Jira Clone

I. Packages:

* I using: Express, typeORM, PostgreSQL.
* Write by: Typescript

II. TypeORM:

1. Create database connect:

* How to create connect to db-postges using typeorm ?

1. Create an Entity

* Step-by-Step Guide
* Create a model
* Create an entity
* Adding table columns
* Creating a primary column
* Creating an auto-generated column
* Column data types
* Creating a one-to-one relation
* Save a one-to-one relation
* Inverse side of the relationship
* Creating a many-to-one / one-to-many relation
* Creating a many-to-many relation

1. Relations

* What are relations
* Relation options
* Cascades
* Cascade Options
* @JoinColumn options
* @JoinTable options

1. FAQ

* What does "owner side" in relations mean or why we need to put @JoinColumn and @JoinTable decorators?

|  |  |  |  |
| --- | --- | --- | --- |
| Keyword | Link document | Version | Other |
| Typeorm | https://www.npmjs.com/package/typeorm/v/0.2.41 | 0.2.41 |  |
| Typeorm | https://typeorm.io | 0.3.11 |  |
| Typeorm | https://typeorm.io/relations | 0.3.11 |  |
| Typeorm | https://typeorm.io/faq#what-does-owner-side-in-a-relations-mean-or-why-we-need-to-use-joincolumn-and-jointable |  |  |

III Express

1. What is Express?

* Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.

1. Basic routing

* Docs: <https://expressjs.com/en/starter/basic-routing.html>

1. Writing middleware for use in Express apps

* Docs: <https://expressjs.com/en/guide/writing-middleware.html>

1. Using middleware

* Docs: <https://expressjs.com/en/guide/using-middleware.html>

1. Error Handling

* Docs: <https://expressjs.com/en/guide/error-handling.html>

1. More….

Thông tin

1. const app = express() – What is app ?

* Khi thực hiện call “express( )”. Nó trả về một instance (cứ hiểu là nó trả về một đối tượng), đối tượng này đại diện cho ứng dụng nodejs được gọi là “app”.
* Nếu làm một ứng dụng website thì thằng express trả về một đối tượng để có thể xây dựng ra website.
* Sử dụng xuyên suốt đến khi website hoàn thành.

1. Base routing

* Handler functions là một function dùng để sử lý một việc gì đó.
* req hiểu đơn giản là biến chứa những thông tin mà ứng dụng ở phía Client yêu cầu lên Server.
* res hiểu đơn giản là biến giúp tùy chỉnh kết quả trả về qua resonse

1. middleware urlencoded

* Dùng để xử lý dữ liệu từ form submit lên.
* Cú pháp: app.use(express.urlencoded()).

1. Midleware json

* Sử dụng trong TH gửi dữ liệu từ client lên server:

+ Một là qua việc submit form (thông qua HTML).

+ Hai thông qua việc sử dụng thư viện trong js hoặc sử dụng code js để submit ví dụ như: XMLHttpRequest, fetch, axios….,

1. Tìm hiểu keywords pody-parse và qs npm
2. Middleware là gì?

**Ý nghĩa:**

* Là phần mềm trung gian (đứng giữa các thành phần trong mô hình phần mềm).
* Chú ý: Khái niệm “middleware” là khái niệm trong phần mềm, middleware là khái niệm chung, nó không phải của nodeJS hay của expressJS.
* Middleware là khái niệm chung trong việc sản xuất, thết kế phần mềm.
* Là thành phần đứng giữa các thành phần trong phần mềm. Bản thân middleware cũng là một thành phân đứng giữa những thành phần khác.
* Làm nhiệm vụ kết nối, trung chuyển, kiểm soát giữa các thành phần.
* Ví dụ:

Brower (client) ---------------- Request ---------------🡪 Server (Node)

Brower (client) 🡨------------- Response ---------------- Server (Node)

Vai trò:

* Giống như “Bác bảo vệ”

Nhà -----------------------🡪 Bác bảo vệ (middleware) : Sự kiện (yc: soát vé)

Nhà 🡨----------------------- Sự kiện

1. Soát vé (kiểm soát 🡪 Validation)
2. Không cho vào
3. Cho phép vào (Validation passed 🡪 cho vào)
4. Chỉnh sửa / thay đổi (data)

* Links document:

<https://www.youtube.com/watch?v=g4z5zwJMSuo> – Video F8

<https://expressjs.com/en/guide/writing-middleware.html>

1. Tham số next trong handler function (middleware):

* Next là một function, next được gọi khi ta cho phép pass middle và đi vào một middleware tiếp theo (các middleware được áp dụng lên một path cụ thể nào đó).
* Ví dụ:

Nhà ------------------🡪 Bảo vệ 1 (middleware) : Bảo vệ 2 : Sự kiện (yc: soát vé)

Nhà 🡨----------------- Sự kiện

Bác bảo vệ 1 cho pass gọi next để đi đến bác bảo vệ 2.

1. Một middleware trống không next cũng không res sẽ làm ứng dụng bị treo.
2. Cách áp dụng một middleware trong xuyên suốt các tuyến đường trong ứng dụng của mình thì phải làm thế nào?

* Cách apply / sử dụng middleware trong xuyên suốt dự án: app.use(express.json())