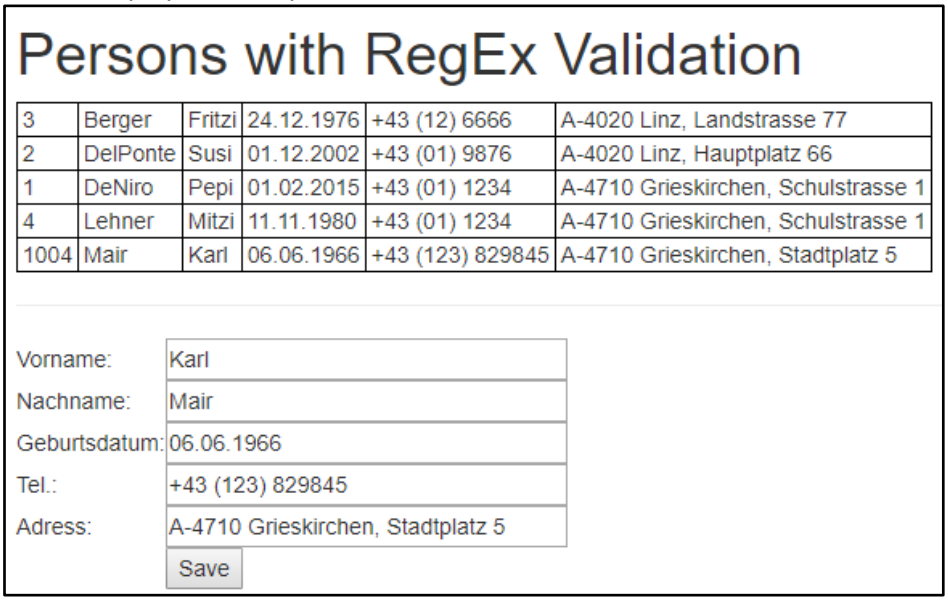
Personenformular mit RegEx

Write a small web site to display and edit persons.



1 Backend

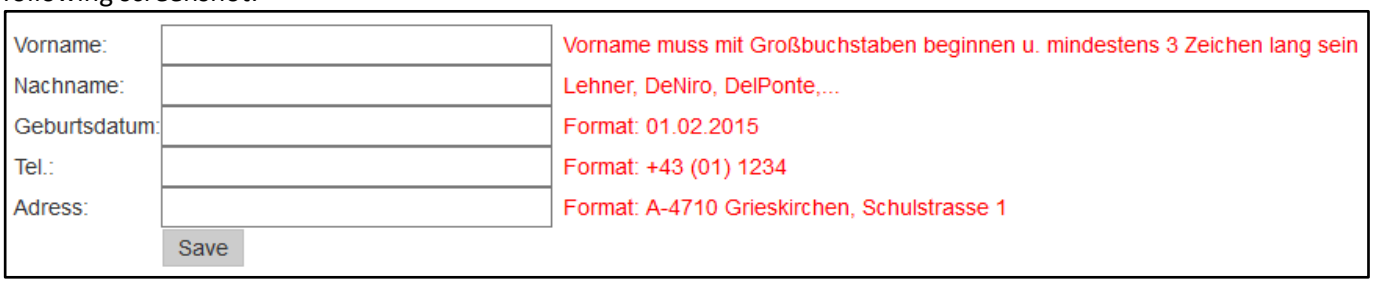
1.1 controller

Attached you find a running project with access to the database file **`Persons.sqlite`**.
Use the structure with a separate service class that holds the business logic.
Write methods to

- get all persons
- get a single person
- insert a new person

1.2 DTOs/Regular Expressions

Write the class PersonDto that holds all information required (see the above screenshot).
Decorate the properties of this class with [RegularExpression(…)] attributes to validate the data as shown in the following screenshot:



Note:

- the Birthdate is of DateTime in the database but should be string (with validation) in the DTO
- the Adress is sent as one string, so the RegEx has to contain groups so that the various parts can be extracted easily (see below when saving a person).

1.3 Database

Also decorate the Entity class Person to check the same regular expressions.
Override the SaveChanges() method as shown in the tutorial.

2 Frontend

The frontend shall look like the above screenshot and consist of two components.

2.1 Service

All data from the backend shall be fetched with a dedicated service.

2.2 person-list component

This component uses the above service to simply show all persons in a table.

2.3 person-edit component

This component is used to enter a new person.
For each input field it has to be validated with a regular expression pattern, if the value is correct (take care to check the whole input and not only parts of it). If the input is not correct on the right hand side an error message is displayed.

2.4 save person

If all inputs are valid, the save button is enabled. Clicking this button sends the person data to the backend using the above service. The backend then saves all data using RegEx (with Groups) to extract the data. The reply shall be displayed as a or an alert.

2.5. Notify service

Create another service that is used to notify a component about changes in the person repository.
After sending the person to the backend the person-edit component shall initiate a notification to which the person-list component listens.
Use the following code for the service:

```
export class NotifierService {
  subject = new Subject<number>();

  listen(): Observable<number> {
    return this.subject.asObservable();
  }

  notify(id: number): void {
    console.log(`NotifierService::notify ${id}`);
    this.subject.next(id);
  }
}
```

3 Extension

As the same regular expression patterns are required on the frontend as well as on the backend try to avoid duplicating this code (DRY).
One option is to write a controller action on the backend that sends the patterns and the error messages to the frontend.