

Instructions:

1. FORK this project: <https://repl.it/@pamelafox/NodeScoreStorer>
2. FIND a previous quiz project of yours that you'd like to store scores for.
3. FIND index.html in this project's public/ folder. Replace the file with the HTML from your quiz.
4. IF you had any external *.css or *.js files for your quiz project, also put those inside public/.
5. START the project and make sure your quiz is working as expected.
6. DELETE the contents of the db.sqlite3 file
(That's where your database is stored, and you don't want to start with my data!)
7. ADD an `<input>` tag to your HTML, so that players can enter their name:

```
<label>
```

Your name:

```
<input id="player" type="text">
```

```
</label>
```

8. ADD an `<ol>` tag to your HTML, which will later display the top scores:

```
<h4>High scores</h4>
```

```
<ol id="scores">
```

Loading...

```
</ol>
```

9. ADD the insertScore function to your code. That's what your code will call when it wants to insert a score in the database. It uses XMLHttpRequest to send the new data to the server for storage.

```
function insertScore(player, score) {  
  var url = '/scores/insert';  
  
  var request = new XMLHttpRequest();  
  request.addEventListener('load', function () {  
    if (this.status !== 200) {  
      alert('Error!');  
      console.log(this.response);  
    } else {  
      getScores();  
    }  
  });  
}
```

```

    }
  });
  request.open('POST', url, true);
  request.setRequestHeader('Content-type', 'application/json');
  var data = {'player': player, 'score': score};
  request.send(JSON.stringify(data));
}

```

10. ADD the `getScores` function to your code. This is the function you can call whenever you want to display/refresh the top scores. It uses `XMLHttpRequest` to request the scores data from the server database, and then renders them in the HTML.

```

function getScores() {
  var url = '/scores/top10';

  var request = new XMLHttpRequest();
  request.addEventListener('load', function () {
    var scores = JSON.parse(this.responseText);
    console.log(scores);
    var scoresDiv = document.getElementById('scores');
    scoresDiv.innerHTML = '';
    for (var i = 0; i < scores.length; i++) {
      scoresDiv.innerHTML += '<li>' + scores[i].player + ': ' +
scores[i].score;
    }
  });
  request.open('GET', url, true);
  request.setRequestHeader('Content-type', 'application/json');
  request.send();
}

```

11. FIGURE OUT where to call `getScores()` and `insertScore()` in your code. When would you want to insert a score? When would you want to display the top 10 scores? You might call one or both of those multiple times. Experiment!
12. Bonus: Try forking this project to create a version of your Mad Libs project that stores all user's stories and displays them on the page.