

THE DEVELOPER'S CONFERENCE

**NodeJs – Como utilizar o Node.js com bancos de dados relacionais.
Uma abordagem com Knex.js e Bookshelf.js**

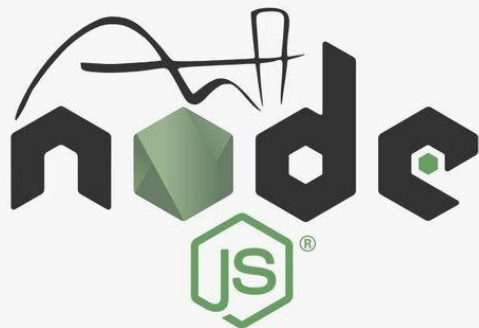
Wesley Lopes
FullStack Developer



THE
DEVELOPER'S
CONFERENCE

Olá!

Sou Wesley Lopes ou @wesdeveloper, Organizador do Meetup Node.Js BH e atualmente Desenvolvedor FullStack JS na Rethink.



Problema



Utilizar um banco de dados relacional com Node.js

Solução



Knex.js e Bookshelf.js

Solução





THE
DEVELOPER'S
CONFERENCE



KNEX.JS

Knex.js



A SQL query builder that is *flexible*, *portable*, and *fun* to use!

Knex.js is a "batteries included" SQL query builder for **Postgres**, **MSSQL**, **MySQL**, **MariaDB**, **SQLite3**, **Oracle**, and **Amazon Redshift** designed to be flexible, portable, and fun to use. It features both traditional node style callbacks as well as a promise interface for cleaner async flow control, a stream interface, full featured **query** and **schema** builders, **transaction support (with savepoints)**, connection pooling and standardized responses between different query clients and dialects.

```
knex.select().table('books')
```

Outputs:

```
select * from "books"
```



THE
DEVELOPER'S
CONFERENCE



BOOKSHELF.JS

Bookshelf.js



Bookshelf is a JavaScript **ORM** for Node.js, built on the **Knex** SQL query builder. Featuring both promise based and traditional callback interfaces, providing **transaction** support, eager/nested-eager relation loading, polymorphic associations, and support for **one-to-one**, **one-to-many**, and **many-to-many** relations.

```
User.where('id', 1).fetch({withRelated: ['posts.tags']}).then(function(user) {  
  console.log(user.related('posts').toJSON());  
}).catch(function(err) {  
  console.error(err);  
});
```

O que é um Query Builder?



THE
DEVELOPER'S
CONFERENCE

Query builder é uma interface de comunicação que nos permite escrever instruções/comandos SQL, independente do banco de dados, de uma forma clara.

O que é um ORM?



ORM (Object Relational Mapper) é uma técnica que nos permite mapear/abstrair os dados armazenados no banco para objetos relacionais e assim conseguirmos manipulá-los de uma forma prática e padronizada.

Prós x Contras na utilização de um Query Builder/ORM



THE
DEVELOPER'S
CONFERENCE

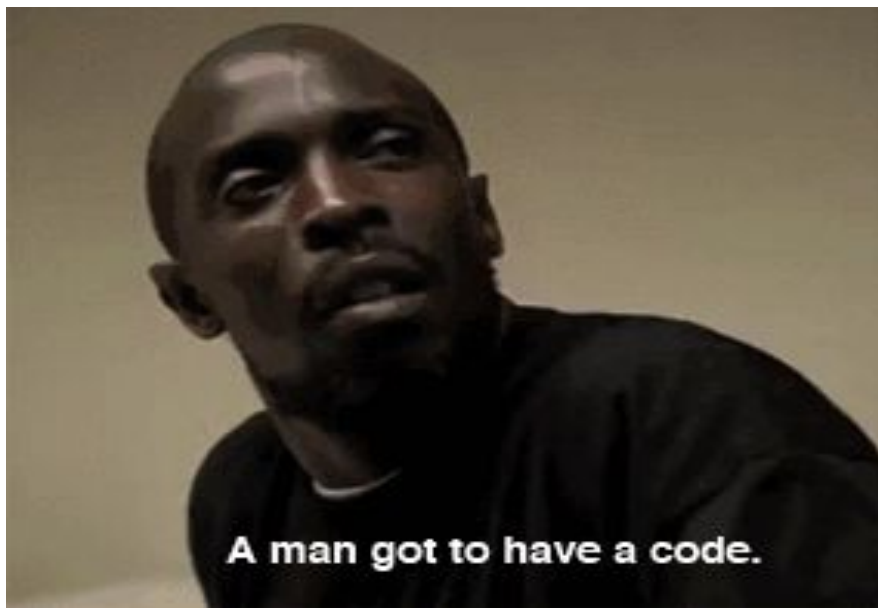
Prós

- *Facilita a modelagem das tabelas.*
- *Abstração do banco de dados.*
- *Baixo acoplamento ao banco de dados.*
- *Abstração da sintaxe SQL.*
- *Permite a visualização dos dados de forma Orientada a Objetos.*
- *Não exige um conhecimento avançado em SQL.*

Contras

- *Pode afetar o desempenho da comunicação com o banco de dados.*
- *Pode gerar mais queries que o necessário.*
- *Pode perder algumas particularidades dos bancos.*

Vejam os um cenário de utilização real



https://github.com/wesdeveloper/node_query_builder_orm



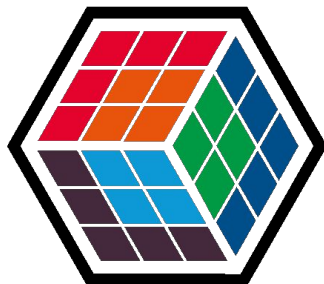
THE
DEVELOPER'S
CONFERENCE

Obrigado!



<https://tinyurl.com/tdcbh-node-dbs>

Wesley Lopes
@wesdeveloper



THE DEVELOPER'S CONFERENCE