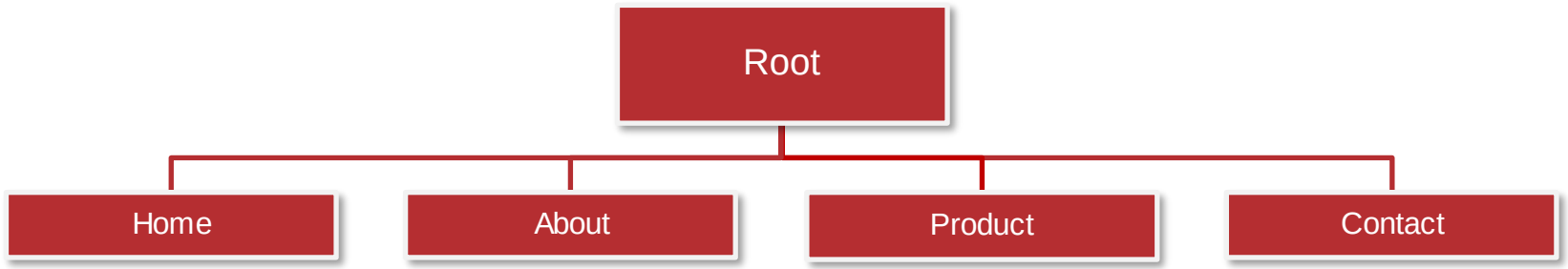
ANGULAR INTERVIEW QUESTIONS

1

What is Angular router

Angular Router is a mechanism in which navigation happens from one view to the next as users perform application tasks. It borrows the concepts or model of browser's application navigation. It enables developers to build Single Page Applications with multiple views and allow navigation between these views.



2

What are router imports

The Angular Router that represents a specific component view for a given URL is not a component of Angular Core. To import the necessary router components, use the @Angular/router library. As an example, we import them into the app module as shown below.

```
import { RouterModule,  Routes } from '@angular/router';
```

3

What is router outlet

A directive from the router library called RouterOutlet serves as a placeholder to indicate where in the template the router should display the parts for that outlet. Using a router outlet as a component

```
<router-outlet></router-outlet>
<!-- Routed components go here -->
```

4

What are router links

A directive on the anchor tags called RouterLink gives the router control over those elements. You can pass string values to the router-link directive as shown below because the navigation pathways are fixed.

```
<h1>Angular  Router</h1>
<nav>
  <a routerLink="/todosList"  >List of todos</a>
  <a routerLink="/completed"  >Completed todos</a>
</nav>
<router-outlet></router-outlet>
```

5

What are active router links

An activated route tree is known as RouterState. Each node in this tree is aware of the extracted parameters, resolved data, and "consumed" URL segments. The Router service and the routerState property both allow access to the current RouterState from any location inside the application.

```
@Component({templateUrl:'template.html'})
class MyComponent  {
  constructor(router: Router) {
    const state: RouterState = router.routerState;
    const root: ActivatedRoute  = state.root;
    const child = root.firstChild;
    const id: Observable<string> = child.params.map(p => p.id);
    //...
  }
}
```

6

What is router state

An activated route tree is known as RouterState. Each node in this tree is aware of the extracted parameters, resolved data, and "consumed" URL segments. The **Router service** and the **routerState** property both allow access to the current RouterState from any location inside the application.

```
@Component({templateUrl:'template.html'})
class MyComponent  {
  constructor(router: Router) {
    const state: RouterState = router.routerState;
    const root: ActivatedRoute  = state.root;
    const child = root.firstChild;
    const id: Observable<string> = child.params.map(p => p.id);
    //...
  }
}
```

7

What are router events

Through the Router.events property, you may trace the lifecycle of the route by observing the navigation events that the Router emits throughout each trip. Below is the order of router events.

Router events
NavigationStart
RouteConfigLoadStart
RouteConfigLoadEnd
RoutesRecognized
GuardsCheckStart
ChildActivationStart
ActivationStart
GuardsCheckEnd
ResolveStart
ResolveEnd
ActivationEnd
ChildActivationEnd
NavigationEnd
NavigationCancel
NavigationError
Scroll

8

What are activated routes

The information about a route connected to a component loaded in an outlet is contained in the ActivatedRoute field. It can also be used to navigate the router state tree. To access the data, the ActivatedRoute will be introduced as a router service.

```
@Component({...})
class MyComponent {
  constructor(route: ActivatedRoute) {
    const id: Observable<string> = route.params.pipe(map(p => p.id));
    const url: Observable<string> = route.url.pipe(map(segments => segments.join('')));
    // route.data includes both `data` and `resolve`
    const user = route.data.pipe(map(d => d.user));
  }
}
```

9

How do you define routes

A list of route definitions must be configured in a router. Using the RouterModule.forRoot() method, you configure the router with routes and add the resulting value to the AppModule's imports array.

```
const appRoutes: Routes = [
  { path: ‘user/:id',      component: UserDetailComponent },
  {
    path: ‘users',
    component: UserListComponent,
    data: { title: ‘Users List' }
  },
  { path: '',
    redirectTo: ‘/users',
    pathMatch: 'full'
  },
  { path: '**', component: PageNotFoundComponent  }
];

@NgModule({
  imports: [
    RouterModule.forRoot(
      appRoutes,
      { enableTracing: true } // <-- debugging purposes only
    )
    // other imports here
  ],
  ...
})
export class AppModule { }
```

10

What is the purpose of Wildcard route

The router throws an error and crashes the application if the URL doesn't match any of the specified routes. The wildcard route can be used in this situation. A route that matches any URL has a path that consists of two asterisks. Using the example below, you can define PageNotFoundComponent for the wildcard route.

```
{ path: '**', component: PageNotFoundComponent }
```

Do we need routing module always

No. The Routing Module was a design decision. When the configuration is straightforward, you can omit the routing module (such as AppRoutingModule) and incorporate the routing configuration right into the auxiliary module (such as AppModule). But when the setup is complex and involves specific guard and resolver services, it is advised.