

Put data into a database from your Node.js Application

- 1 In your web browser, go to IBM Bluemix™ at <https://ace.ng.bluemix.net> and LOG IN.
- 2 On a separate tab in your browser, go to DevOps Services [rewillen | NodeJS_Simple_3](#).
- 3 Click on [EDIT CODE](#)
- 4 [FORK](#) the Application, [Enter a Name](#) for your project, and [Save](#).
- 5 Click on [app.js](#) to open the code for the application.
- 6 Find the [client.incrby](#) line. This is the statement that is incrementing a DBUserCountBy1 variable in Redis, and returning the updated value in a variable, [replyBy1](#).
- 7 Add a second Redis counter, DBUserCountBy2. Use another [client.incrby](#) statement to increment it by 2, and return the value in a new variable, [replyBy2](#). To ensure that the calls to Redis are synchronized, this statement needs to be nested inside the increment of the DBUserCountBy1. Be sure to add the closing `}` also.

```
56     client.incrby("DBUserCountBy1", 1, function (err, replyBy1) {
57         client.incrby("DBUserCountBy2", 2, function (err, replyBy2) {
```

- 8 Next, update the render to also pass the DBUserCountBy2 which was returned in the [replyBy2](#) variable from the Redis function call.

```
res.render('index', {userCount: userCount, DBUserCountBy1: replyBy1, DBUserCountBy2: replyBy2});
```

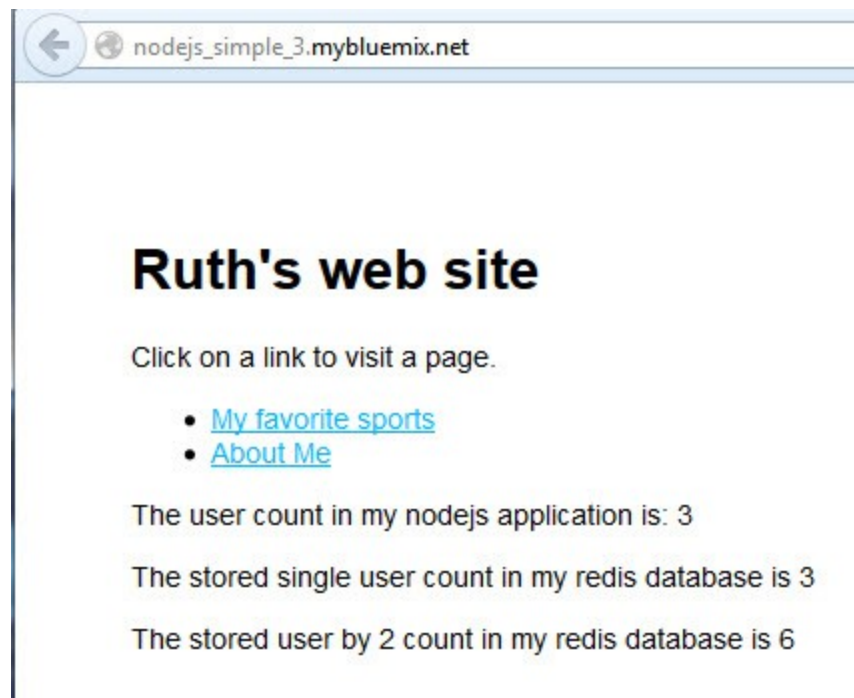
Your [app.get](#) code block should now look like this:

```
53 app.get('/', function(req, res) {
54     userCount = userCount + 1;
55     // Increment the value in the database and render the results
56     client.incrby("DBUserCountBy1", 1, function (err, replyBy1) {
57         client.incrby("DBUserCountBy2", 2, function (err, replyBy2) {
58             res.render('index', {userCount: userCount, DBUserCountBy1: replyBy1, DBUserCountBy2: replyBy2});
59         });
60     });
61 });
```

- 9 In the views directory, open [index.ejs](#).
- 10 Update the index.ejs HTML page to render to DbuserCountBy2 like this: `<p>The stored user count by two in my redis database is <%= DBUserCountBy2 %></p>`.

- 11 [DEPLOY](#) the application and use Manual Deployment like you did on card 1. You should see your

changes in the browser. Hit the refresh button.



12 There are a lot of things you can now do on your own. I'll add some more ideas here when I have time. For example, perhaps some additional experiments with [Redis](#) such as using strings instead of numbers, or some basic array processing?