Optional: Class-based Resolvers

In the previous lecture, you learned how to define route data resolvers as functions. This is the modern, recommended approach.

But you can also define resolvers as classes - like this:

```
@Injectable({ providedIn: 'root' })
export class UserNameResolver implements Resolve<string> {
  constructor(private usersService: UsersService) {}
  resolve(activatedRoute: ActivatedRouteSnapshot, state: RouterStateSnapshot) {
    const userName =
      this.usersService.users.find(
        (u) => u.id === activatedRoute.paramMap.get('userId')
      )?.name || '';
    return userName;
  }
}
```

The resolve class is then assigned to a route like this:

```
{
  path: 'users/:userId', // <your-domain>/users/<uid>
  component: UserTasksComponent,
  children: userRoutes,
  data: {
    message: 'Hello!',
  },
  resolve: {
    userName: UserNameResolver,
  },
}
```

Please note that this approach is deprecated though - the modern, function-based one (shown in the previous lecture) is therefore recommended!