Connect to MongoDB

1 ## Create your Node.js Application

Create a file to contain your application called `index.js` in your `node_quickstart` project directory.

Copy and paste the following code into the `index.js` file:

```js
const { MongoClient } = require("mongodb");
// Replace the uri string with your connection string.
const uri = "<connection string uri>";
const client = new MongoClient(uri);
async function run() {
try {
const database = client.db('sample_mflix');
const movies = database.collection('movies');
// Query for a movie that has the title 'Back to the Future'
const query = { title: 'Back to the Future' };
const movie = await movies.findOne(query);
console.log(movie);
} finally {
// Ensures that the client will close when you finish/error
await client.close();
}
}
run().catch(console.dir);
```

2 ## Assign the Connection String

Replace the `<connection string uri>` placeholder with the connection string that you copied from the Create a Connection String step of this guide.

3 ## Run your Node.js Application

In your shell, run the following command to start this application:

```
node index.js
```

You should see the details of the retrieved movie document in the command line output:

```
{
_id: ...,
```

```
plot: 'A young man is accidentally sent 30 years into the past...',
genres: [ 'Adventure', 'Comedy', 'Sci-Fi' ],
...
title: 'Back to the Future',
...
}
```

If you encounter an error or see no output, check whether you specified the proper connection string in the `index.js` file, and that you loaded the sample data.

After you complete these steps, you should have a working application that uses the driver to connect to your MongoDB deployment, runs a query on the sample data, and prints out the result.

> ⓘ
> **NOTE**
>
> If you run into issues on this step, ask for help in the MongoDB Community Forums or submit feedback using the **Share Feedback** tab on the right or bottom right side of this page.