Learn Angular

BY MAKING A QUIZ APP FROM SCRATCH

HUNGRYTURTLECODE.COM

Part 18

Check Out The Whole Course Index

No Quiz Is Complete Without It Being Marked

In the last part we reference a function on the quizMetrics factory that we have not yet created. So it will be the task of this part to create that function – a function that will mark the answers to the quiz and calculate how many correct answers that user gave. Let's get on with marking the quiz.

If you want to see the app for yourself, check it out here.

The git repo can be found here.

Click the image to go to the video on youtube or go here https://www.youtube.com/watch?v=mBDbwKr4DyQ&list=PLqr0oBkln1FBmOjK24_B4y_VAA8736wPq&index=18

The next part can be found here

Without wasting any time, jump into the quizMetrics factory and get started. The first thing we will need to mark the quiz is the correct answers to all the questions. The correct answers will be another piece of data in the dataService factory. Much like we have copy and pasted in the data for the list and the quiz, we will also paste in the data for the correct answers.

So at the bottom of the data service factory, paste the following:

```
var correctAnswers = [1, 2, 3, 0, 2, 0, 3, 2, 0, 3];
```

Then inside dataObj of that service we add the correctAnswers. Like this:

```
var dataObj = {
    turtlesData: turtlesData,
    quizQuestions: quizQuestions,
    correctAnswers: correctAnswers
};
```

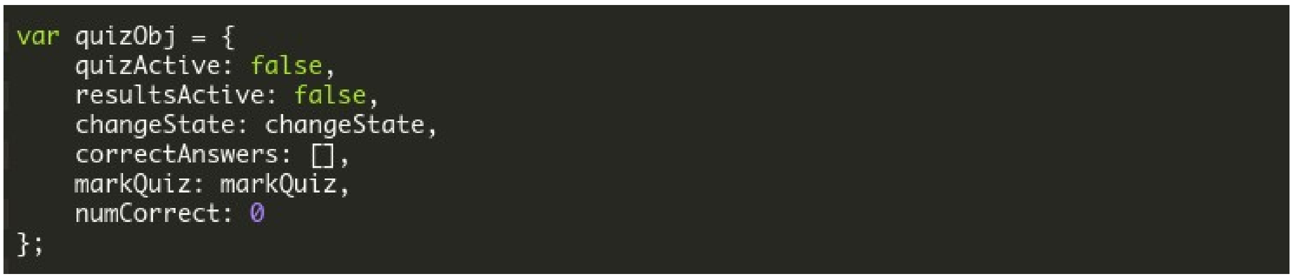
To get hold of this data from inside the quizMetrics factory we will need to inject the dataService into the factory (yes, we can inject dependencies into services too!). This process is exactly the same as the dependency injection that you have seen already in the controllers.

```
angular
    .module("turtleFacts")
    .factory("quizMetrics", QuizMetrics);

    QuizMetrics.$inject = ['DataService'];

    function QuizMetrics(DataService){
        // Rest of quizMetrics code left out to keep snippet clean
    }
```

The Function That Will Be Marking The Quiz

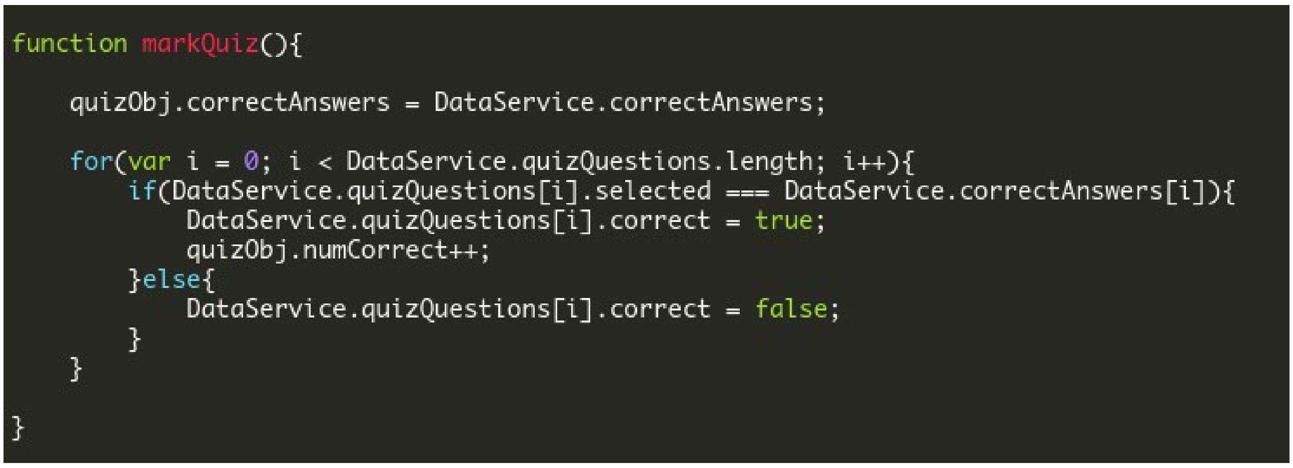
Now that we have the correct answers and have access to them in the quizMetrics factory, we can create the markQuiz function. Before we start that we will add a reference to that function in the dataObj of the quizMetrics factory, along with a few properties.

These other properties include a property to track the total number of correct answers the user has given (defaulted to 0), and a property to house the correct answers inside the quizMetrics factory object. Of course, we could just access the answers from the dataService, but the answers are a bit of data related to the quiz, so we will just keep things easy to understand and keep that stored there.



```
var quizObj = {
    quizActive: false,
    resultsActive: false,
    changeState: changeState,
    correctAnswers: [],
    markQuiz: markQuiz,
    numCorrect: 0
};
```

Now onto the implementation of the markQuiz function. This is simply a case of looping through all the questions in the quiz and then comparing the answer the user gave (which is stored in the selected property for each question) against the correct answer. If they match, we will set the "correct" property in the data to true and increment the numCorrect property. If the answers do not match then the correct flag is set to false.



```
function markQuiz(){

    quizObj.correctAnswers = DataService.correctAnswers;

    for(var i = 0; i < DataService.quizQuestions.length; i++){
        if(DataService.quizQuestions[i].selected === DataService.correctAnswers[i]){
            DataService.quizQuestions[i].correct = true;
            quizObj.numCorrect++;
        }else{
            DataService.quizQuestions[i].correct = false;
        }
    }

}
```

This Part Was Short, On To The Next One

We have now finished the entire quiz controller and can move on to thinking about the implementation of the results controller.

See you in part 19.

Adrian