



Each Blocks

At the moment, our helper function just returns a static value. What we want to happen, though, is for the function to retrieve all of the documents from the `PlayersList` collection. This will allow us to display data from those documents within the interface. To achieve this, replace the static text that's being returned in the helper function with

```
'player': function() {  
    return PlayersList.find();  
}
```

We now have access to the returned data from inside the `leaderboard` template. Remove the `player` tag from the `leaderboard` template. As long as we're returning an array of data, we have to loop through the data that's returned. To do this, we can use the `Spacebar` syntax to create an `each` block:

```
{{#each player}}  
    test  
{{/each}}
```

This code will loop through all of the documents that are retrieved by the helper function and for each document that is retrieved, the word "test" will appear inside the interface. Because there are currently six players in the `players list` collection, this word will appear six times.

Within the `each` block, we can also retrieve the values of the fields from inside our documents. So because we're pulling data from the `PlayersList` collection, we can display the values of the `name` and `score` fields:

```
{{#each player}}  
    {{name}}: {{score}}  
{{/each}}
```

To make the interface a little more usable, wrap the `each` block in an `unordered list`, and wrap each individual player inside a `list item`:

```
<ul>  
  {{#each player}}  
    <li>{{name}}: {{score}}</li>  
  {{/each}}  
</ul>
```

By default players will be sorted by the time they were inserted into the collection from the players added first, to the players added more recently, but this is something we'll change in a later chapter.