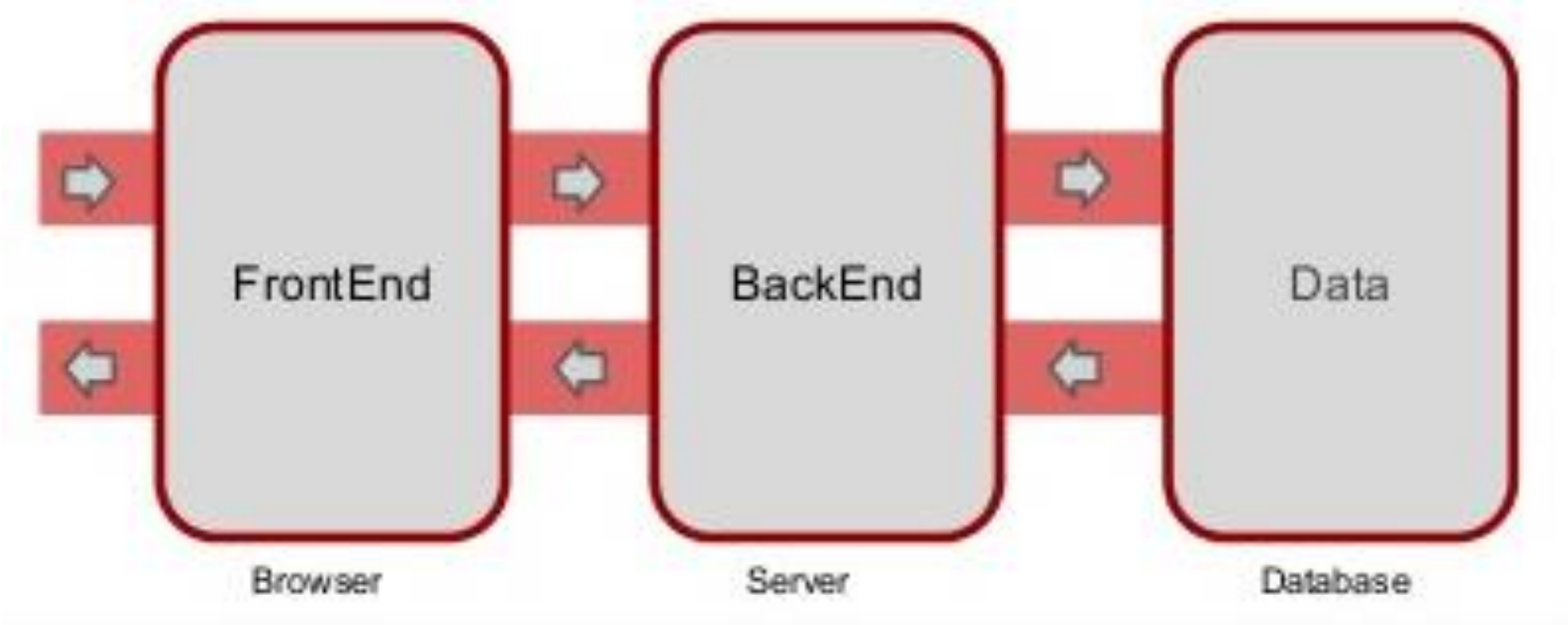
Back-End Development

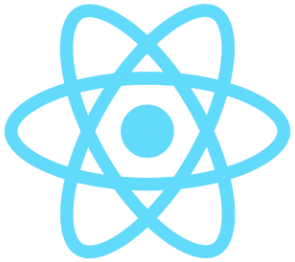
Express & PostgreSQL

#13d Using Sequelize

Purwadhika
Startup and Coding School

Web App Architecture





#1 Activate Server

- Open terminal/command prompt

```
$ cd C:\Program Files\PostgreSQL\10\bin
$ psql -U postgres
```

//password = 12345

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\usr>cd C:\Program Files\PostgreSQL\10\bin

C:\Program Files\PostgreSQL\10\bin>psql -U postgres
psql (10.1)
WARNING: Console code page (437) differs from Windows code page (1252)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
Type "help" for help.

postgres=# \l
```

#2 Create Database

Menampilkan daftar database:

```
$ \l
```

Membuat database “dojo”:

```
$ CREATE DATABASE dojo;
```

Terhubung ke database “dojo”:

```
$ \c dojo
```

Menampilkan daftar table di “dojo”:

```
$ \d
```

*Tabel akan dibuat via Sequelize!

Purwadhika
Startup and Coding School

Sequelize

- ***Sequelize*** is a promise-based ORM for Node.js v4 and up. It supports the dialects of PostgreSQL, MySQL, SQLite, MSSQL and features solid transaction support, relations, read replication and more.
- More info: *docs.sequelizejs.com*

Purwadhika
Startup and Coding School

#3 Setup Sequelize

- On project dir, install Sequelize package:

```
$ npm install sequelize --save
```

- Then install one of Sequelize connector:

```
$ npm install --save pg pg-hstore //PostgreSQL
$ npm install --save mysql2       //MySQL
$ npm install --save sqlite3      //SQLite
$ npm install --save tedious      //MSSQL
```

Purwadhika
Startup and Coding School

Using Sequelize Package

#1 Connect to PostgreSQL

```
const url =
'postgres://postgres:12345@localhost:5432/dojo';

const Sequelize = require('sequelize');
const sequelize = new Sequelize(url);

sequelize
  .authenticate()
  .then(() => {
      console.log('Sukses terhubung!');
   })
  .catch(err => {
      console.error('Gagal terhubung:', err);
  });
```

Purwadhika
Startup and Coding School

```
* Write it after connection code!
. . . . . .
// Create model
const Ninja = sequelize.define('ninja', {
    nama: {type: Sequelize.STRING},
    usia: {type: Sequelize.INTEGER}
});

// Create table
// force:true will drop the table if it already exists
Ninja.sync({force: false}).then(() => {
    console.log('Tabel dibuat!')
});
```

Using Sequelize

#2 Create Model & Table

```
dojo=# \d
             List of relations
 Schema |     Name      |   Type   |  Owner
--------+---------------+----------+----------
 public | ninjas        | table    | postgres
 public | ninjas_id_seq | sequence | postgres
(2 rows)
```

Using Sequelize Package

#3 Insert data

** Write it after creating model & table code!*

. . . . . .

```
Ninja.create({
    nama: 'Andi',
    usia: 23
}).then(data =>
  console.log('Data masuk!')
);
```

```
dojo=# select * from ninjas;
 id | nama | usia |          createdAt          |          updatedAt
----+------+------+-----------------------------+-----------------------------
  1 | Andi |   23 | 2018-04-04 09:43:12.559+07 | 2018-04-04 09:43:12.559+07
(1 row)
```

Using Sequelize Package

#4 Get data

** Write it after creating model & table code!*

```
  . . . . . .

Ninja.findAll().then(data => {
   console.log(data.map((val,i)=>{
      var data = val.dataValues;
      return data;
   }))
})
```

Purwadhika
Startup and Coding School

Express & PostgreSQL

express

PostgreSQL

Purwadhika
Startup and Coding School

Express & ElephantSQL

express

Purwadhika
Startup and Coding School