



Introduction

The main feature of the Leaderboard application is the list of players. Without displaying a list of players within the interface, we can't build anything else of value. As such, this is a good place to start, from the middle of the application and working out toward the finer details. But this begs the question, where do we store the data that will be associated with each player? Where do we store their names, for instance, or their scores?

Collections

The broad answer would be in a database but the more useful answer is in a **collection**. As we covered in the previous video, a collection is equivalent to an SQL table. To illustrate the purpose of a collection, imagine we're creating our own version of WordPress with Meteor. If that were the case, we would create a collection for the posts, and a collection for the comments, and a collection for the pages.

So, add the following statement to the very top of the JavaScript file:

```
new Mongo.Collection('players');
```

You can choose whatever name you like but the name must be unique to the project.

Now we need a way to reference the collection:

```
PlayersList = new Mongo.Collection('players');
```

Notice that we didn't use the `var` keyword - this is because we want the variable to be a global variable which will allow us to reference and manipulate the collection through out all of the project's files.

Confirm that the Collection Exists

To confirm that the collection exists, save the file, switch back to the browser, and enter the `PlayersList` variable into the console. The console should spit back some information that confirms the collection is active. If an error is returned, it's probably because you mistyped the variable name in the console, or made a mistake with the syntax in the code.