

Exercise (Instructions): Angular and Promise

Part 1

Objectives and Outcomes

In this exercise you will use JavaScript promises to redesign your Angular services so that they can operate asynchronously and return the results whenever available. At the end of this exercise you will be able to:

- Make use of JavaScript promises within your Angular application
- Redesign your services to return promises and resolve them when the results are available
- Consume the promises within your components and handle the resolution or rejection of the promise

Reconfigure Services to Return Promises

- Open `dish.service.ts` and update its methods as follows:

```
1  getDishes(): Promise<Dish[]> {  
2    return Promise.resolve(DISHES);  
3  }  
4  
5  getDish(id: number): Promise<Dish> {  
6    return Promise.resolve(DISHES.filter((dish) => (dish.id === id))[0]);  
7  }  
8  
9  getFeaturedDish(): Promise<Dish> {  
10   return Promise.resolve(DISHES.filter((dish) => dish.featured)[0]);  
11 }
```

- Similarly update `leader.service.ts` and `promotion.service.ts` to return promises.

Handling Promises

- Open `menu.component.ts` and update its handling of the call to `getDishes()` method as follows:

```
1  this.dishService.getDishes()  
2    .then(dishes => this.dishes = dishes);
```

- Similarly update `about.component.ts`, `dishdetail.component.ts` and `home.component.ts` where the services are invoked to fetch the data.
- Save all the changes and do a Git commit with the message "Promise Part 1"

Conclusions

In this exercise you learnt the use of JavaScript promises to handle asynchronous delivery of data from the services to the components.

Mark as completed

