

Inhaltsverzeichnis

Installation.....	1
Application Start.....	3
Startpage.....	4
Log-In and BlogUser.....	5
Blogview for logged in Users.....	6
Logout.....	7
Edit Blog Entry.....	8
Add new Blog Entry.....	10
About Form.....	11
Contact-Formular.....	12
Classes.....	13
Tests with Jasmin Karma.....	15

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 'AngularBlog'; ...
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.

```

ea234@MsiZ370: ~/angular/AngularBlog
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
ea234@MsiZ370:~$ cd angular
ea234@MsiZ370:~/angular$ cd AngularBlog/
ea234@MsiZ370:~/angular/AngularBlog$ ng serve
Component HMR has been enabled, see https://angular.dev/hmr for more info.
Browser bundles

Initial chunk files | Names | Raw size
styles.css | styles | 276.79 kB
main.js | main | 175.07 kB
polyfills.js | polyfills | 91.33 kB
Initial total | 543.18 kB

Server bundles
Initial chunk files | Names | Raw size
polyfills.server.mjs | polyfills.server | 568.67 kB
main.server.mjs | main.server | 176.77 kB
server.mjs | server | 1.86 kB
Application bundle generation complete. [2.458 seconds]

Watch mode enabled. Watching for file changes...
NOTE: Raw file sizes do not reflect development server per-request transformations.
+ Local: http://localhost:4200/
+ press h + enter to show help

ea234@MsiZ370:~/angular/AngularBlog$ npm run server
> angularblog@0.0.0 server
> json-server --watch db.json --port 5000

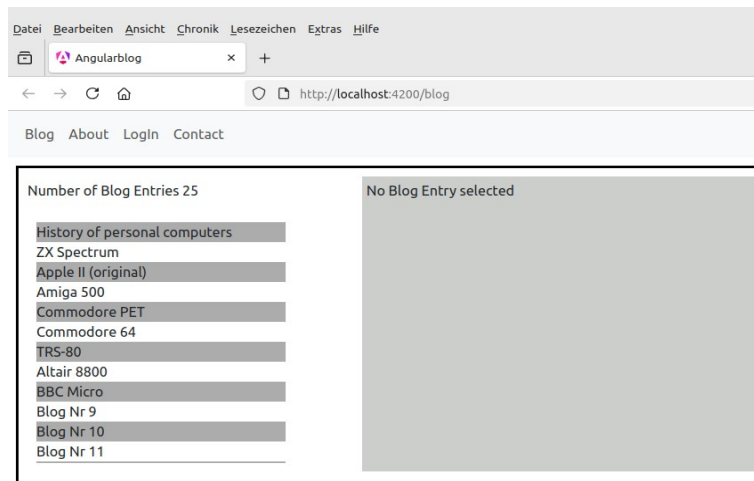
--watch/-w can be omitted, JSON Server 1+ watches for file changes by default
JSON Server started on PORT :5000
Press CTRL-C to stop
Watching db.json...

Index:
http://localhost:5000/

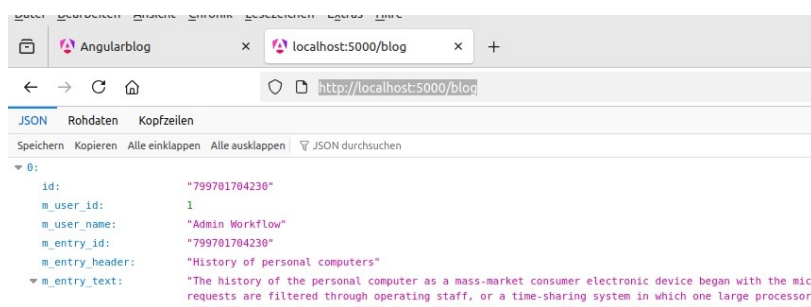
Static files:
Serving ./public directory if it exists

Endpoints:
http://localhost:5000/blog
  
```

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



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



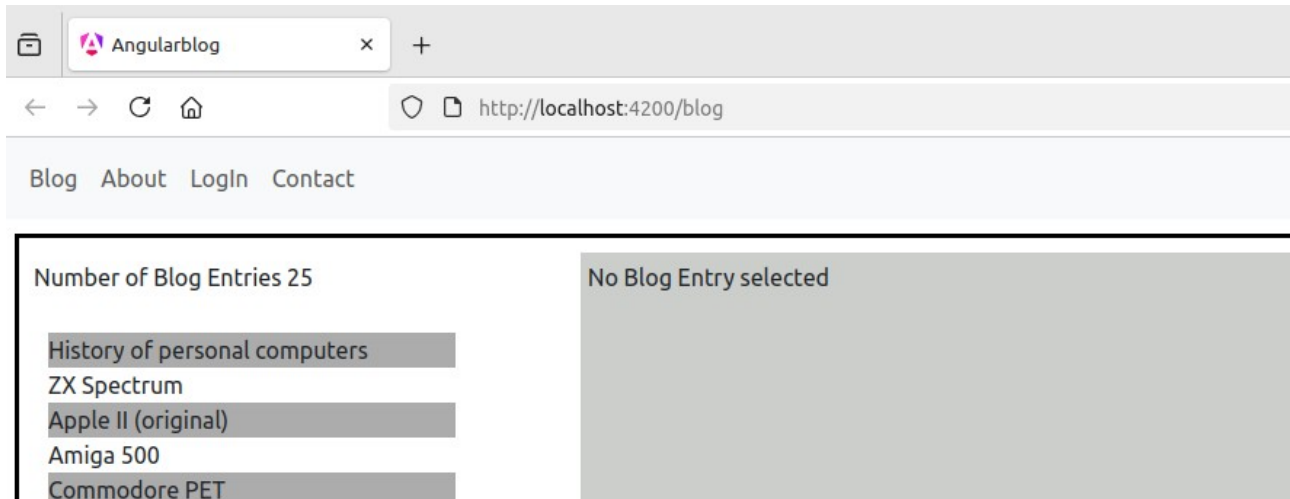
Startpage

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.



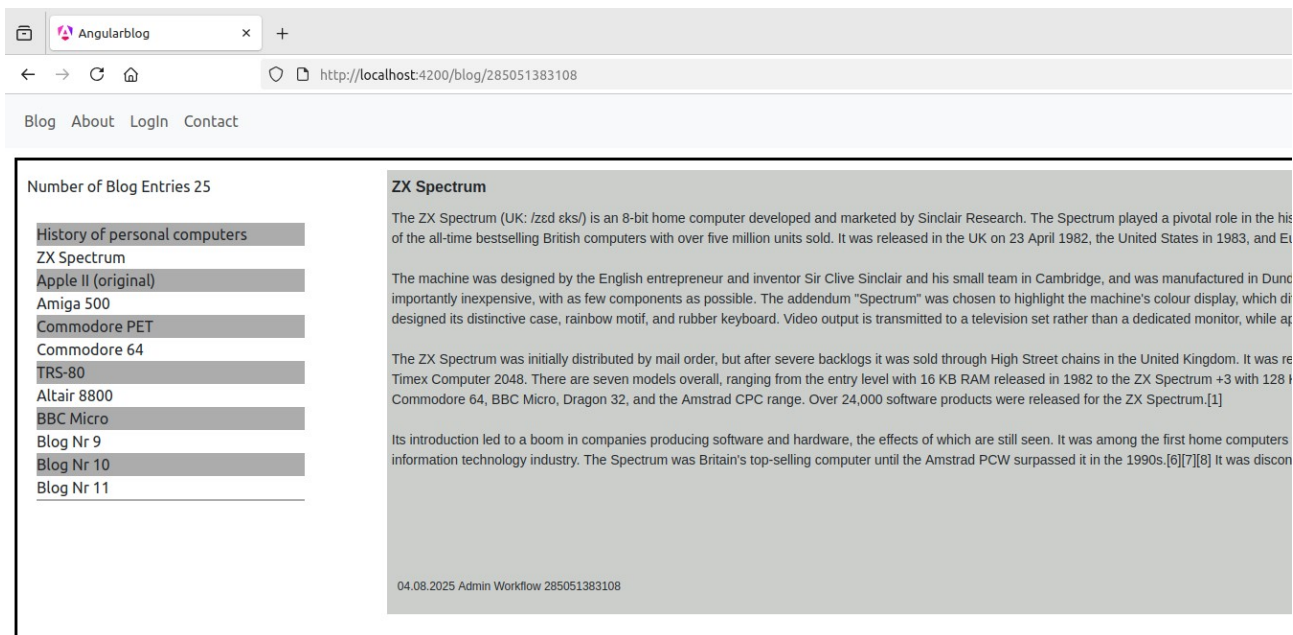
If an blog-entry is selected, it will be shown with the „BlogDetailLongComponent“.

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“.

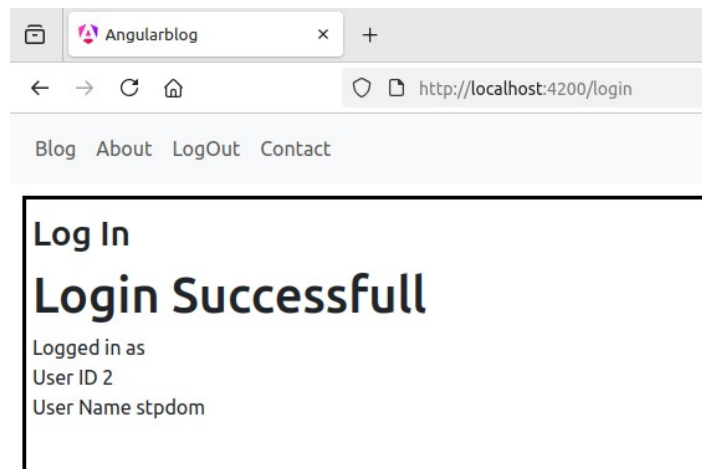
Since there isn't a real user-database behind it, the password for all users is „login“.

Errors will be displayed.

The screenshot shows the "Log In" form in the AngularBlog application. The form has two input fields: "Name" (with a placeholder "User name") and "User Password" (with a placeholder "User password"). There is a "Log In" button at the bottom of the form. The browser's address bar displays `http://localhost:4200/login`.

The screenshot shows the "Log In" form in the AngularBlog application with error messages. The "Name" field has a red error message: "Username is too short. Minlength is 2". The "User Password" field has a red error message: "User password is required". There is a "Log In" button at the bottom of the form. The browser's address bar displays `http://localhost:4200/login`. Below the form, there is a red error message: "Error Log-In for User dd".

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



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

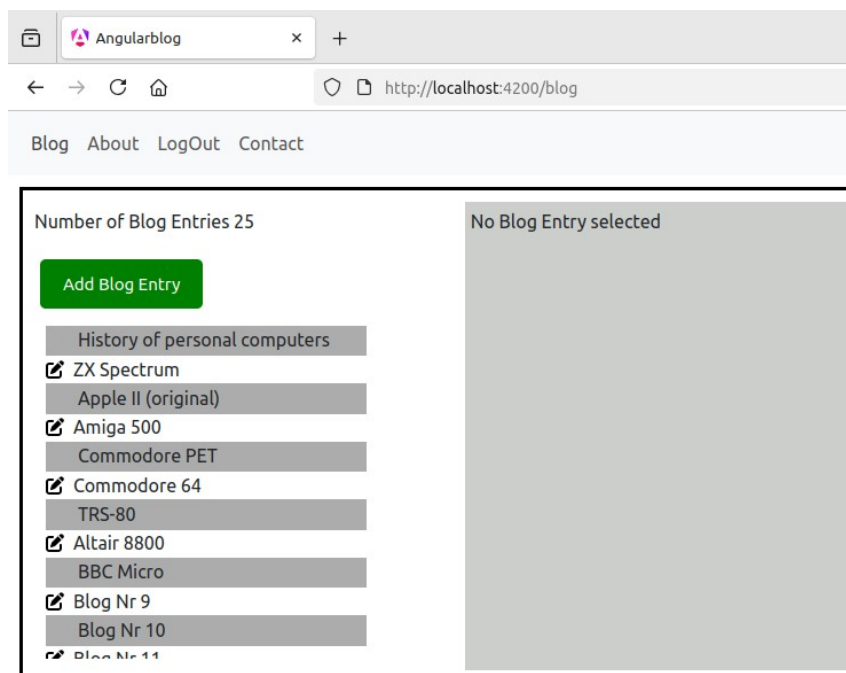
After a log in, you can navigate again to the blog section.

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.



Logout

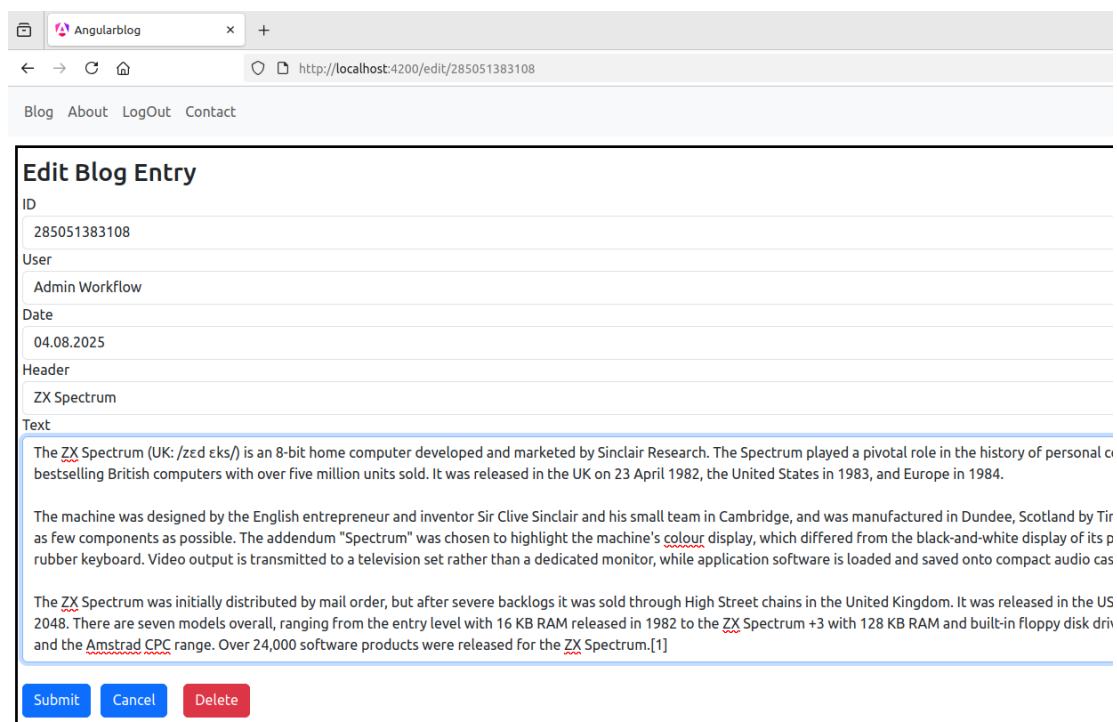
The Logout-Component logs the user out and redirects the user to the Route „./blog“.

The Logout-Component doesn't show anything.

Edit Blog Entry

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

The component is „BlogEditFormComponent“



Angularblog

← → ↺ ↻ http://localhost:4200/edit/285051383108

Blog About LogOut Contact

Edit Blog Entry

ID

285051383108

User

Admin Workflow

Date

04.08.2025

Header

ZX Spectrum

Text

The ZX Spectrum (UK: /zɛd ɛks/) is an 8-bit home computer developed and marketed by Sinclair Research. The Spectrum played a pivotal role in the history of personal computers, becoming the bestselling British computers with over five million units sold. It was released in the UK on 23 April 1982, the United States in 1983, and Europe in 1984.

The machine was designed by the English entrepreneur and inventor Sir Clive Sinclair and his small team in Cambridge, and was manufactured in Dundee, Scotland by TIR as few components as possible. The addendum "Spectrum" was chosen to highlight the machine's colour display, which differed from the black-and-white display of its predecessor. Video output is transmitted to a television set rather than a dedicated monitor, while application software is loaded and saved onto compact audio cassettes.

The ZX Spectrum was initially distributed by mail order, but after severe backlogs it was sold through High Street chains in the United Kingdom. It was released in the US in 1983. There are seven models overall, ranging from the entry level with 16 KB RAM released in 1982 to the ZX Spectrum +3 with 128 KB RAM and built-in floppy disk drive and the Amstrad CPC range. Over 24,000 software products were released for the ZX Spectrum.[1]

Submit Cancel Delete

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 display information.

Errors will be displayed.

Angularblog x +

← → ↺ 🏠 http://localhost:4200/edit/285051383108

Blog About LogOut Contact

Edit Blog Entry

ID
285051383108

User
Admin Workflow

Date
04.08.2025

Header
Header Text

Blog Header is required

Text
Th

Blog Text is to short. Minlength is 5

Submit Cancel Delete

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

Angularblog x +

← → ↺ 🏠 http://localhost:4200/edit/911568090905

Blog About LogOut Contact

Edit Blog Entry

ID
911568090905

User
Admin Workflow

Date
04.06.2025

Header
Amiga 500

Text
The Amiga 500, also known as the A500, was a home computer developed by Commodore International. It was designed as a more affordable and user-friendly version of the Amiga line, featuring a built-in floppy disk drive and a keyboard. It was marketed as a "complete" system, meaning it included everything needed to get started with computing, including software like a word processor and a database. The Amiga 500 was a significant success for Commodore, becoming one of the most popular home computers of the late 1980s and early 1990s. It was praised for its ease of use and its ability to handle a variety of tasks, from word processing to gaming. The Amiga 500 was also known for its colorful graphics and sound capabilities, which were considered impressive for its time. It was a true all-in-one computer, making it a popular choice for families and individuals looking for a simple and affordable way to enter the world of personal computing.

Commodore announced the Amiga 500 at the January 1987 winter Consumer Electronics Show – at the same time as the Commodore 64. It was initially available in the Netherlands in April 1987, then the rest of Europe in May.[3] In North America, it was released in October 1987 with a US\$699/£499 list price. It competed directly against models in the Atari ST line.

The Amiga 500 was sold in the same retail outlets as the Commodore 64, as opposed to the computer stores that carried the Amiga line.

Leave Page?

Abbrechen OK

Submit Cancel Delete

Add new Blog Entry

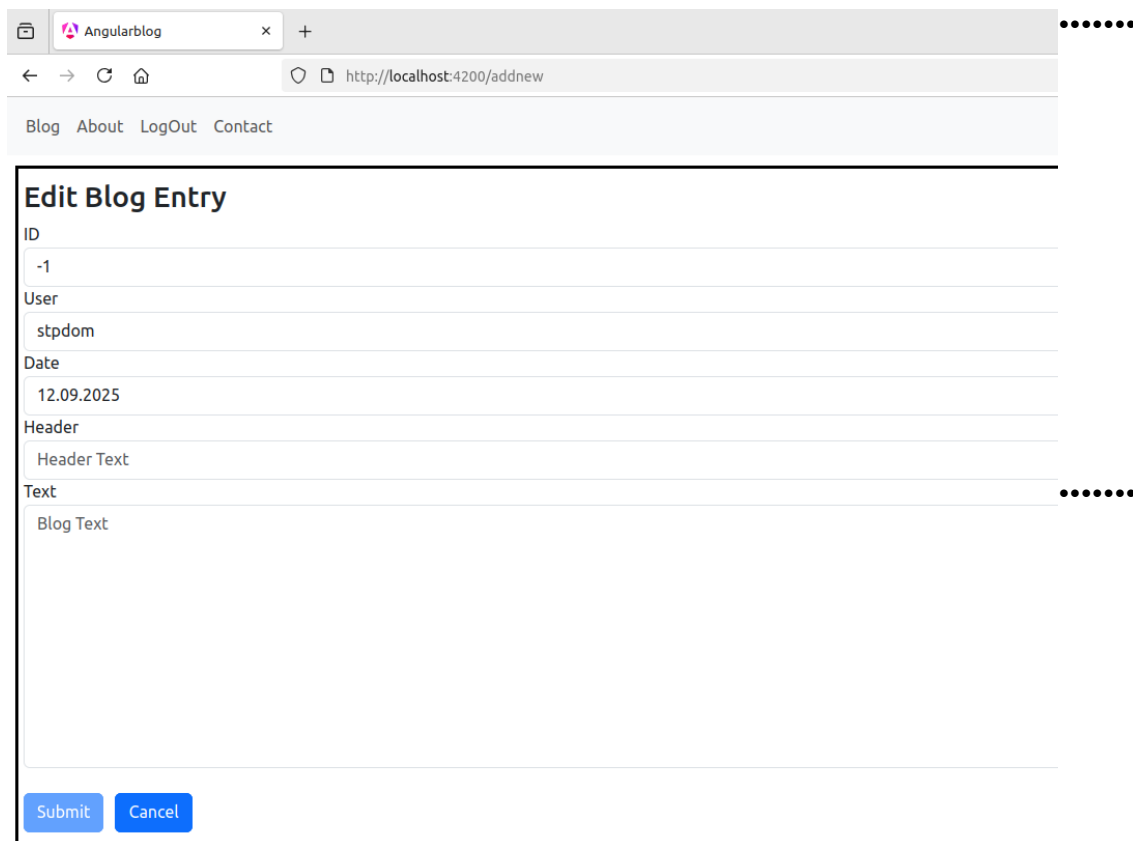
If the „Add Blog Entry“ button is clicked, the add form will be shown.

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“.

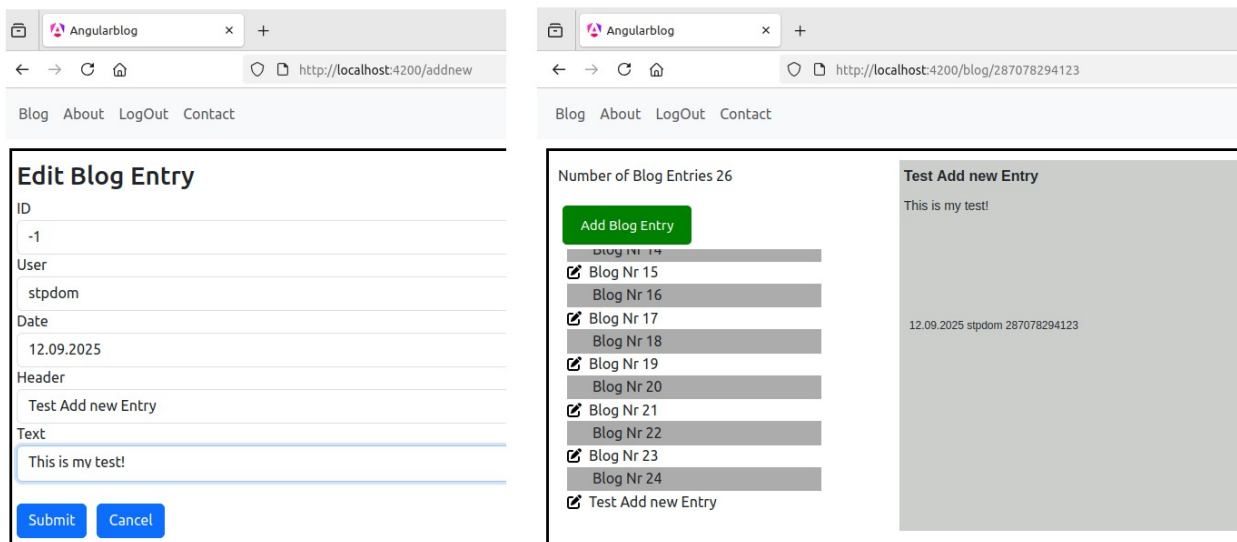


The screenshot shows a web browser window with the title "Angularblog" and the address bar displaying "http://localhost:4200/addnew". The navigation bar includes links for "Blog", "About", "LogOut", and "Contact". The main content area is titled "Edit Blog Entry" and contains a form with the following fields:

- ID: -1
- User: stpdom
- Date: 12.09.2025
- Header: Header Text
- Text: Blog Text

At the bottom of the form, there are two buttons: "Submit" and "Cancel".

Test for a new Blog-Entry



The left screenshot shows the 'Edit Blog Entry' form with the following fields:

- ID: -1
- User: stpdom
- Date: 12.09.2025
- Header: Test Add new Entry
- Text: This is my test!

The right screenshot shows the 'Test Add new Entry' form with the following fields:

- Number of Blog Entries: 26
- Add Blog Entry button
- Test Add new Entry button
- Test Add new Entry text area: This is my test!

About Form

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



The screenshot shows the 'About' page of the AngularBlog application. The page title is 'About: Angular Blog'. Below the title, it says 'Version 1'. The instructions are: 'Start Server with "npm run server"' and a link to 'http://localhost:5000'.

Contact-Formular

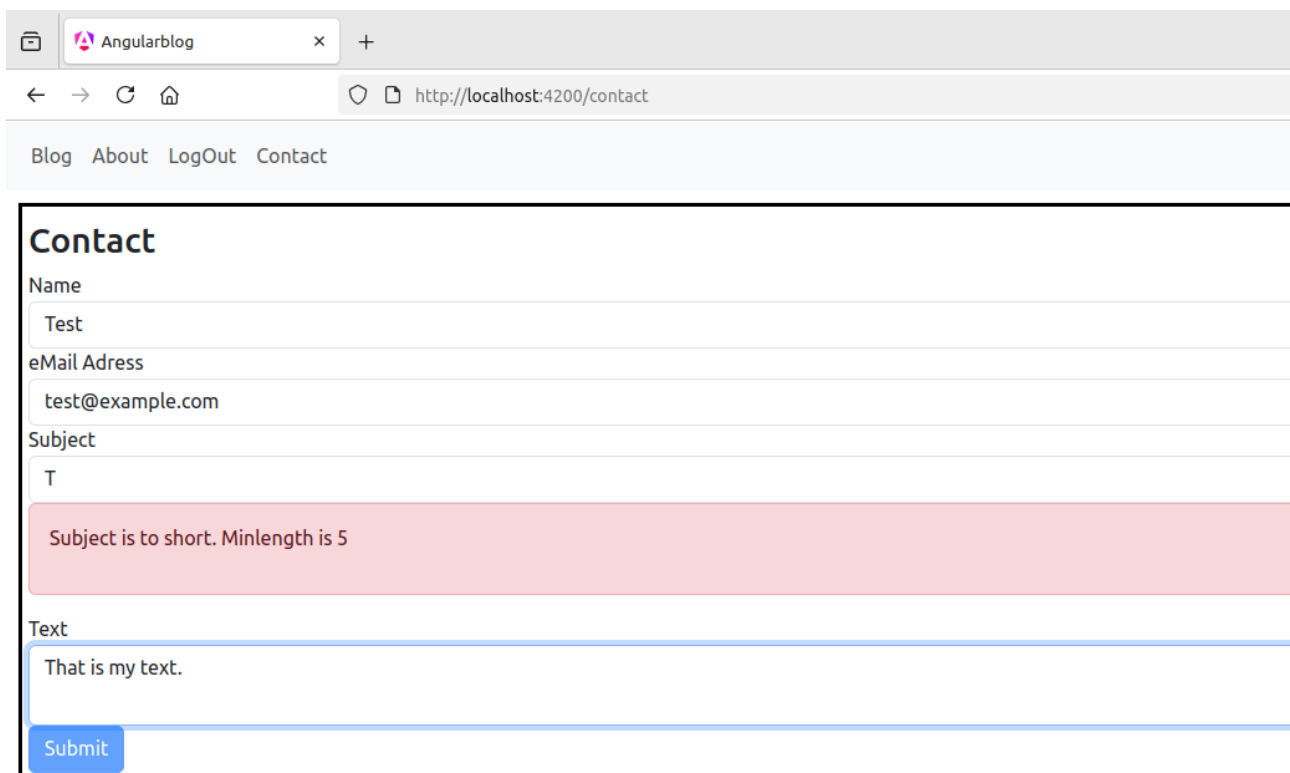
The contact-formular is handled by the component „ContactFormComponent“.

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 omitted that functionality.



The screenshot shows a web browser window with the address bar displaying `http://localhost:4200/contact`. The browser's tab is labeled "Angularblog". The page has a navigation bar with links: "Blog", "About", "LogOut", and "Contact". The main content area is titled "Contact" and contains a form with the following fields:

- Name:** A text input field containing the value "Test".
- eMail Address:** A text input field containing the value "test@example.com".
- Subject:** A text input field containing the value "T". Below this field, a red error message is displayed: "Subject is to short. Minlength is 5".
- Text:** A text area containing the value "That is my text."

At the bottom of the form is a blue "Submit" button.

Classes

Classname	Path	Comment
AppRoutingModule		Defines the routes of the app.
AboutComponent	/about	Shows the „about“ page
LogInComponent	/login	Shows the „Log-In“ page
LogOutComponent	/logout	Logs the user out and redirects the user to „/blog“.
ContactFormComponent	/contact	Shows the „Contact“ page.
BlogComponent	/blog	Shows the main page
BlogDetailLongComponent	/blog:blog_entry_id	Shows the blog-entry in the main page
BlogDetailEmptyComponent	/blog	Shows the „No Blog Entry“, when there is no blog-entry selected.
BlogEditFormComponent	/edit:blog_entry_id	Shows the Edit form for the given blog-entry.
BlogEditFormComponent	/addnew	Shows the edit form for a new blog-entry
HeaderComponent		Shows the App-Header
ButtonComponent		Shows a Click-Button
confirmationGuard		
ClsBlogBackend		Interface for the entry-services
BlogEntryService		MockUp-EntryService for tests
BlogJsonserverService		HTTP-Service for the Json-Server
BlogUserService		
ClsBlogEntry.ts		Interface and Class for a BlogEntry
ClsBlogUser.ts		Interface and Class for a BlogUser
FkDate.ts		Date functions
BlogAppMain		MockUp-Blog-Entry List for tests

Tests with Jasmin Karma

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

