sitepoint premium

Handout

Your First Meteor 1.2 Application

Lesson 10 - Step 3

David Turnbull

One Leaderboard Per User

To make our application useful, we need to make it so registered users each have their own independent list of players. It might not be immediately obvious how to go about doing this, and the most difficult part of programming is figuring out how to approach such problems, but the process itself doesn't involve a lot of steps.

To begin, create a `currentUserId` variable inside the submit form event and make it so this variable is equal to `Meteor.userId`.

```
'submit': function() {
    var currentUserId = Meteor.userId;
    // ...
}
```

This `Meteor.userId` function is an easy way to retrieve the unique ID of the currently logged in user, and it's provided to us by the accounts password package.

Next, add a `createdBy` field to the `insert` function and pass through the `currentUserId` variable as the value:

```
Template.addPlayerForm.events = function() {
    'submit form': function(event) {
        event.preventDefault();
        var currentUserId = Meteor.userId;
        var playerName = event.target.playerName.value;
        PlayersList.insert({name: playerName, score: 0, createdBy: currentUserId});
        event.target.playerName.value = '';
    }
}
```

Because of this code, when a user adds a player to the leaderboard, the unique Id of that user will be associated with the player's document.

Demo

To demonstrate this, save the file, switch back to the browser, and add a player to the leaderboard. Then use the `find` and `fetch` functions on the `PlayersList` collection, and click the downward-facing arrow for the most recently created document. You'll see how this document contains the id of the user who added this player to the collection.

Modifying the Player Function.

Next, we'll modify the `player` function. Inside this function, set up another `currentUserId` variable and change the `return` statement so it only returns players witha `createdBy` field equal to the ID of the currently logged in user:

```
'player': function() {
    var currentUserId = Meteor.userId;
    return PlayersList.find({createdBy: currentUserId}, {sort: {name: 1, score: -1}});
}
```

This makes it so users only see players they added to the leaderboard, thereby creating the effect that each user has their own unique list of players. But because of this change, we now have a number of players in the database that are not attached to any particular user, which means they don't need to exist.

We could remove them manually, but instead, let's reset the entire project, wiping the database clean and giving ourselves a fresh start. To do this, switch to the command line, stop the local server with the `Ctrl+C` keyboard shortcut, and run the following command:

```
meteor reset
```

This will remove all of the data from the database, which means that from this point onward, all players will be attached to a user.