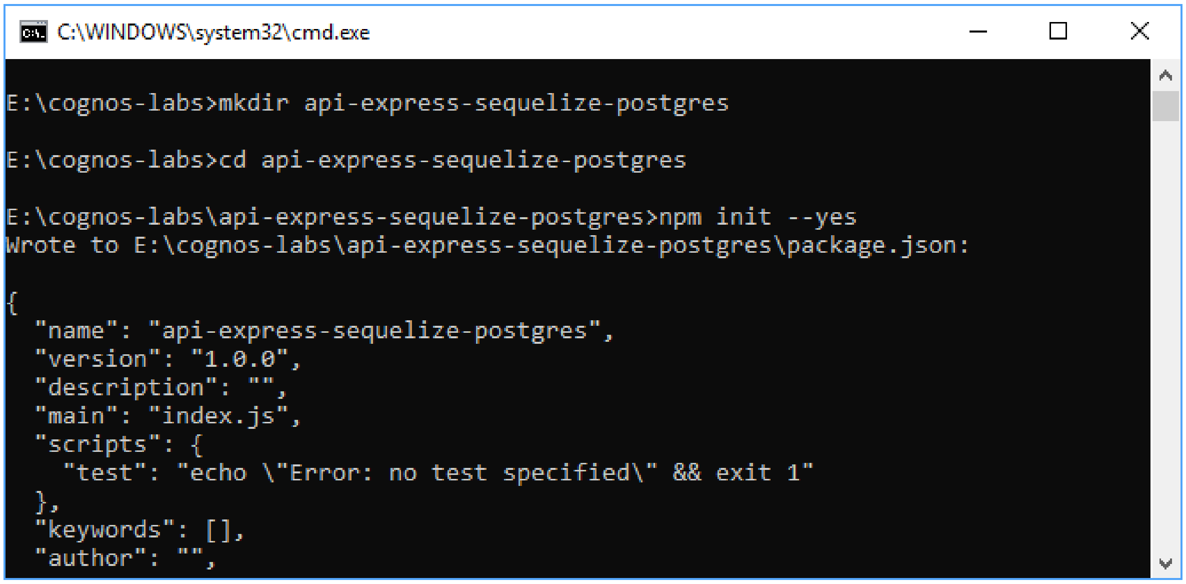
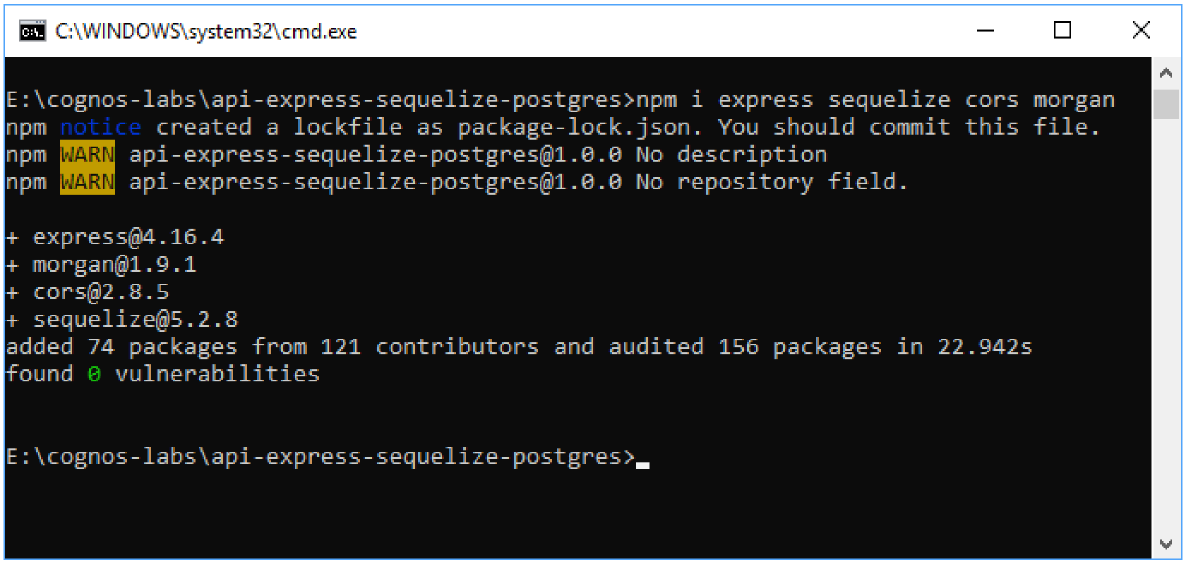
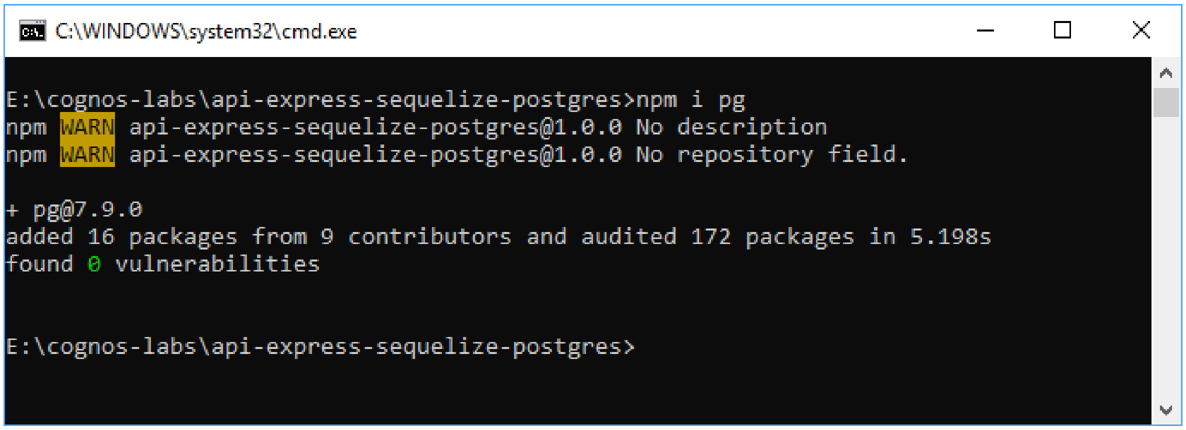


```
E:\cognos-labs>mkdir api-express-sequelize-postgres

E:\cognos-labs>cd api-express-sequelize-postgres

E:\cognos-labs\api-express-sequelize-postgres>npm init --yes
Wrote to E:\cognos-labs\api-express-sequelize-postgres\package.json:

{
  "name": "api-express-sequelize-postgres",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
```

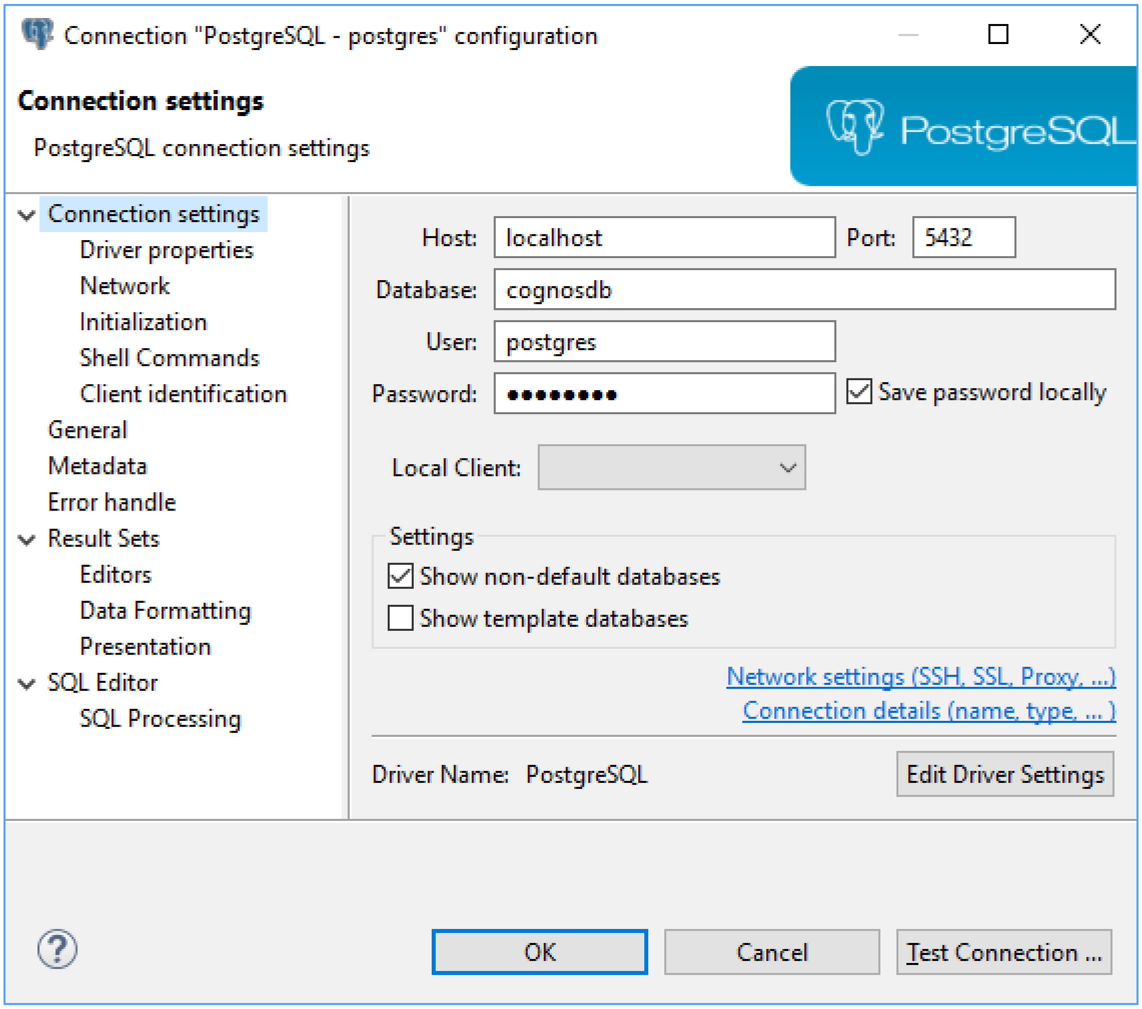
npm install pg sequelize



```
E:\cognos-labs\api-express-sequelize-postgres>npm i express sequelize cors morgan
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN api-express-sequelize-postgres@1.0.0 No description
npm WARN api-express-sequelize-postgres@1.0.0 No repository field.

+ express@4.16.4
+ morgan@1.9.1
+ cors@2.8.5
+ sequelize@5.2.8
added 74 packages from 121 contributors and audited 156 packages in 22.942s
found 0 vulnerabilities


E:\cognos-labs\api-express-sequelize-postgres>_
```



```
E:\cognos-labs\api-express-sequelize-postgres>npm i pg
npm WARN api-express-sequelize-postgres@1.0.0 No description
npm WARN api-express-sequelize-postgres@1.0.0 No repository field.

+ pg@7.9.0
added 16 packages from 9 contributors and audited 172 packages in 5.198s
found 0 vulnerabilities


E:\cognos-labs\api-express-sequelize-postgres>
```



Connection "PostgreSQL - postgres" configuration

Connection settings

PostgreSQL connection settings

Host: localhost Port: 5432

Database: cognosdb

User: postgres

Password: ●●●●●●●●● ☑ Save password locally

Local Client:

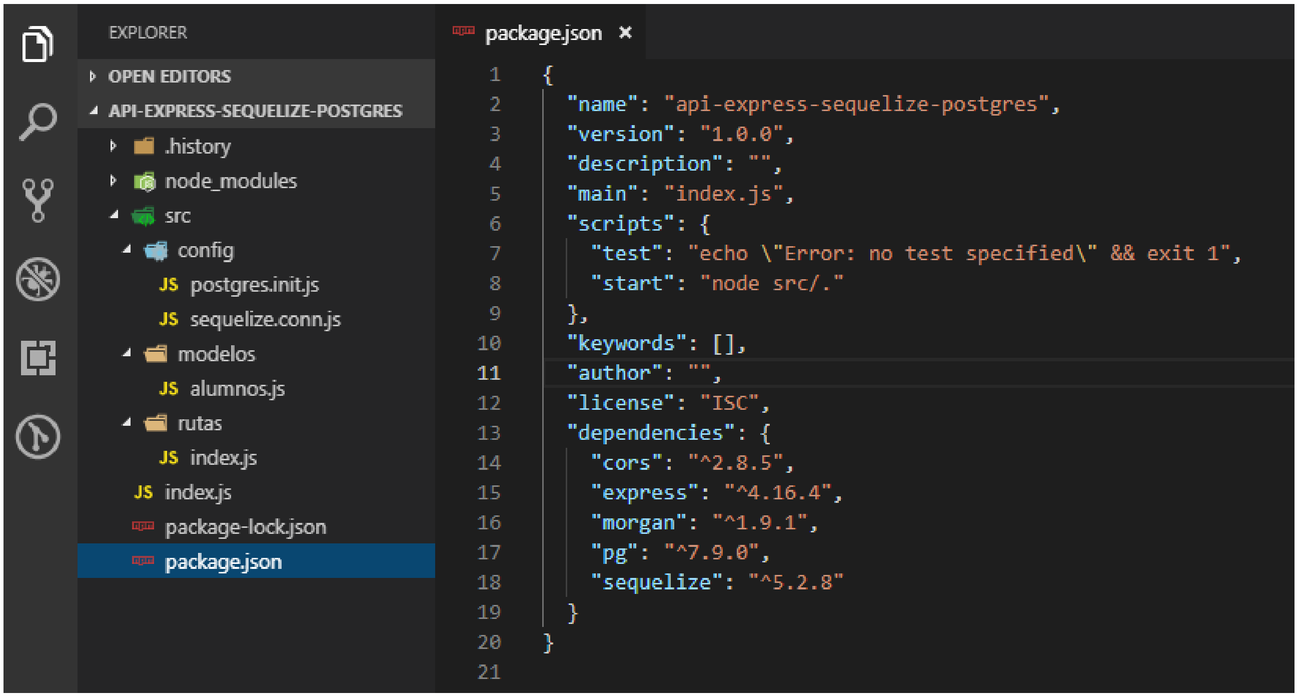
Settings

☑ Show non-default databases

☐ Show template databases

Network settings (SSH, SSL, Proxy, ...)
Connection details (name, type, ...)

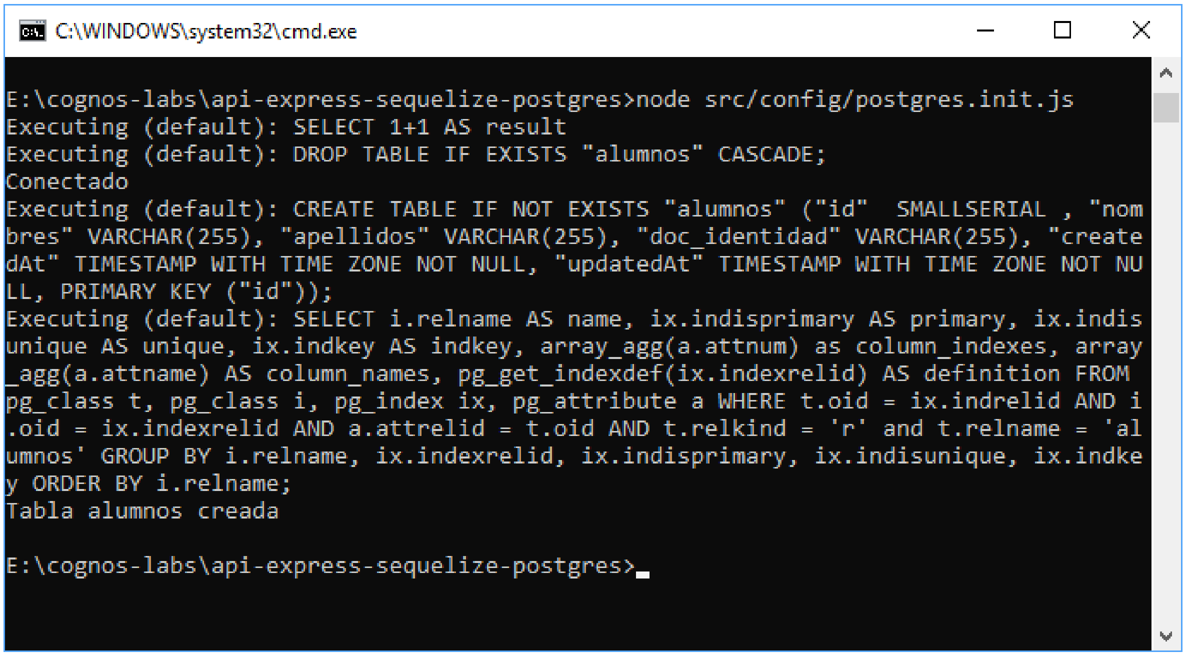
Driver Name: PostgreSQL



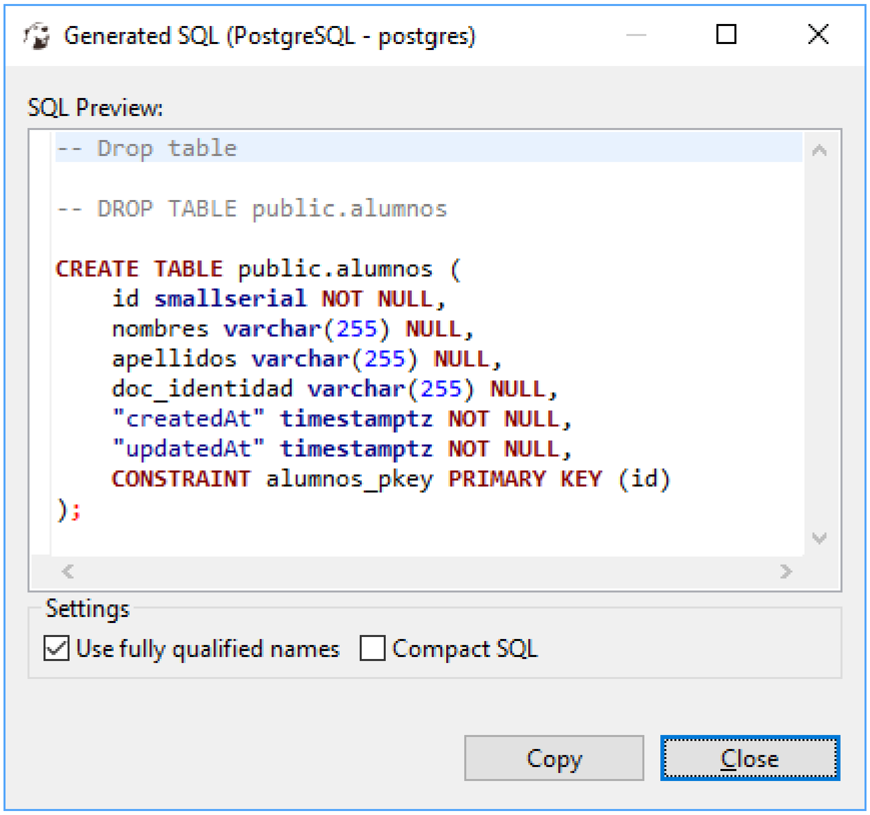
```json
{
  "name": "api-express-sequelize-postgres",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "node src/."
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "cors": "^2.8.5",
    "express": "^4.16.4",
    "morgan": "^1.9.1",
    "pg": "^7.9.0",
    "sequelize": "^5.2.8"
  }
}
```

/src/config/postgres.init.js

```
const Sequelize = require('sequelize')

const sequelize = new Sequelize('cognosdb', 'postgres', 'postgres', {
  host: 'localhost',
  dialect: 'postgres',
})

sequelize.authenticate()
  .then(() => {
    console.log('Conectado')
  })
  .catch(err => {
    console.log('No se conecto')
});

const Alumnos = sequelize.define('alumnos', {
    id: {type: Sequelize.SMALLINT, primaryKey: true, autoIncrement: true},
    nombres: Sequelize.STRING,
    apellidos: Sequelize.STRING,
    doc_identidad: Sequelize.STRING
  });

  Alumnos.sync({ force: true }).then(() => {
    console.log('Tabla alumnos creada');
    return true;
  });

module.exports = sequelize;
```



```
E:\cognos-labs\api-express-sequelize-postgres>node src/config/postgres.init.js
Executing (default): SELECT 1+1 AS result
Executing (default): DROP TABLE IF EXISTS "alumnos" CASCADE;
Conectado
Executing (default): CREATE TABLE IF NOT EXISTS "alumnos" ("id"  SMALLSERIAL , "nom
bres" VARCHAR(255), "apellidos" VARCHAR(255), "doc_identidad" VARCHAR(255), "create
dAt" TIMESTAMP WITH TIME ZONE NOT NULL, "updatedAt" TIMESTAMP WITH TIME ZONE NOT NU
LL, PRIMARY KEY ("id"));
Executing (default): SELECT i.relname AS name, ix.indisprimary AS primary, ix.indis
unique AS unique, ix.indkey AS indkey, array_agg(a.attnum) as column_indexes, array
_agg(a.attname) AS column_names, pg_get_indexdef(ix.indexrelid) AS definition FROM
pg_class t, pg_class i, pg_index ix, pg_attribute a WHERE t.oid = ix.indrelid AND i
.oid = ix.indexrelid AND a.attrelid = t.oid AND t.relkind = 'r' and t.relname = 'al
umnos' GROUP BY i.relname, ix.indexrelid, ix.indisprimary, ix.indisunique, ix.indke
y ORDER BY i.relname;
Tabla alumnos creada

E:\cognos-labs\api-express-sequelize-postgres>_
```

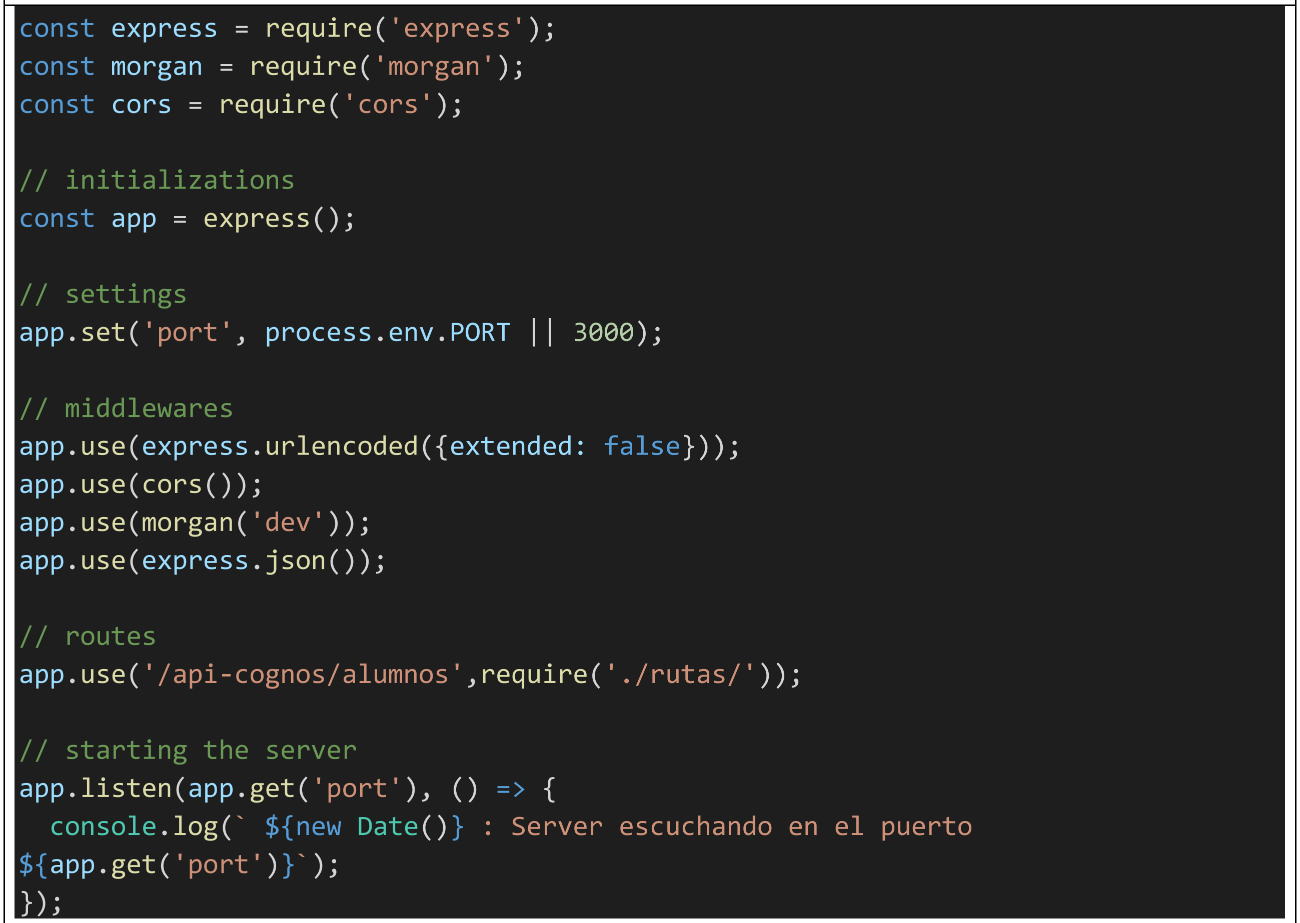


```
-- Drop table

-- DROP TABLE public.alumnos

CREATE TABLE public.alumnos (
    id smallserial NOT NULL,
    nombres varchar(255) NULL,
    apellidos varchar(255) NULL,
    doc_identidad varchar(255) NULL,
    "createdAt" timestamptz NOT NULL,
    "updatedAt" timestamptz NOT NULL,
    CONSTRAINT alumnos_pkey PRIMARY KEY (id)
);
```

Settings

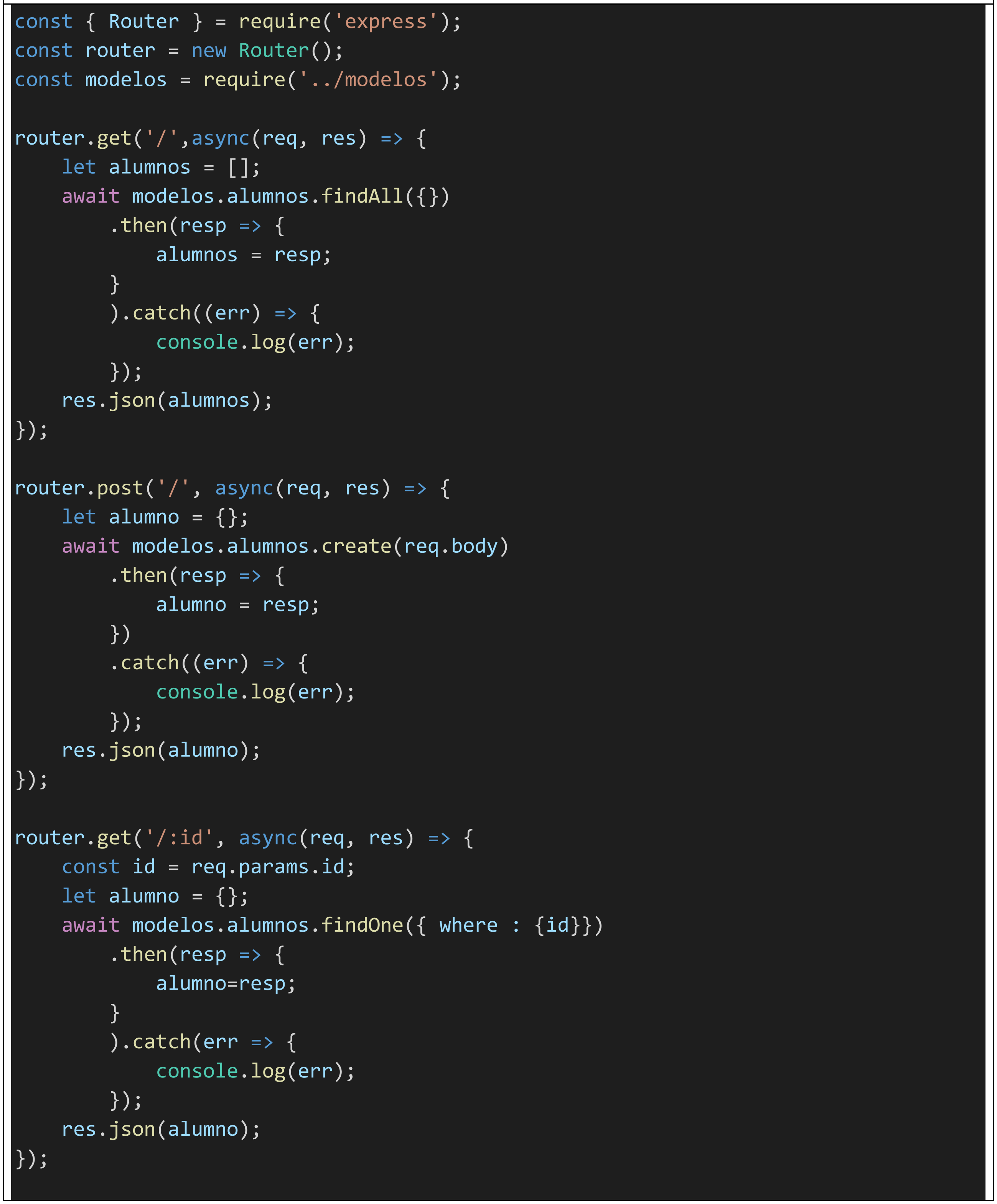
☑ Use fully qualified names □ Compact SQL

/src/index.js

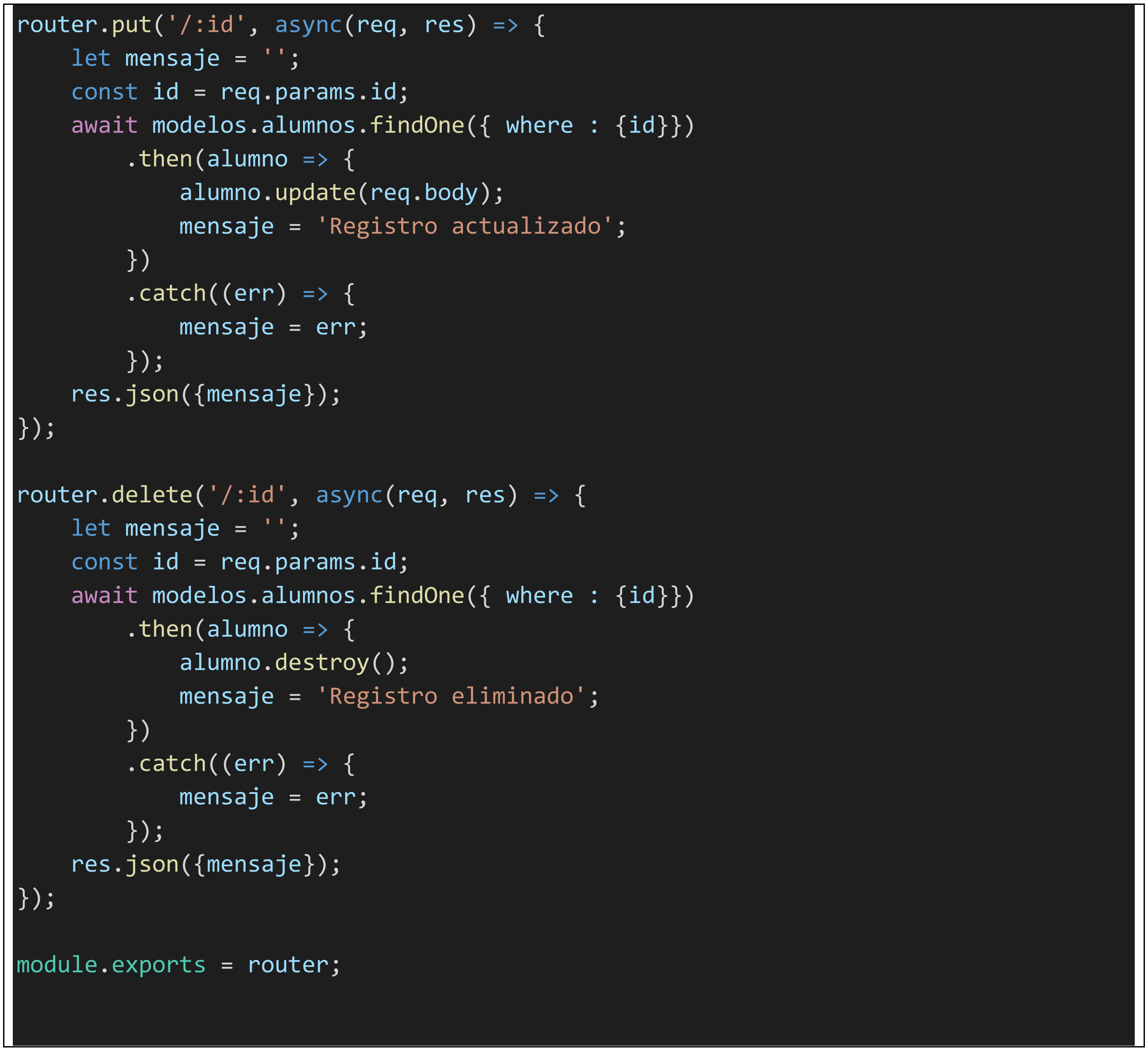


```
const express = require('express');
const morgan = require('morgan');
const cors = require('cors');

// initializations
const app = express();

// settings
app.set('port', process.env.PORT || 3000);

// middlewares
app.use(express.urlencoded({extended: false}));
app.use(cors());
app.use(morgan('dev'));
app.use(express.json());

// routes
app.use('/api-cognos/alumnos',require('./rutas/'));

// starting the server
app.listen(app.get('port'), () => {
  console.log(` ${new Date()} : Server escuchando en el puerto
${app.get('port')}`);
});
```

/src/rutas/index.js

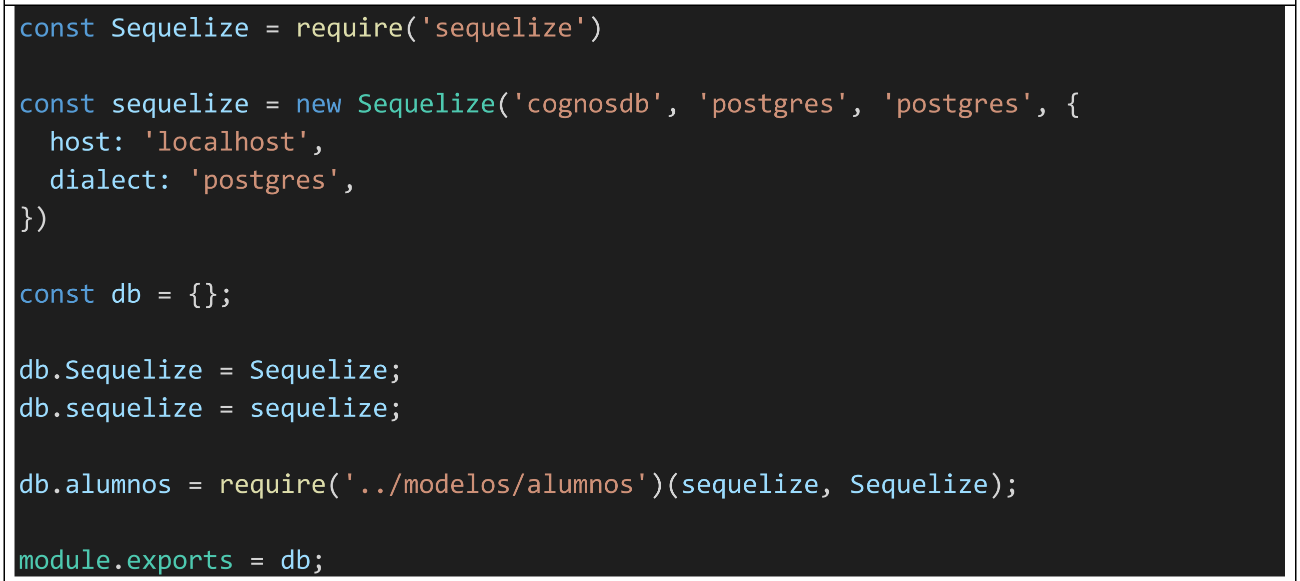


```
const { Router } = require('express');
const router = new Router();
const modelos = require('../modelos');

router.get('/',async(req, res) => {
    let alumnos = [];
    await modelos.alumnos.findAll({})
        .then(resp => {
            alumnos = resp;
        }
        ).catch((err) => {
            console.log(err);
        });
    res.json(alumnos);
});

router.post('/', async(req, res) => {
    let alumno = {};
    await modelos.alumnos.create(req.body)
        .then(resp => {
            alumno = resp;
        })
        .catch((err) => {
            console.log(err);
        });
    res.json(alumno);
});

router.get('/:id', async(req, res) => {
    const id = req.params.id;
    let alumno = {};
    await modelos.alumnos.findOne({ where : {id}})
        .then(resp => {
            alumno=resp;
        }
        ).catch(err => {
            console.log(err);
        });
    res.json(alumno);
});
```



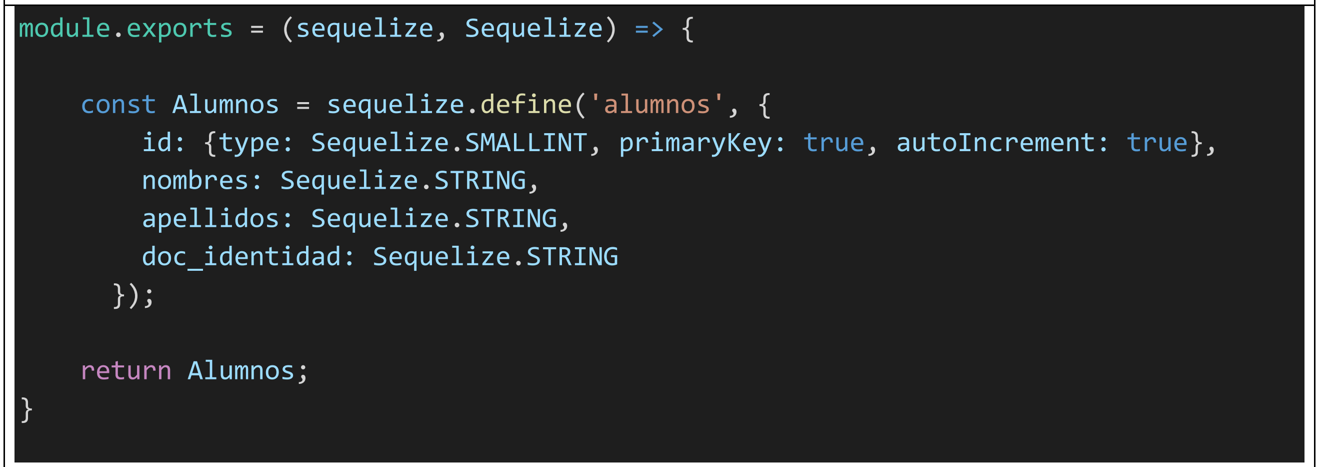
```
router.put('/:id', async(req, res) => {
    let mensaje = '';
    const id = req.params.id;
    await modelos.alumnos.findOne({ where : {id}})
        .then(alumno => {
            alumno.update(req.body);
            mensaje = 'Registro actualizado';
        })
        .catch((err) => {
            mensaje = err;
        });
    res.json({mensaje});
});

router.delete('/:id', async(req, res) => {
    let mensaje = '';
    const id = req.params.id;
    await modelos.alumnos.findOne({ where : {id}})
        .then(alumno => {
            alumno.destroy();
            mensaje = 'Registro eliminado';
        })
        .catch((err) => {
            mensaje = err;
        });
    res.json({mensaje});
});

module.exports = router;
```

/src/modelos/index.js

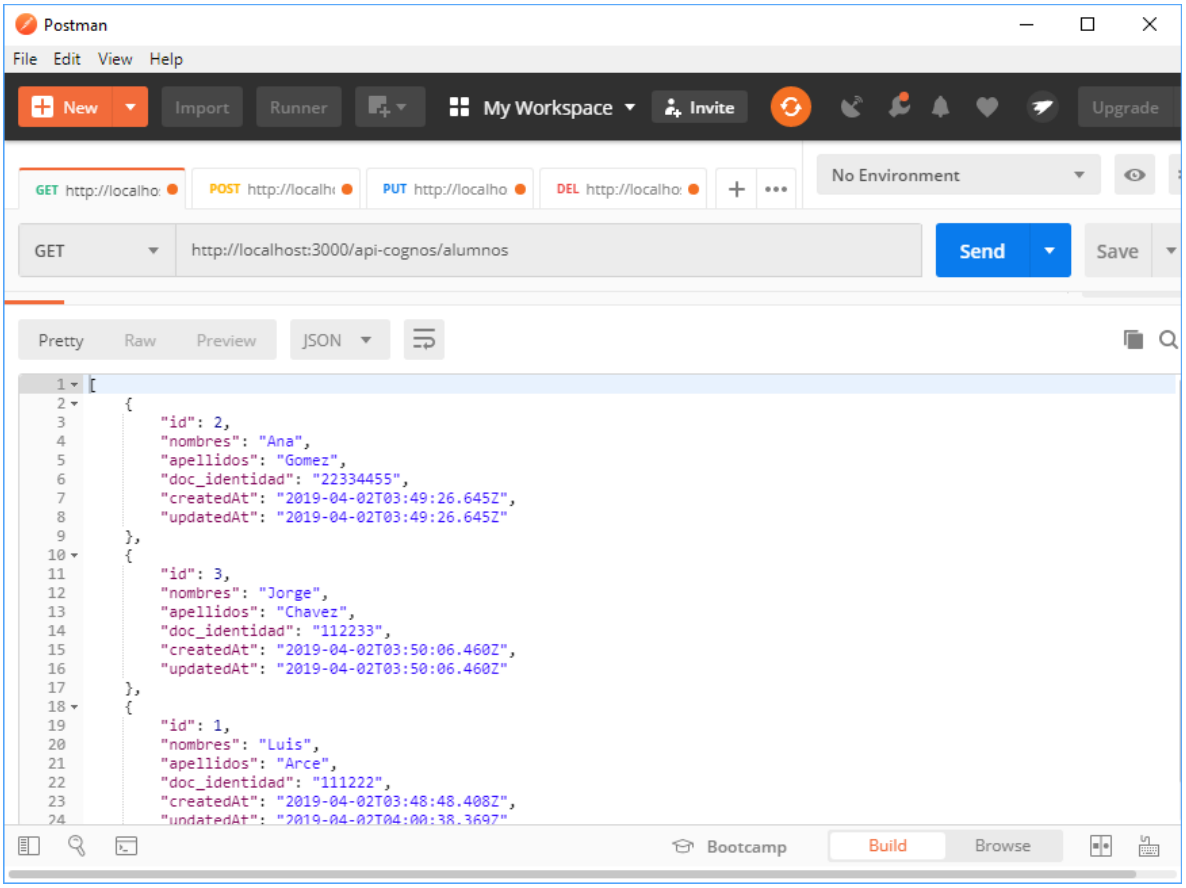


```
const Sequelize = require('sequelize')

const sequelize = new Sequelize('cognosdb', 'postgres', 'postgres', {
  host: 'localhost',
  dialect: 'postgres',
})

const db = {};

db.Sequelize = Sequelize;
db.sequelize = sequelize;

db.alumnos = require('../modelos/alumnos')(sequelize, Sequelize);

module.exports = db;
```

/src/modelos/alumnos.js



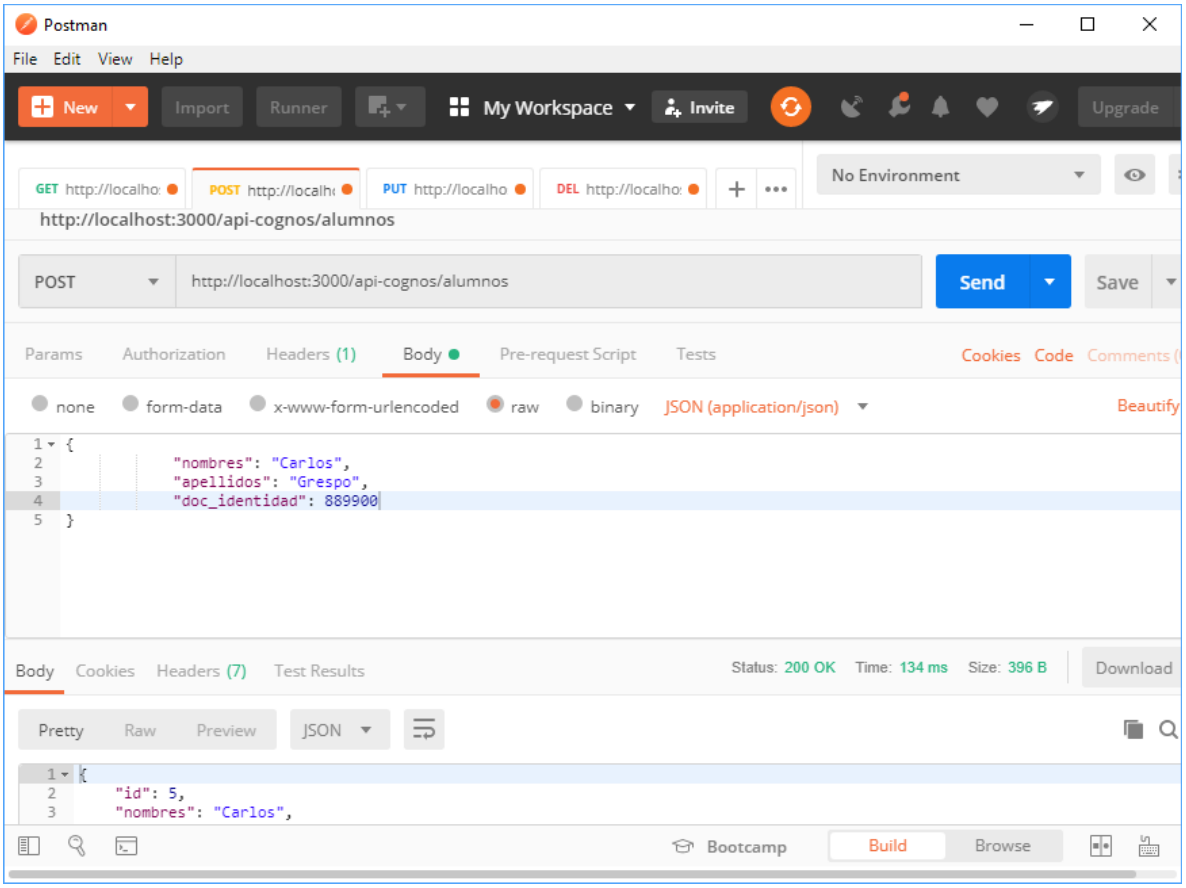
```
module.exports = (sequelize, Sequelize) => {

    const Alumnos = sequelize.define('alumnos', {
        id: {type: Sequelize.SMALLINT, primaryKey: true, autoIncrement: true},
        nombres: Sequelize.STRING,
        apellidos: Sequelize.STRING,
        doc_identidad: Sequelize.STRING
      });

    return Alumnos;
}
```



No Environment

GET http://localhost:3000/api-cognos/alumnos

```
[
    {
        "id": 2,
        "nombres": "Ana",
        "apellidos": "Gomez",
        "doc_identidad": "22334455",
        "createdAt": "2019-04-02T03:49:26.645Z",
        "updatedAt": "2019-04-02T03:49:26.645Z"
    },
    {
        "id": 3,
        "nombres": "Jorge",
        "apellidos": "Chavez",
        "doc_identidad": "112233",
        "createdAt": "2019-04-02T03:50:06.460Z",
        "updatedAt": "2019-04-02T03:50:06.460Z"
    },
    {
        "id": 1,
        "nombres": "Luis",
        "apellidos": "Arce",
        "doc_identidad": "111222",
        "createdAt": "2019-04-02T03:48:48.408Z",
        "updatedAt": "2019-04-02T04:00:38.369Z"
```



http://localhost:3000/api-cognos/alumnos

POST http://localhost:3000/api-cognos/alumnos

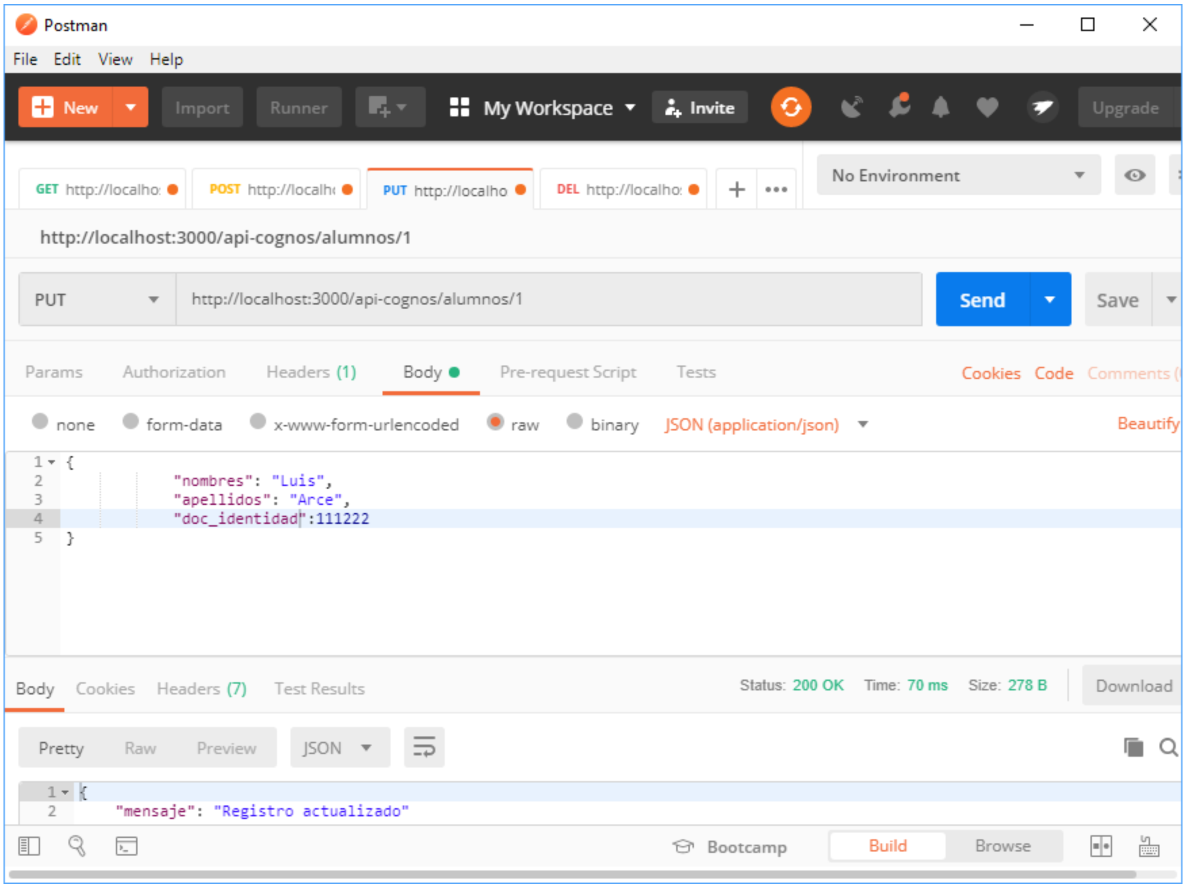
Body: ○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary JSON (application/json)

```
{
        "nombres": "Carlos",
        "apellidos": "Grespo",
        "doc_identidad": 889900
}
```

Status: 200 OK Time: 134 ms Size: 396 B

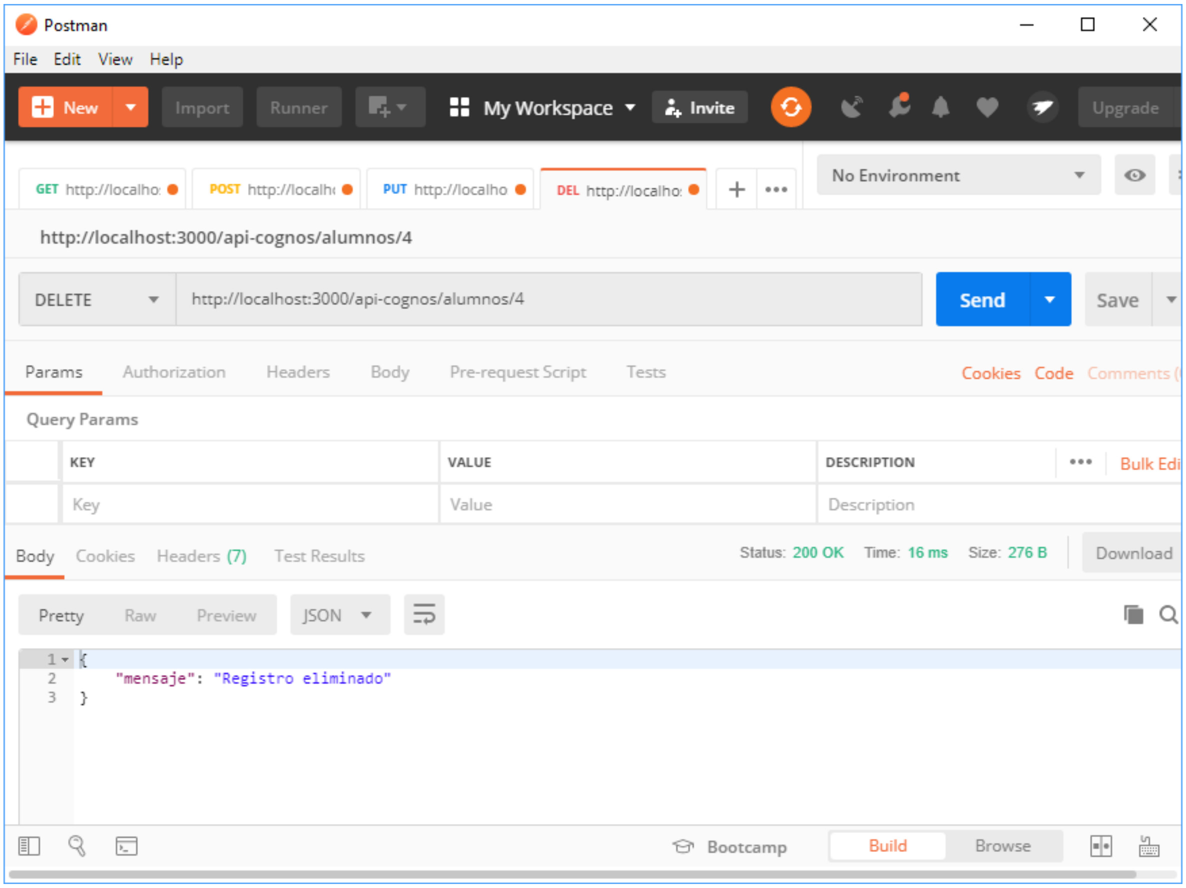
Body JSON

```
{
    "id": 5,
    "nombres": "Carlos",
```



```
{
        "nombres": "Luis",
        "apellidos": "Arce",
        "doc_identidad":111222
}
```

```
{
    "mensaje": "Registro actualizado"
```



http://localhost:3000/api-cognos/alumnos/4

DELETE http://localhost:3000/api-cognos/alumnos/4

Send
Save
Params Authorization Headers Body Pre-request Script Tests
Cookies Code Comments
Query Params
KEY VALUE DESCRIPTION
Bulk Edi
Key Value Description
Body Cookies Headers (7) Test Results

Status: 200 OK Time: 16 ms Size: 276 B

```
{
    "mensaje": "Registro eliminado"
}
```