

Javascript Fullstack - Code Challenge

Create a REST API.

Prerequisites:

- Have a working MySQL 5.x (or greater) database server (a Docker container works too).
- Download and preload [datacharmer/test_db](#)

Challenge Description:

The `test_db` project contains historical fictional database. Create a new NodeJS project that connects to the `test_db` MySQL database. Create the following REST endpoints:

Back end:

- `/employeesExample` - This endpoint returns all records from the `employees` table that match the following criteria:
 - The employees were hired between Jan 1, 1990 and Jan 15,1990 (inclusive).
 - The response orders the records by last name (ascending).
- `/employee/${emp_no}` - This endpoint will return the data for the given employee number (Get method) that matches the database field (`emp_no`).
- `/employee/${emp_no}/titles` - Returns the matching titles for the given `emp_no` from the `titles` table, indicating the date periods and the title. The returned records must be ordered in descending chronological order.
- `/employee/add` - This endpoint receives a POST payload and inserts a new record in the `employees` table.
 - Please note YOUR server will decide the `emp_no` value (employee primary key) and **NOT** the MySQL engine nor the payload content.

Front End:

- A screen that displays the `/employeesExample` endpoint.
 - Pagination is optional
- Every record from the first screen has a link that displays a second screen with the employee details. These details include the employee details and the titles from the corresponding endpoints.

Constraints:

- All payloads in JSON(utf8)
- For the front end (if applies), you can use [React](#), [Angular](#) or [Vue](#).

Deliverables:

- GitHub link for NodeJS project.
- Unit tests
- `README.md` in the project containing the build/run instructions.
- Make sure your gitignore file is correctly set up to avoid junk files and folders that must not be committed (e.g. `.DS_STORE` and `node_modules/`)
- You can have the back end and front end in separate projects (folders) or in the same as long as they live in the same git repo.

Pointers:

- Get all the functionality working before prettifying your project

- Get your code as production ready as possible.