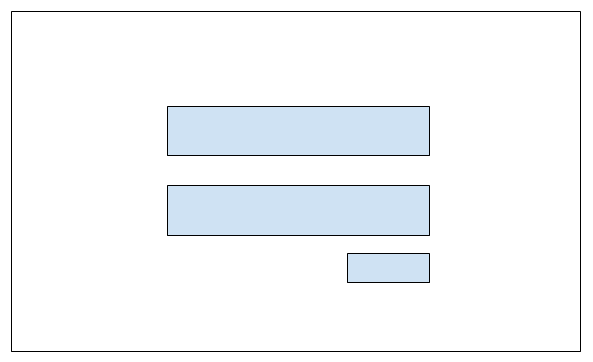
# Instructions - Exercise 3.3 – Passing Data to Routes, Part 2

**Layout**

Sign-in



profile-app, part 3

**Instructions**

* Make a copy of the enhanced-profile-app from Exercise 2.3 and add it to your week-3 directory
* Rename the application to secure-profile-app
* Delete the node\_modules directory
* Delete the package-lock.json file
* Open the angular.json file and find and replace all “enhanced-profile-app” entries with “secure-profile-app”
* Open the package.json file and change the name to “secure-profile-app”
* Run npm install and ng serve
  + You are doing this to test the application and confirm there are no errors
* Generate a new component and name it sign-in
* Generate a new component and name it home
* Add a new file to the app directory and name it app.routing.ts
  + This exercise demonstrates there are multiple ways to adding routing in an Angular project
* app.routing.ts
  + Add an import for Angular’s built-in Router module
    - Import { Routes } from ‘@angular/router’;
  + Add imports for the SignInComponent and HomeComponent
  + Create and export an array of Route objects
    - export const AppRoutes: Routes = []
    - path: ‘’, component: SignInComponent
    - path: ‘home’, component: HomeComponent
* app.module.ts
  + Add an import for the AppRoutes object from the app.routing.ts file
    - import { AppRoutes } from ‘./app.routing’;
  + Add an import for the RouterModule from Anuglar’s built-in Router
    - import { RouterModule } from ‘@angular/router’;
  + Under the imports array add a call to the RouterModule and pass-in the AppRoutes array of objects
    - RouterModule.forRoot(AppRoutes)
* Move the code from the app.component.html file to the home.component.html file
* app.component.html
  + Move the HTML code to the home.component.html file
  + Add Angular’s <router-outlet> directive
* app.component.ts
  + Remove the isLoggedIn variable
  + Change the variable title to ‘Secure Profile App’
* home.component.ts
  + Add a variable named isLoggedIn of type boolean
* sign-in.component.html
  + Create a bootstrap form with two fields: username and password and one button: Submit
    - The Bootstrap form should be inside a Bootstrap card, 35% width, 10% margin-top, and centered in the middle of the page
      * margin: 0 auto;
* sign-in.component.ts
  + Add an import statement for Angular’s built-in Router
  + Add Angular’s built-in Router to the components constructor
    - constructor(private router: Router)
  + Add a variable named isLoggedIn of type boolean and assign it a default value of false
  + Add a function named signIn
  + signIn()
    - Update the isLoggedIn variables value to true
    - Using the router.navigate() function navigate to the HomeComponent and pass-in the isLoggedIn value through the query params argument
      * this.router.navigate([‘/home’], {queryParams: {isLoggedIn: this.isLoggedIn}, skipLocationChange: true})
* home.component.ts
  + Add an import statement for Angular’s built-in ActivatedRoute
    - Import { ActivatedRoute } from ‘@angular/router’;
  + Add the ActivatedRoute object to the components constructor
    - constructor(private route: ActivatedRoute)
  + Assign the isLoggedIn variable the value being passed from the sign-in.component.ts using Angular’s built-in queryParamMap function
    - this.isLoggedIn = Boolean(this.route.snapshot.queryParamMap.get(‘isloggedIn’);
* home.component.html
  + Change the exercises name to Exercise 3.3 - Passing Data to Routes, Part 2

Note: the goal of this exercise is to demonstrate there are multiple ways to navigate and pass data between components.  You can pass data through the browsers URL (what we did in exercise 3.2) or you can pass data through the queryParams object and set skipLocationChange to true, which prevents route parameters from being shown in the browsers URL.