

Phone Book

Time limit: 3.5 hours

The backend server provided to you maintains the list of contacts and provides a REST API for fetching, creating, updating and deleting contacts. A contact consists of a name and a (single) phone number.

First, start the backend

Install the packages in the backend first, and start it. The API endpoint is <http://localhost:4000/contacts> and it supports GET, POST, PUT, DELETE methods for CRUD operations.

Next, build the Angular frontend app

Create the Angular app to build a phone book, that lets the user manage a list of contacts, and serves as a frontend for the supplied backend application.

Feature 1: List contacts

When the app first loads, it shows the following form to add a contact (texts shown in inputs are placeholders). This is rendered by an **AddContact** component. Additionally, the initial set of contacts are fetched from the backend and shown. The list is rendered by a **ContactsList** component, the individual contact by **ContactListItem**.

Phone Book

Name

John Doe

Phone number

+001-1234567890

Add

John Doe
+001-1234567890

Jane Doe
+001-2345678901

Feature 2: Add contact

When the user enters contact details (name and phone number), and clicks the Add button, the contact is added. After a few contacts have been added it looks like this.

Phone Book

Name

John Doe

Phone number

+001-1234567890

Add

John Doe
+001-1234567890

Jane Doe
+001-2345678901

Mark Smith
+001-3456789012

Mary Smith
+001-4567890123

Tom Cat
+001-5678901234

Jerry Mouse
+001-6789012345

Feature 3, 4: Edit contact, Delete contact (buttons)

When hovering over contact (say, Jane Doe), the **Edit** and **Delete** buttons appear (if that is not possible, you may make these appear on every card even when not hovering).

Phone Book

Name: Phone number:

John Doe
+001-1234567890

Jane Doe
+001-2345678901

Mark Smith
+001-3456789012

Mary Smith
+001-4567890123

Tom Cat
+001-5678901234

Jerry Mouse
+001-6789012345

Feature 3: Edit contact

Phone Book

Name: Phone number:

John Doe
+001-1234567890

Mark Smith
+001-3456789012

Mary Smith
+001-4567890123

Tom Cat
+001-5678901234

Jerry Mouse
+001-6789012345

Figure 4.1 (Edit contact): Clicking the Edit button, opens up a form inside the clicked card, with details of the contact populated in the inputs.

Phone Book

Name: Phone number:

John Doe
+001-1234567890

Janette Doe
+001-2345678901

Mark Smith
+001-3456789012

Mary Smith
+001-4567890123

Tom Cat
+001-5678901234

Jerry Mouse
+001-6789012345

Figure 4.2 (Edit contact): Clicking Save updates the contact and shows the lists of contacts again (here Jane Doe has been updated to Janette Doe)

Hint for Editing feature: Maintain an **isEditing** boolean variable in the ContactListItem component. Set its value to true when the Edit button is clicked. Conditionally render either the contact details, or the form based on the isEditing variable's value. Send the edited name, phone number using an @Output event to ContactList when the Save button is clicked.

Feature 4: Delete contact

Phone Book

Name: Phone number:

John Doe
+001-1234567890

Jane Doe
+001-2345678901

Mark Smith
+001-3456789012

Mary Smith
+001-4567890123

Tom Cat
+001-5678901234

Jerry Mouse
+001-6789012345

Figure 6.1 (Delete contact): Clicking the Delete button deletes the contact (screenshot is from before deleting Mark Smith)

Phone Book

Name: Phone number:

John Doe
+001-1234567890

Jane Doe
+001-2345678901

Mary Smith
+001-4567890123

Tom Cat
+001-5678901234

Jerry Mouse
+001-6789012345

Figure 6.2 (Delete contact): Clicking the Delete button deletes the contact (screenshot is from after deleting Mark Smith)