



Ziel

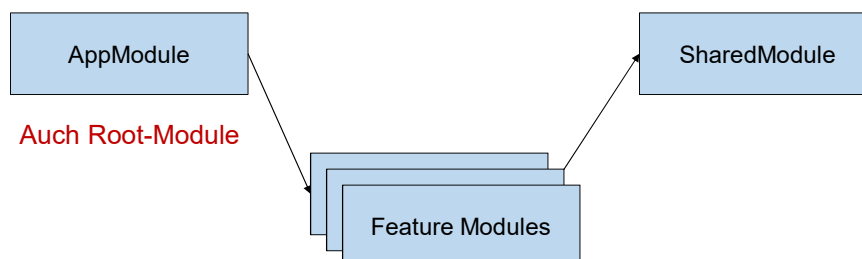
Möglichkeiten zum strukturieren einer Angular-Anwendung mit Module kennen lernen und nutzen können.

Aktuelles AppModule

```
@NgModule({  
  imports: [  
    BrowserModule, HttpClientModule, FormsModule  
  ],  
  declarations: [  
    AppComponent,  
    FlightSearchComponent,  
    CityPipe  
  ],  
  bootstrap: [  
    AppComponent  
  ]  
})  
export class AppModule {  
}
```

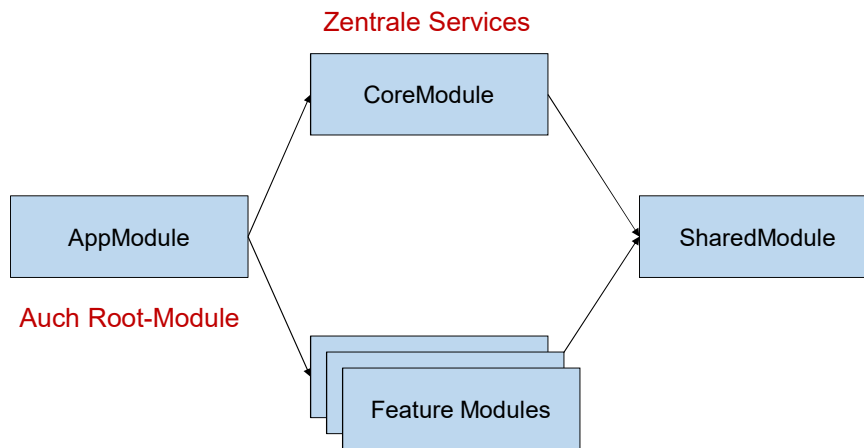
SOFTWAREarchitekt.at

Typische Modul-Struktur



SOFTWAREarchitekt.at

Typische Modul-Struktur



SOFTWAREarchitekt.at

SharedModule

```
@NgModule({
  imports: [
    CommonModule
  ],
  declarations: [
    CityPipe
  ],
  exports: [
    CityPipe
  ]
})
export class SharedModule {
}
```

Declarations müssen für andere Module exportiert werden!

SOFTWAREarchitekt.at

FlightBookingModule (Feature Module)

```
@NgModule({
  imports: [
    CommonModule, FormsModule, SharedModule
  ],
  declarations: [
    FlightSearchComponent
  ],
  providers: [
    FlightService <----- Provider sind global verfügbar!
  ],
  exports: [
    FlightSearchComponent
  ]
})
export class FlightBookingModule {
}
```

SOFTWAREarchitekt.at

AppModule (Root Module)

```
@NgModule({
  imports: [
    BrowserModule, HttpClientModule, FlightBookingModule
  ],
  declarations: [
    AppComponent
  ],
  bootstrap: [
    AppComponent
  ]
})
export class AppModule {
}
```

SOFTWAREarchitekt.at

Re-Exportieren von Modulen

```
@NgModule({  
  imports: [  
    CommonModule, FormsModule  
  ],  
  declarations: [  
    CityPipe  
  ],  
  exports: [  
    CityPipe, FormsModule  
  ]  
})  
export class SharedModule {  
}
```

SOFTWAREarchitekt.at

DEMO

SOFTWAREarchitekt.at

Übung

SOFTWAREarchitekt.at