UNIVERSIDADE da MADEIRA

CTeSP CURSOS TÉCNICOS SUPERIORES PROFISSIONAIS

Tecnologias e Programação de Sistemas de Informação

NODE & MYSQL

Desenvolvimento Web - Back-End | David Jardim

Cofinanciado por:

MADEIRA 14-20

REGIÃO AUTÓNOMA DA MADEIRA

PORTUGAL 2020

UNIÃO EUROPEIA
Fundo Social Europeu

mysql

public

Readme | 4 Dependencies | 3,251 Dependents | 60 Versions

mysql

Table of Contents

install

```
> npm i mysql
```

weekly downloads

233,364

version	license
2.15.0	MIT

open issues	pull requests
107	24

repository

github

last publish

7 months ago

collaborators

Install Express and MySQL

```
$ mkdir myapp
$ cd myapp
```

Use the `npm init` command to create a `package.json` file for your application. For more information on how `package.json` works, see Specifics of npm's package.json handling.

```
$ npm init
```

This command prompts you for a number of things, such as the name and version of your application. For now, you can simply hit RETURN to accept the defaults for most of them, with the following exception:

```
entry point: (index.js)
```

Enter `app.js`, or whatever you want the name of the main file to be. If you want it to be `index.js`, hit RETURN to accept the suggested default file name.

Now install Express in the `myapp` directory and save it in the dependencies list. For example:

```
$ npm install express --save
```

```
$ npm install mysql
```

Example

```
var mysql = require('mysql')
var connection = mysql.createConnection({
  host     : 'localhost',
  user     : 'dbuser',
  password : 's3kreee7',
  database : 'my_db'
});

connection.connect()

connection.query('SELECT 1 + 1 AS solution', function (err, rows, fields) {
  if (err) throw err

  console.log('The solution is: ', rows[0].solution)
})

connection.end()
```

Perform queries

```
.query(sqlString, callback)
```

```
connection.query('SELECT * FROM `books` WHERE `author` = "David"', function (error, results, fields) {
  // error will be an Error if one occurred during the query
  // results will contain the results of the query
  // fields will contain information about the returned results fields (if any)
});
```

Perform queries

```
.query(sqlString, values, callback)
```

```
connection.query('SELECT * FROM `books` WHERE `author` = ?', ['David'], function (error, results, fields) {
  // error will be an Error if one occurred during the query
  // results will contain the results of the query
  // fields will contain information about the returned results fields (if any)
});
```

Terminating connections

```
connection.end(function(err) {
  // The connection is terminated now
});
```

This will make sure all previously enqueued queries are executed before sending a quit request to the MySQL server.

```
connection.destroy();
```

This will cause an immediate termination of the underlying socket.

MySQL Connection Pool

```
var pool = mysql.createPool({
    connectionLimit: 100, //important
    host: '127.0.0.1',
    user: '***',
    password: '***',
    database: 'user',
    debug: false
});
```

Connection pooling is a technique of creating and managing a pool of connections that are ready for use by any thread that needs them.

Connection pooling can:

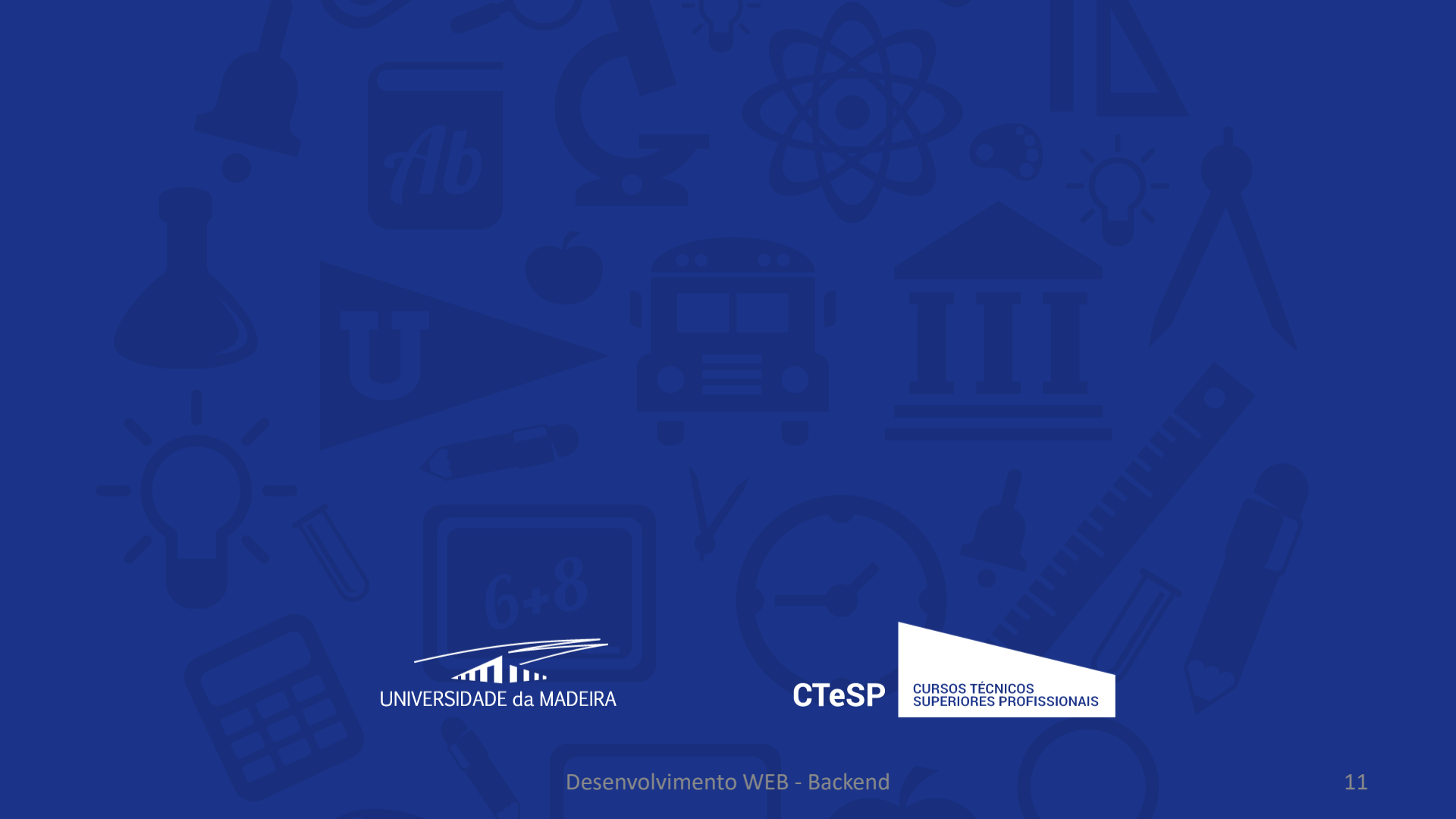
- Greatly increase the performance
- Reduced connection creation time
- Simplified programming model
- Controlled resource usage

Handling and managing pool connections

```
var mysql = require('mysql');
var pool  = mysql.createPool({
  connectionLimit : 10,
  host            : 'example.org',
  user            : 'bob',
  password        : 'secret',
  database        : 'my_db'
});

pool.query('SELECT 1 + 1 AS solution', function (error, results, fields) {
  if (error) throw error;
  console.log('The solution is: ', results[0].solution);
});
```

Handling and managing pool connections

```
pool.getConnection(function(err, connection) {
        if (err) {
            connection.release();
            res.json({ "code": 100, "status": "Error in connection database" });
            return;
        }

        connection.query(user_query, function(err, rows, fields) {
            if (err) throw err;

            if (rows.length == 0) {
                console.log("No device token found");
                callback(null, null);
            } else {
                callback(null, rows[0]);
            }
        });
    });
```


UNIVERSIDADE da MADEIRA
CTeSP
CURSOS TÉCNICOS
SUPERIORES PROFISSIONAIS