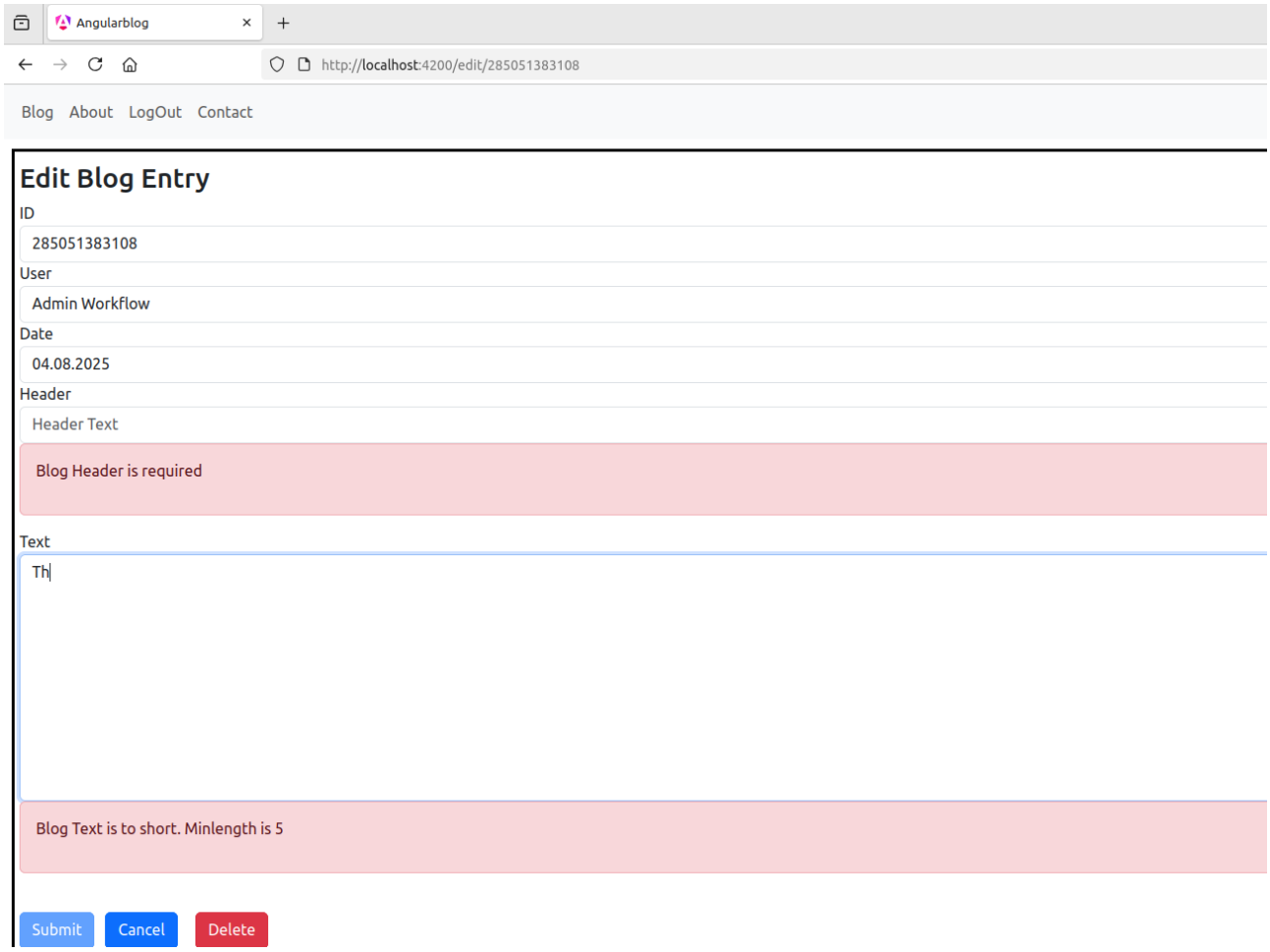
If cancel is clicked and the form is not „dirty“, the user will be routed back to the blog-route.

If cancel is clicked and the form is „dirty“ a confirmation-dialog will be shown.

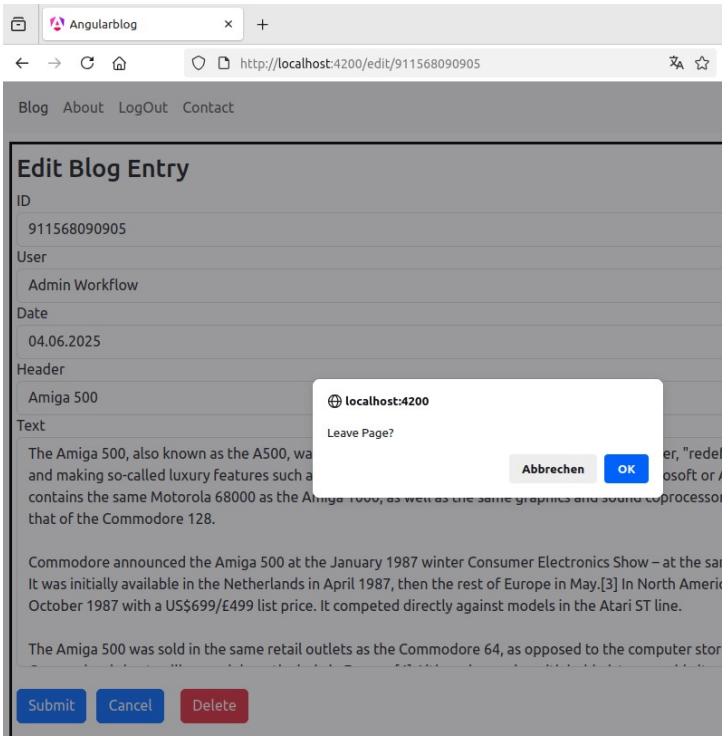
If delete is clicked, a confirmation-dialog will be shown. If the user confirms the deletion, the blog will be deleted via the „BlogJsonserverService“.

The Id, User and Date inputs can’t be changed. This is only displayed information.

Errors will be displayed.



There is a confirmation-guard, to prevent the user from activating another link.



Add new Blog Entry

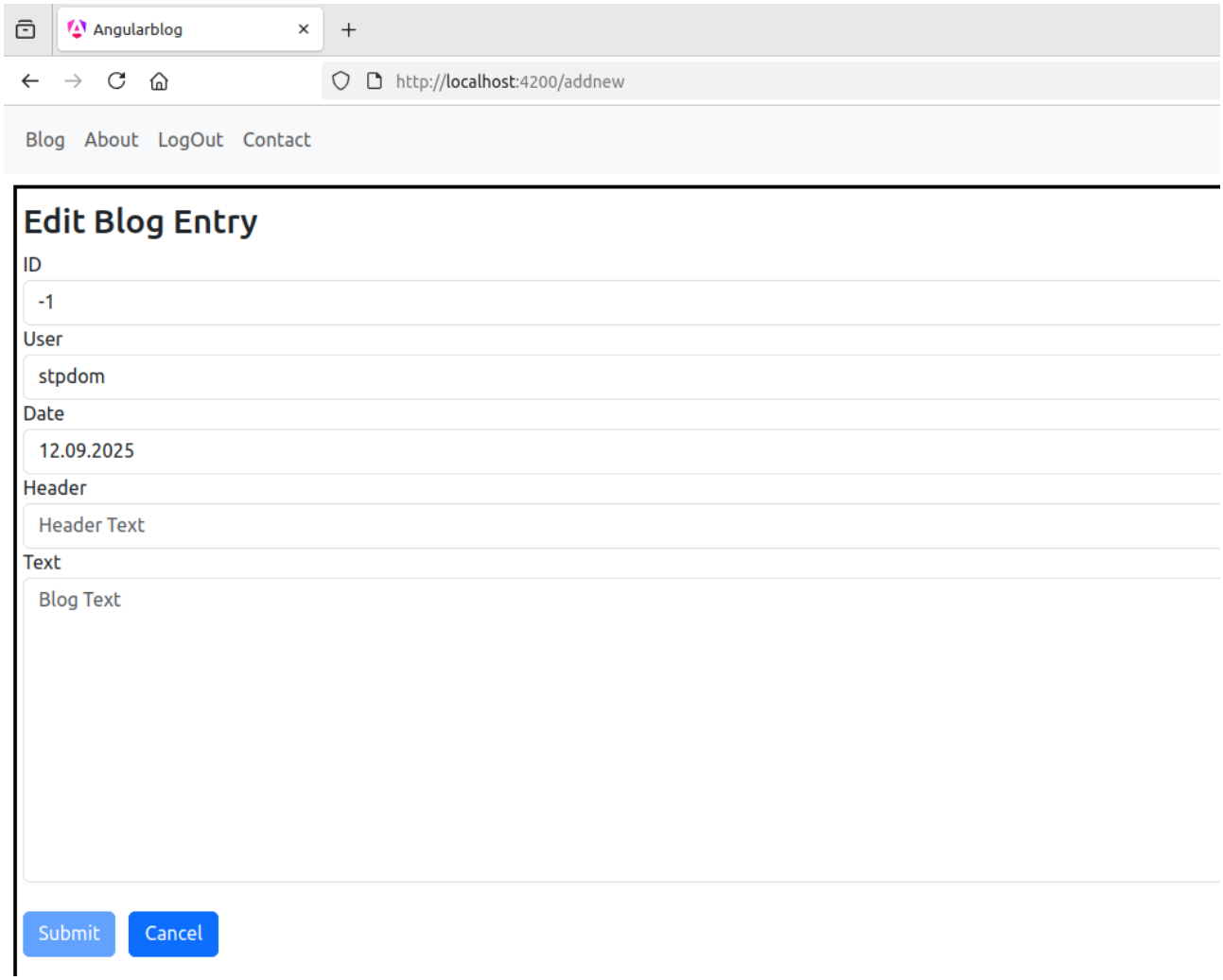
If the „Add Blog Entry“ button is clicked, the add form will be shown.

The component is also the „BlogEditFormComponent“.

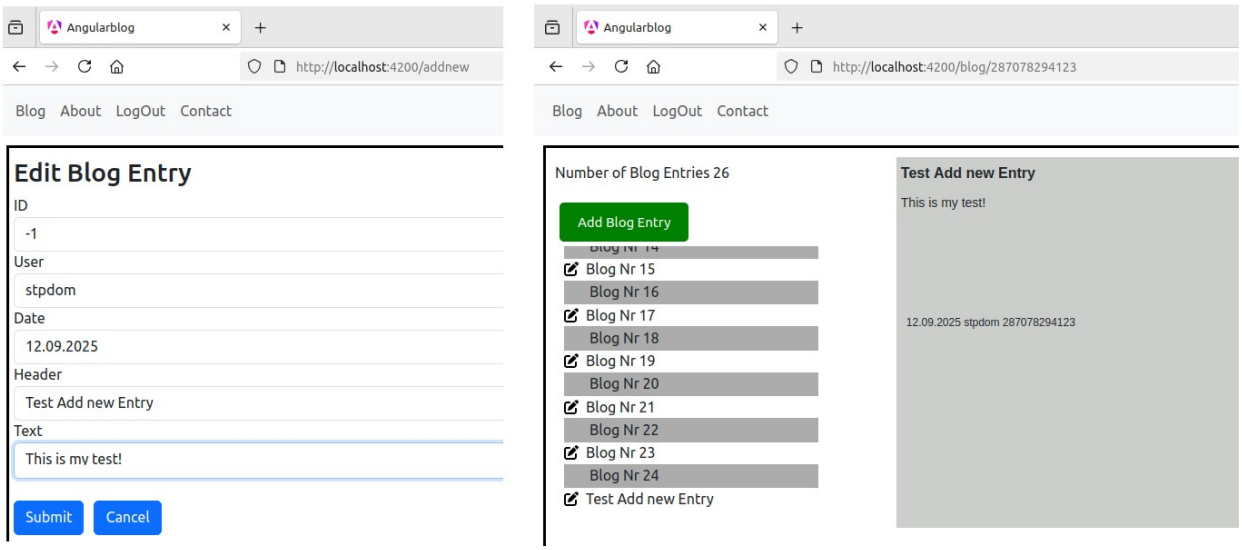
The routerlink contains now the word „addnew“. This „addnew“ is checked. If it is in the activated route, the edit-form becomes now the add-form.

The add-form has no „Delete“ button.

A new blog-entry has the id „-1“.

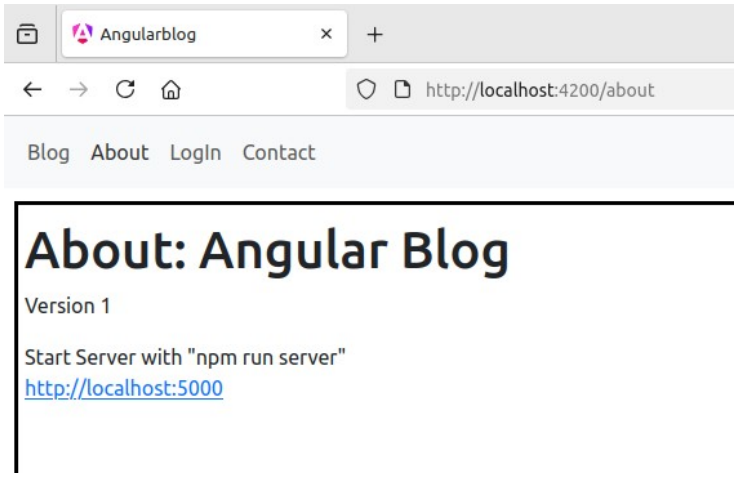


Test for a new Blog-Entry



About Form

The „AboutForm“ is handled by the Component „AboutComponent“.



Contact-Formular

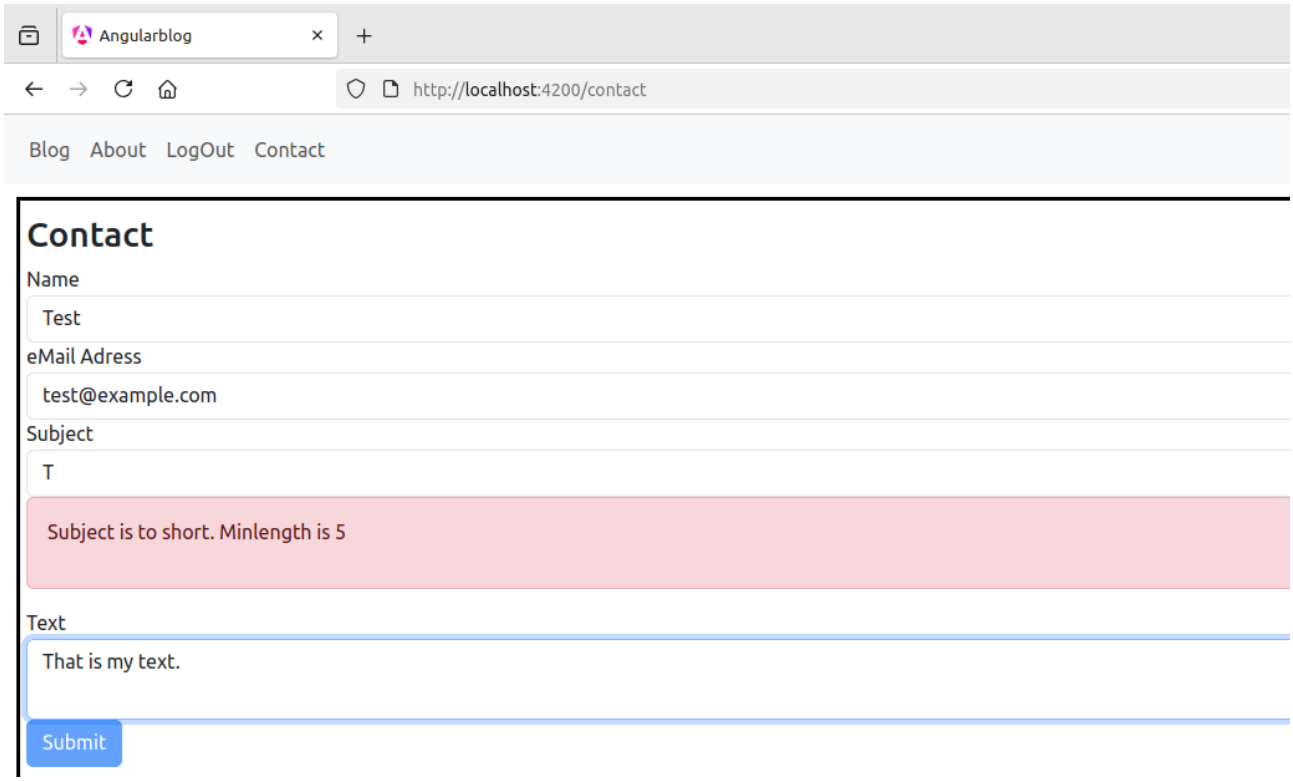
The contact-formular is handled by the component „ContactFormComponent“.

It uses a reactive form, with a custom validator for the email-adress input.

The submit-button is enabled, when the form is valid.

If a form was submitted, a confirmation message is displayed.

For now, the contact-form is not processed in the background. It was not a requirement, so i omited that functionality.



Classes

BlogAppMain

This class is used for mockup purposes in the tests.

This was the first class for the blog-entry vektor, before i switched to json-server.

Tests with Jasmin Karma

There are some tests, which can be startet with the command „ng test“.