

## 1. Lazy Loading [U-120]

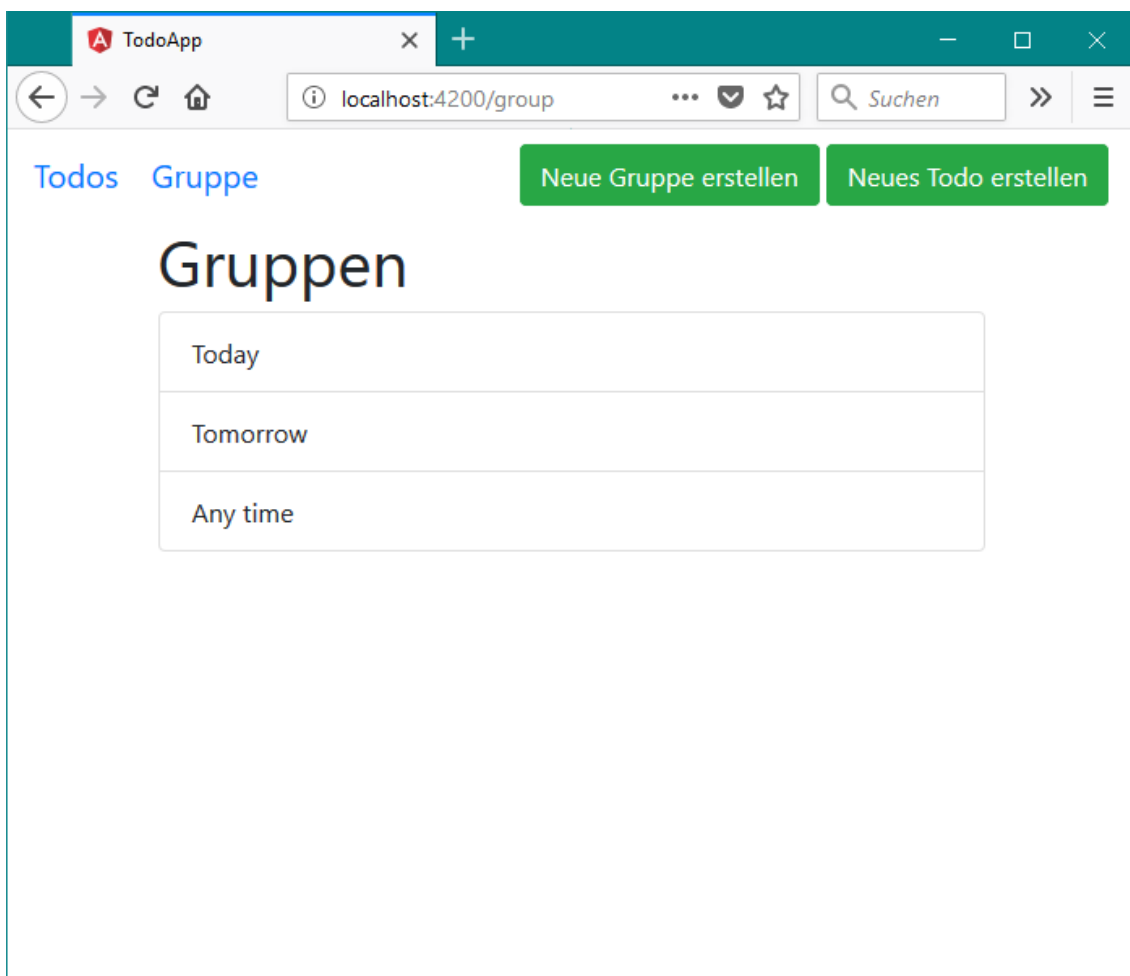
*Als Entwickler möchte ich meine Anwendung in unabhängige Module aufteilen und diese nur bei Bedarf nachladen.*

*Als Nutzer möchte ich einen schnellen Seitenaufbau, wenn ich die Seite besuche.*

### 1. Lernziele

- ✓ Sie sind befähigt einzelnen Komponenten in Module aufzugliedern.
- ✓ Sie verwenden loadChildren um Lazy Loading in Angular zu realisieren.

### 2. Ergebnis



### 3. Benötigte Daten

- `package.json`
- `src/app/app.component.html`
- `src/app/app.module.ts`
- `src/app/app-routing.module.ts`
- `src/app/group/group.module.ts`
- `src/app/group/group-routing.module.ts`
- `src/app/group/group.service.ts`
- `src/app/group/group-create.component.ts`
- `src/app/group/group-create.component.html`
- `src/app/group/group-list.component.ts`
- `src/app/group/group-list.component.html`

### 4. Anleitung

#### INITIALES SETUP UND CODE ANALYSE

---

Schritt 1:      Anfangszustand auschecken und Code analysieren

1. Checken Sie in git den Commit „Initialer Stand fuer zweiten Teil“ aus.
2. Starten Sie die Anwendung und machen Sie sich mit der Funktionalität kurz vertraut.
3. Schauen Sie sich den Code in **`app.component.html`**, **`app.module.ts`** sowie **`app-routing.module.ts`** an. Sie sollten hier keine Überraschung erleben.
4. Schauen Sie sich ebenfalls den Code der **`group-list`** und **`group-create`** Komponenten sowie den **`group.service.ts`** an, um ein wenig damit bekannt zu werden. Zu guter Letzt schauen Sie sich die Definition des **`GroupItems`** an.

## GROUP-MODUL ERSTELLEN

---

Schritt 2:      Group Module in neuem Ordner `/src/app/group`

5. Erstellen Sie den Ordner `/src/app/group`.
6. Verschieben Sie die Ordner der Komponenten **GroupCreateComponent** und **GroupListComponent** in den neu erstellten Ordner.
7. Verschieben Sie den Service **GroupService** in einen neu erstellten Ordner `/group/services`.
8. Korrigieren Sie **import**-Pfade falls notwendig.

---

Schritt 3:      **GroupModule** erstellen, Komponenten, abhängige Module (**CommonModule**, **HttpClientModule**, **FormsModule**) und **GroupService** aufnehmen und aus **AppModule** entfernen

9. Erstellen Sie die Klasse **GroupModule** in `/group/group.module.ts`.
10. Dekorieren Sie die Klasse mit **@NgModule** und übergeben zuerst ein Objekt mit leeren Arrays für die Attribute **declarations**, und **imports** an.
11. Nehmen Sie unter **declarations** die Komponenten **GroupListComponent** und **GroupCreateComponent** auf und entfernen diese entsprechend aus **app.module.ts** (die Importe nicht vergessen)
12. Importieren Sie **CommonModule**, **HttpClientModule** und **FormsModule**, indem Sie diese im **imports**-Array spezifizieren

---

Schritt 4:      **GroupRoutingModule** definieren und in **GroupModule** importieren

13. Erstellen Sie in `group/group-routing.module.ts` die Klasse **GroupRoutingModule**.
14. Dekorieren Sie die Klasse mit **@NgModule**, welchem Sie ein Objekt mit den Attributen **imports** und **exports** übergeben, denen Sie vorerst jeweils leere Arrays zuweisen.

15. Erstellen Sie außerhalb der Klasse die Konstante **groupRoutes** vom Typ **Routes** und weisen dieser Konstante ein Array mit einem Objekt zu.
16. Im Objekt spezifizieren Sie die Properties **path** mit dem Wert **""** (leerer String) und **children** mit einem leeren Array.
17. Spezifizieren Sie im Array zwei Objekte mit Objekteigenschaften für **path** und **component**, wobei sie beim **path ""** die **GroupListComponent** angeben und dem **path "new"** die **GroupCreateComponent** zuweisen.
18. Übergeben Sie dem Attribut **imports** in **@NgModule** den Wert

```
imports: [RouterModule.forChild(groupRoutes)]
```
19. In dem **exports**-Array von **@NgModule** spezifizieren Sie das **RouterModule**.
20. Nehmen Sie **GroupRoutingModule** im Dekorator des **GroupModule** unter **imports** auf.

## LAZY LOADING IMPLEMENTIEREN

---

Schritt 5:      Referenzen zu **GroupComponents** aus **AppRoutingModule** entfernen und das neue **Groups**-Modul einbinden

21. Entfernen Sie aus der Konstante **routes** in **app-routing.module.ts** die **children** Property unter der Spezifikation von **path: "groups"** , inklusive der Objekte zur Beschreibung der Routen **""** und **"new"**.
22. Das entfernte **children** Attribut wird nun durch das Attribut **loadChildren** ersetzt. Übergeben Sie diesem den folgenden String:  
**"./group/group.module#GroupModule"**

---

Schritt 6:      Lazy Loading des Gruppenmoduls im Browser nachverfolgen

23. Starten Sie die Anwendung in der Konsole mit **ng serve** und navigieren Sie zur Todo-Liste im Browser. Laden Sie diese Seite erneut, öffnen den Netzwerk-Tab in den Browser Entwicklertools (F12).

24. Löschen Sie die Historie des Netzwerk Tabs in den Entwicklertools und klicken auf „GRUPPE“ in der Navigation.
25. Sie werden sehen, dass nach dem Aufruf das eben erstellte JavaScript-Modul **group-group-module** vom Browser nachgeladen wird.

## 5. Kontrollfragen

- Welche Attribute sind für Lazy Loading im Routing-Modul des Eltern-Moduls zu setzen?
- Wie erstellen Sie ein neues Modul und verwenden dies?

## 6. Weiterführende Materialien

- Angular-Deployment Lazy Loading  
<https://tinyurl.com/gs-angular-deployment-1-1>
- Angular Asynchronous Routing  
<https://tinyurl.com/gs-angular-async-routing>