

Biometric Time Clock documentation - Yassir technical test

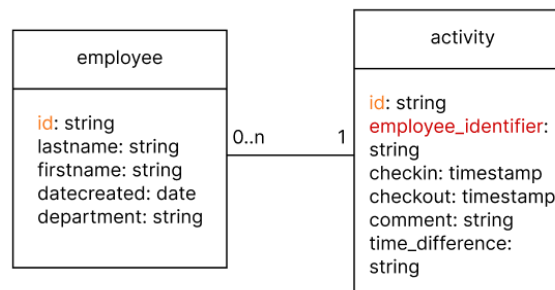
The Biometric Time Clock application is a new tool used by the ABC school to manage its employees. This project is the technical test of the Full-Stack Software Engineer position at Yassir.

The application has a client-server architecture where the server contains the requested API and the database connection built with NodeJS, while the client contains the front-end components with integration to the API using ReactJS.

The database used is postgresql for the need of a relational database only.

1. Database:

The database is called 'biometricTimeClock' and has 2 tables: employee to store the employee information and activity to store the checkin and checkout information, as shown in the following:



2. Server:

The server contains three files: `server.js` for the API endpoints, `pg.js` to initiate a connection with the database and the `db.sql` for the used sql queries to help create the database and the 2 tables.

The API routes are:

<code>"/newemployee/:lastname/:firstname/:department"</code>	To create a new employee by giving the lastname, firstname and department. The datecreated will be the date of the system given by javascript.
<code>"/getlist"</code>	Get a list of all employees with their information stored in the employee table.
<code>"/filterbydate/:creationdate"</code>	Filter by the date picked by the user on the client side.
<code>"/check-in/:employeeId/:comment"</code>	Checkin using the employeeId and save a comment.
<code>"/check-out/:employeeId/:comment"</code>	Checkout using the employeeId and save a new comment in the comment column of the activity table.

The API endpoints are documented in the code to explain how they work.

3. Client:

The client side has the app.js for the routes, the index.js to render the application, index.css to style the components, the components folder and the images folder.

The client is built in order to integrate and test the API endpoints using the fetch method. For example when getting the list of all employees in the list.js components:

```
const getEmployees = async() =>{
  try{
    fetch(`http://localhost:8000/getlist`).then(
      response => response.json()
    ).then(
      data => {
        setlistemployee(data)
      }
    )
  }catch (err) {
    console.error(err)
  }
}
```

The list component shows the list of all employees that are in the dataset (rendering a list of Employee components). While the new_employee component is used to create a new employee account with 3 inputs: the last name input, the first name input and the department input.

4. Test units:

A set of tests are conducted as follows:


4.1. Create a new employee:

Testing to create a new employee account with 'John' as last name, 'Lamversky' as first name and department as 'Department B'.



Create new employee account

NB: If the user tries to create a new employee without setting input values, an error message will be displayed.



ABC' school

Test de recrutement pour le poste de
full stack developer chez Yassir

Create new employee account

John


Lamversky

Department B

Create account


Activer Windows
Accédez aux paramètres pour activer Windows.

4.2. Get the list of all employees:
Here the created employees shows in the home page. However other employees were created to display a longer list.

ABC school 

List of employees

new employee

 Filter

Last name	First name	Department	Creation date
John	Lamversky	Department B	2023-10-31T23:00:00.000Z

Activer Windows
Accédez aux paramètres pour activer Windows.

Query

Query History

Scratch Pad

1

select * from employee;

Data Output

Messages

Notifications

	id [PK] character varying (255)	lastname character varying (255)	firstname character varying (255)	datecreation date	department character varying (255)
1	2c44403d-69f9-4e6b-b658-95e45a9a28...	John	Lamversky	2023-11-01	Department B
2	bkznpiefzef3243r15rg4r68f6e5df545df48	Camelia	Caramelle	2023-10-31	Department A
3	kgr5tg4354g684ref384834843848g83g4	William	Andigo	2023-11-01	Depatment A

Here is the new list of employees:

ABC school

List of employees

new employee

Filter

Last name	First name	Department	Creation date
John	Lamversky	Department B	2023-10-31T23:00:00.000Z
Camelia	Caramelle	Department A	2023-10-30T23:00:00.000Z
William	Andigo	Depatment A	2023-10-31T23:00:00.000Z

4.3. Filter by date of creation:

The user can filter employees by date of creation using a date picker.

If the date is not set and the filter button is pressed then the filter will take the system date to filter the employee data.

ABC school

List of employees

new employee

Filter

Last name	First name	Department	Creation date
John	Lamversky	Department B	2023-10-31T23:00:00.000Z
Camelia	Caramelle	Department A	2023-10-30T23:00:00.000Z
William	Andigo	Depatment A	2023-10-31T23:00:00.000Z

11/01/2023

November 2023

Su	Mo	Tu	We	Th	Fr	Sa
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	1	2

Example: Filter the employee accounts that were created on November 1st 2023:

ABC school

List of employees

new employee

Filter

Last name	First name	Department	Creation date
John	Lamversky	Department B	2023-10-31T23:00:00.000Z
William	Andigo	Depatment A	2023-10-31T23:00:00.000Z

Activer Windows
Accédez aux paramètres pour activer Windows.

NB: As you can see, the time shows a difference of 23h (T23:00:00) because of the different timezone used in the PgAdmin. However, the Filter API route gives a correct result, as it is only the rendering in the front end side that is not making time zone conversions.

4.4. Perform a check-in and a check-out:

This is a test for the checkin and checkout done in the pgAdmin directly since this part is not integrated in the client side. That is because the 2 API routes need the employeeeld parameter which is generally returned after authentication when login in.

For the check-in: The checkin route automatically inserts a new row in the activity table with the current system date in the checkin column.

```
insert into activity (id, employee_identifier, checkin, comment)
values ('dfcdfvdf', '1546546cdssssssdf', now(), 'Checkin comment');
select * from activity;
```

Output Messages Notifications

id	employee_identifier	checkin	checkout	comment	time_difference
[PK] character varying (255)	character varying (255)	timestamp without time zone	timestamp without time zone	character varying (255)	character varying (255)
dfcdfvdf	1546546cdssssssdf	2023-11-02 09:21:31.982268	[null]	Checkin comment	[null]

The checkout route updates the existing row data of the employee on the same day.

Query

Query History

Scratch Pad

1

update activity set checkout = now(), comment = 'Checkout comment' where employee_identifier =

2

select * from activity;

Data Output

Messages

Notifications

	id [PK] character varying (255)	employee_identifier character varying (255)	checkin timestamp without time zone	checkout timestamp without time zone	comment character varying (255)	time_difference character varying (255)
1	dfcdfvdf	1546546cdssssssdf	2023-11-02 09:21:31.982268	2023-11-02 09:23:24.056277	Checkout comment	[null]

4.5. Calculate the time difference between the checkin and checkout:

The time difference is then calculated and stored in the time_difference column.

Query

Query History

Scratch Pad

```
1 update activity set time_difference = (now() - (select checkin from activity
2 where employee_identifier = '1546546cdssssssdf' and
3 EXTRACT('MONTH' FROM checkin) = EXTRACT('MONTH' FROM now()) and
4 EXTRACT('DAY' FROM checkin) = EXTRACT('DAY' FROM now()) and
5 EXTRACT('YEAR' FROM checkin) = EXTRACT('YEAR' FROM now()));
6 select * from activity;
```

Data Output

Messages

Notifications

	id [PK] character varying (255)	employee_identifier character varying (255)	checkin timestamp without time zone	checkout timestamp without time zone	comment character varying (255)	time_difference character varying (255)
1	dfcdfvdf	1546546cdssssssdf	2023-11-02 09:21:31.982268	2023-11-02 09:23:24.056277	Checkout comment	00:04:38.945327

This query is in the checkout route as the time difference needs to be saved when checking out.

5. Deployment:

The project is available on GitHub.