

# GABRIELE ROMANATO

## Angular: le route parametriche

Angular è un framework JavaScript ampiamente utilizzato per lo sviluppo di applicazioni web dinamiche e complesse. Una delle sue caratteristiche distintive è il sistema di routing, che consente di gestire la navigazione tra diverse visualizzazioni o pagine all'interno di un'applicazione. Le route parametriche sono un concetto avanzato all'interno del routing di Angular che consente di rendere le route più dinamiche e flessibili.

Le route parametriche consentono di definire percorsi dinamici all'interno di un'applicazione Angular. Piuttosto che avere percorsi statici, dove ogni URL corrisponde a una pagina specifica, le route parametriche consentono di catturare valori specifici dall'URL e utilizzarli dinamicamente nei componenti dell'applicazione.

Ad esempio, considera un'applicazione che gestisce una lista di prodotti e desidera visualizzare i dettagli di un prodotto specifico quando un utente fa clic su un collegamento. Utilizzando route parametriche, è possibile creare un percorso dinamico che cattura l'ID del prodotto dall'URL e lo utilizza per recuperare e visualizzare i dettagli del prodotto corrispondente.

Per utilizzare route parametriche in Angular, è necessario configurare le route all'interno del modulo dell'applicazione. Considera il seguente esempio:

```
const routes: Routes = [  
  { path: 'products', component: ProductsListComponent },  
  { path: 'products/:id', component: ProductDetailsComponent  
}  
];  
  
@NgModule({  
  imports: [RouterModule.forRoot(routes)],
```

```
    exports: [RouterModule]
  })
  export class AppRoutingModule { }
```

Nell'esempio sopra, abbiamo definito due route. La prima route corrisponde alla lista di prodotti e utilizza un percorso statico ('products'). La seconda route è una route parametrica ('products/:id'), dove ':id' è un segnaposto per un valore dinamico, in questo caso, l'ID del prodotto.

Una volta definita una route parametrica, è possibile accedere ai suoi parametri all'interno del componente associato. Ad esempio, nel nostro componente `ProductDetailsComponent`, possiamo recuperare l'ID del prodotto utilizzando l'oggetto `ActivatedRoute` di Angular. Ecco un esempio di come farlo:

```
import { ActivatedRoute } from '@angular/router';

export class ProductDetailsComponent implements OnInit {
  productId: string;

  constructor(private route: ActivatedRoute) { }

  ngOnInit(): void {
    this.productId = this.route.snapshot.paramMap.get('id');
    // Ora 'this.productId' contiene l'ID del prodotto dalla
    route.
  }
}
```

Quando si generano collegamenti o si naviga tra le route parametriche, è possibile passare i valori dei parametri dinamici. Ecco un esempio di come creare un collegamento a un prodotto specifico all'interno del componente `ProductsListComponent`:

```
<a [routerLink]="['/products', product.id]">Visualizza  
Dettagli</a>
```

In questo modo, Angular sostituirà automaticamente ':id' con l'ID del prodotto specifico quando viene generato il collegamento.

## Conclusioni

Le route parametriche in Angular offrono un modo potente per gestire dinamicamente le navigazioni all'interno di un'applicazione. Consentono di creare percorsi flessibili e riutilizzabili, migliorando l'esperienza dell'utente e semplificando la gestione delle informazioni dinamiche. Integrando correttamente le route parametriche nelle tue applicazioni Angular, puoi creare un'esperienza di navigazione più ricca e interattiva per gli utenti.