



## Player's ID

When a user clicks on one of the players, we want to grab the unique ID of that player and store it inside the selected player session. This will allow us to change the background color of that player's list item.

If you're not sure what I mean by the unique ID of the player, think back to when we inserted players into the `PlayersList` collection. Each time we used the `insert` function, a random jumble of numbers and letters would appear - that jumble was the unique ID of that player.

To implement this feature, create a `playerId` variable in the `click` event and pass this variable into the `Session.set` function. The trick is to assign the unique ID to the `playerId` variable:

```
jsj
Template.leaderboard.events({
  'click .player': function() {
    var playerId = this._id;
    Session.set('selectedPlayer', playerId);
    var selectedPlayer = Session.get('selectedPlayer');
    console.log(selectedPlayer);
  }
});
```

There's two things going on. First, we have a reference to `this`, and the value of `this` depends on the context. In this particular context, `this` refers to the document of the player that has just been clicked.

## Unique ID

Second, the `_id` part is the name of the field that contains the unique ID of the player in each of our players' documents. So in the same way we created a name and score field for our players, Mongo creates an `_id` field for each document. The underscore, itself, doesn't have any special significance, though. It's simply part of the field's name.

Because of this change, we can now click on any of the players in the list to see the ID of that player appear in the console. It's worth mentioning that only one value can be stored inside a session at any time. This means each time the user clicks on a player, the value inside the selected player session will be replaced by the new value.

You can also remove the `Session.get` function and the `console.log` statement from the event, as these were only needed for the demonstration.