

Back-end Node.js

=====

1. Simple Database querying

=====

Terdapat sebuah table "USER" yg memiliki 3 kolom: ID, UserName, Parent. Di mana:

Kolom ID adalah Primary Key

Kolom UserName adalah Nama User

Kolom Parent adalah ID dari User yang menjadi Creator untuk User tertentu.

eg.

ID	UserName	Parent
----	----------	--------

1	Ali	2
---	-----	---

2	Budi	0
---	------	---

3	Cecep	1
---	-------	---

Tuliskan SQL Query untuk mendapatkan data berisi:

ID	UserName	ParentUserName
----	----------	----------------

1	Ali	Budi
---	-----	------

2	Budi	NULL
---	------	------

3	Cecep	Ali
---	-------	-----

*Kolom ParentUserName adalah UserName berdasarkan value Parent

=====

2. Please write a small ExpressJS server to search movies from

<http://www.omdbapi.com/>

The Backend should :

- Have 2 endpoint named `"/search"` with GET method and `"/detail"` with GET method (single movie detail)

- Each API calls should be logged into a MySQL table, specifying DateTime, API endpoint getting called and the parameters passed

- Contain access credential to call the API as such :

Key : "faf7e5bb&s"

URL : <http://www.omdbapi.com/>

* Example url call to search is --> GET

<http://www.omdbapi.com/?apikey=faf7e5bb&s=Batman&page=2>

- Be written in ExpressJS framework

Important aspects :

- Readability of code

- Good display on the knowledge of "Separation of Concerns for Codes"

- Write unit tests on some of the important files. Bigger plus points for complete unit test cases

- Good use of asynchronousy

Plus points:

- Deploy your result to a public URL so we can check the result

- Implementation of Clean Architecture is a BIG plus

- Complete Unit tests

=====

3. Please refactor the code below to make it more concise, efficient and readable with good logic flow.

```
function findFirstStringInBracket(str){
  if(str.length > 0){
    let indexFirstBracketFound = str.indexOf("(");
    if(indexFirstBracketFound >= 0){
      let wordsAfterFirstBracket = str.substr( indexFirstBracketFound );
      if(wordsAfterFirstBracket){
        wordsAfterFirstBracket = wordsAfterFirstBracket.substr(1);
        let indexClosingBracketFound = wordsAfterFirstBracket.indexOf(")");
        if(indexClosingBracketFound >= 0){
          return wordsAfterFirstBracket.substring(0,
indexClosingBracketFound);
        }
      }
    }
  }
}
```

```
[
  ["kita", "atik", "tika"],
  ["aku", "kua"],
  ["makan"],
  ["kia"]
]
```