

postgreSQL Public

JavaScript Updated 7 hours ago

詳細コードは上記のリポジトリ確認

公式HP
<https://www.postgresql.org/>

インストール時のpw

password

password

PORT : 54

☆ Star

SQL文でDBを作成する方法

まず SQLshellを開く



SQL Shell (psql)

- ・ 現存するDB一覧を表示

⇒ 「¥|」

- ・ 「users」という名前のDBを作成する

⇒ 「create database users;」

- ・ 「users」というDBの中に移動する。

⇒ 「¥c users」

- ・ テーブルを作成

⇒ 「create table users(
ID serial primary key,
name varchar(255),
email varchar(255),
age int);」

- ・ DBの情報を確認

⇒ 「¥dt」

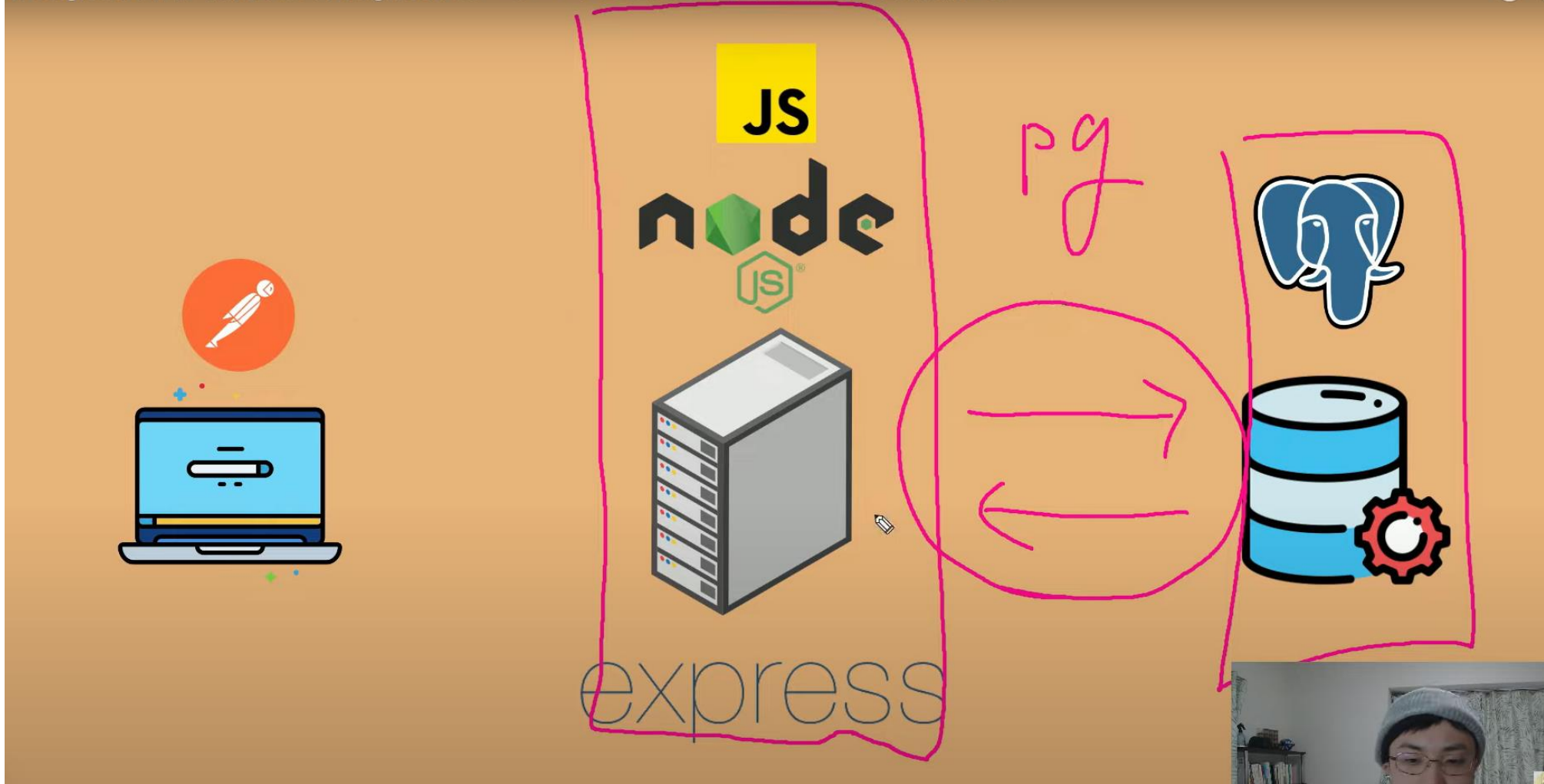
- ・ usersの中にデータを入れ

⇒ 「insert into users (name, email, age)」

⇒ 「values ('shincode', 'shincode@gmail.com', 25), ('testUser', 'test@gmail.com', 34);」

- ・ DBの中身すべてを確認

⇒ 「select * from users;」



Pgライブラリーを利用して、バックエンドからpostgreSQLにアクセスする

```
$ npm install pg
```

Pg HP : <https://node-postgres.com/>

Server側で使い方

◆初期設定

```
js db.js > [🔗] pool
1  const Pool = require("pg").Pool;
2
3  const pool = new Pool({
4    user: "postgres",
5    host: "localhost",
6    database: "users",
7    password: "",
8    port: 5432,
9  });
10
11  module.exports = pool;
12
```

◆server側のAPI設計例

```
//ユーザー情報をすべて取得する
app.get("/users", (req, res) => {
  pool.query("select * from users", (error, results) => {
    if (error) throw error;
    // console.log("全権取得できました。");
    return res.status(200).json(results.rows);
  })
});

//特定のユーザーを取得する。
app.get("/users/:id", (req, res) => {
  const id = req.params.id;
  pool.query("select * from users where id = $1", [id], (error, results) => {
    if (error) throw error;
    return res.status(200).json(results.rows);
  })
});

//ユーザーを追加する
app.post("/users", (req, res) => {
  const { name, email, age } = req.body;
  //ユーザーがすでに存在しているかを確認する
  pool.query("select s from users s where s.email = $1", [email], (error, results) => {
    if (results.rows.length) {
      return res.send("すでにユーザーが存在します");
    }
    pool.query("insert into users (name,email,age) values($1,$2,$3)", [name, email, age],
      (error, results) => {
        if (error) throw error;
        res.status(201).send("ユーザー作成に成功しました");
      })
  })
});
```

```
//ユーザーを削除する。
app.delete("/users/:id", (req, res) => {
  const id = req.params.id;

  pool.query("select * from users where id = $1", [id], (error, results) => {
    if (error) throw error;
    const userExisting = results.rows.length;

    if (!userExisting) res.send("ユーザーが存在しません。");

    pool.query("delete from users where id = $1", [id], (error, results) => {
      if (error) throw error;
      return res.status(200).send("削除に成功しました");
    });
  })
});

//ユーザーを更新する。
app.put("/users/:id", (req, res) => {
  const id = req.params.id;
  const name = req.body.name;

  pool.query("select * from users where id = $1", [id], (error, results) => {
    if (error) throw error;
    const userExisting = results.rows.length;

    if (!userExisting) res.send("ユーザーが存在しません。");

    pool.query("update users set name = $1 where id = $2", [name, id], (error, results) => {
      if (error) throw error;
      return res.status(200).send("更新に成功しました");
    });
  })
});
```