

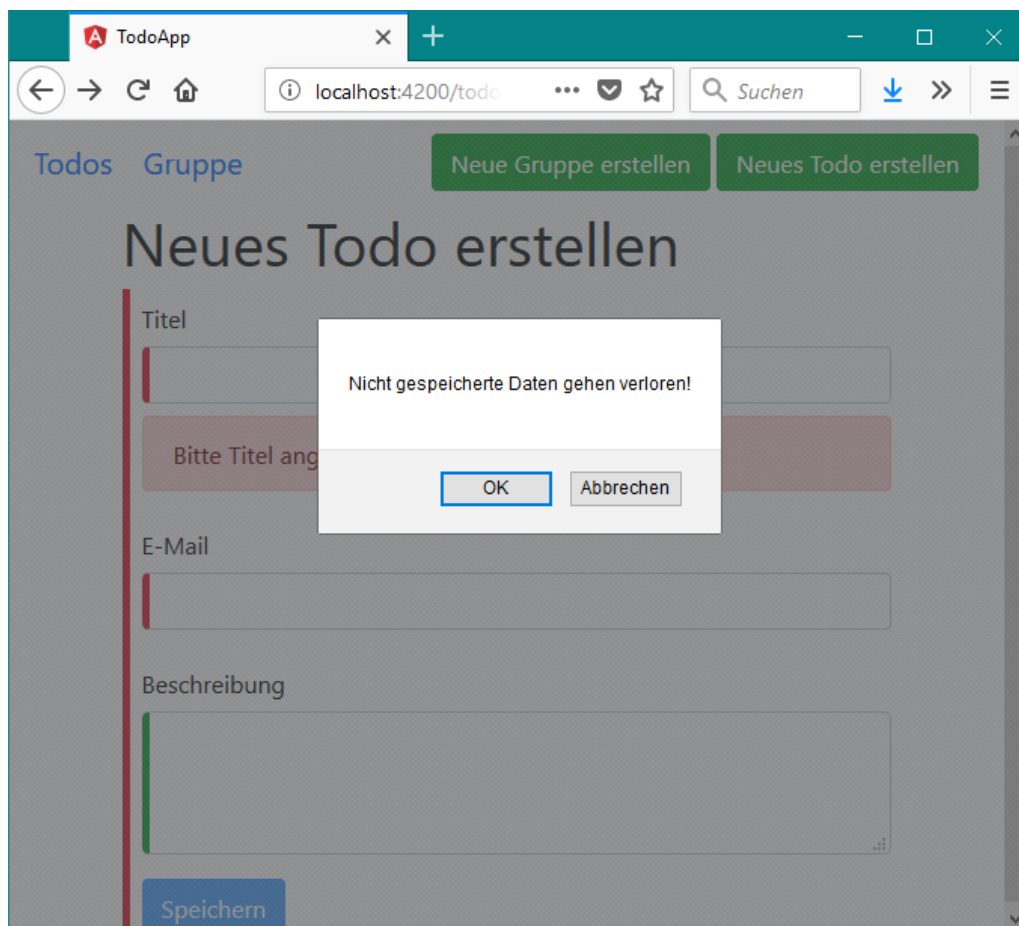
1. Resolve und Guards [U-130]

Als Nutzer möchte ich auf ungespeicherte Änderungen hingewiesen werden.

1. Lernziele

- ✓ Sie verwenden Resolver um Daten zu laden, bevor die Route aktiviert ist.
- ✓ Sie verwenden den Guard **CanDeactivate**, um unabsichtliches Verlassen der Seite durch den User zu verhindern.

2. Ergebnis



3. Benötigte Daten

- `src/app/app-routing.module.ts`
- `src/app/todo-create.component.ts`
- `src/app/todo-detail.component.ts`
- `src/app/todo-create-saved.guard.ts`
- `src/app/todo-detail.resolver.ts`

4. Anleitung

RESOLVER FÜR TODO DETAIL

Schritt 1: Erstellen der Service-Klasse **TodoDetailResolver**, welche **Resolve<TodoItem>** implementiert

1. Erstellen Sie in der Datei `/src/app/resolvers/todo-detail.resolver.ts` die Klasse **TodoDetailResolver**, welche das Interface **Resolve<TodoItem>** implementiert.
2. Definieren Sie in der Klasse **TodoDetailResolver** die öffentlich zugängliche Methode **resolve**
3. Übergeben Sie der Methode **resolve** den Parameter **route** vom Typ **ActivatedRouteSnapshot**
4. Definieren Sie einen Konstruktor, welcher mittels Dependency Injection eine Instanz des **TodoService** injiziert bekommt.
5. Fügen Sie ein **@Injectable()** vor der Klassendefinition an, welcher Sie ein Object übergeben mit der Property **providedIn** und dem Wert **"root"**

Schritt 2: Daten des Todos anhand der ID des angeklickten Eintrags zurückgeben.

6. Ermitteln Sie in der **resolve** Funktion anhand der Methode **route.paramMap.get** den Wert von **id**. Von welchem Typ ist der zurückgegebene Wert?

7. Konvertieren Sie den soeben erhaltenen Wert in eine Zahl entweder mit **Number** oder **parseInt()** und weisen den Wert der Konstanten **id** zu.
8. Rufen Sie die Methode **getById** des **TodoService** mit der Konstanten **id** als Parameter auf. Welchen Datentyp erwarten Sie als Rückgabewert?
9. Geben Sie den Rückgabewert des **getById** Aufrufs zurück und spezifizieren Sie den Rückgabebetyp der Funktion **resolve**.
10. Lassen Sie die Anwendung laufen und überzeugen Sie sich davon, dass Ihre Anwendung fehlerfrei ausgeführt wird.

Schritt 3: **TodoDetailResolver** in **AppRoutingModule** eintragen und als Resolver der Route **/todos/:id** festlegen

11. Ergänzen Sie in **app-routing.module.ts** das Objekt, das in der **TodoDetailComponent** als **component** angegeben ist, um das Attribut **resolve**
12. Weisen Sie diesem Attribut ein Objekt mit dem Attribut **todo** und dem Wert **TodoDetailResolver** zu.

Schritt 4: Methoden-Körper der **ngOnInit** in **TodoDetailComponent** löschen und durch **route.snapshot.data.todo** mit Wertzuweisung von **todo** ersetzen

13. Entfernen Sie die Implementierung der **ngOnInit**-Methode der **TodoDetailComponent**
14. Weisen Sie der Objektvariablen **todo** den Wert des Attributs **snapshot.data.todo** der Objektvariablen **route** zu.
15. Entfernen Sie das Injizieren des **TodoService** in die Komponente.
16. Starten Sie die Anwendung im Browser und stellen sicher, dass Ihre Anwendung läuft. Welche Unterschiede im Verhalten der Anwendung stellen Sie fest?

GUARD IN TODO-FORM

Schritt 5: Erstellen der Service-Klasse **TodoCreateSavedGuard**, welche **CanDeactivate<TodoCreateComponent>** implementiert. Einbinden der Klasse in **AppRoutingModule** für Nutzung in **TodoCreateComponent**.

17. Erstellen Sie die Klasse **TodoCreateSavedGuard** in der Datei **todo-create-saved.guard.ts**, welche das Interface **CanDeactivate<TodoCreateComponent>** implementiert.
18. Definieren Sie die Methode **canDeactivate** mit dem Parameter **component** vom Typ **TodoCreateComponent** und einem Rückgabewert vom Typ **boolean**.
19. Fügen Sie der Klasse einen **@Injectable** Dekorator hinzu, welchem Sie ein Objekt mit **providedIn: "root"** übergeben.

Schritt 6: Erstellen Sie eine Methode **confirmAbort()** in **TodoCreateComponent**, welche den Rückgabewert von **window.confirm** zurückgibt

20. Definieren Sie in der Klasse **TodoCreateComponent** die Methode **confirmAbort()**, welche einen Boolean zurückgeben wird.
21. Rufen Sie in dieser Methode **window.confirm** mit einer Nachricht ihrer Wahl (z.B. „Nicht gespeicherte Daten gehen verloren!“) auf
22. Geben Sie den Rückgabewert von **window.confirm** als Rückgabewert der Methode zurück.

Schritt 7: **confirmAbort()** in **canDeactivate** aufrufen und Rückgabewert zurückgeben

23. Rufen Sie in der Methode **canDeactivate** der Klasse **TodoCreateSavedGuard** die Methode **confirmAbort** von **TodoCreateComponent** auf
24. Geben Sie den Rückgabewert von **confirmAbort()** aus **canDeactivate()** zurück

Schritt 8: `TodoCreateSavedGuard` mit `canDeactivate` der Route `/todo/new` zuweisen.

25. Ergänzen Sie in `app-routing.module.ts` das Objekt um die Eigenschaft `path: 'new'` der `TodoCreateComponent` mit dem Attribut `canDeactivate`

26. Weisen Sie diesem Attribut ein Array mit einem Wert `TodoCreateSavedGuard` zu

5. Kontrollfragen

- Zu welchem Zweck können Resolver und Guards verwendet werden?
- Wie weisen Sie einer Route einen Resolver zu?
- Welche Interfaces ermöglichen die Verwendung von Guards?