

Solution 2

Software components of Blockstarter 1.0

- Models
- DAO
- Controllers
- Routes

Models

- *models/project.js*
- *models/creator.js*
- *models/backer.js*

Model example: backer.js

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

module.exports = global.db.define('backer', {
  name: {
    type: Sequelize.STRING
  },
  balance: {
    type: Sequelize.INTEGER
  }
}, {
  timestamps: false
});
```

Model example: SQL

```
CREATE TABLE backers(  
    id SERIAL UNIQUE PRIMARY KEY NOT NULL,  
    balance INT,  
    name TEXT  
);
```

DAO: *dao/postgresql.js*

```
module.exports = class DAO {  
  constructor(Model) {  
    this.Model = Model;  
  }  
  create(model) { /* ... */ }  
  get(query) { /* ... */ }  
  getAll() { /* ... */ }  
  update(query, body) { /* ... */ }  
  remove(query) { /* ... */ }  
}
```

DAO: *dao/postgresql.js*

```
module.exports = class DAO {  
  constructor(Model) {  
    this.Model = Model;  
  }  
  create(model) {  
    return new Promise((resolve, reject) => {  
      this.Model.create(model).then(data =>  
        resolve(data)).catch(err => reject(err));  
    });  
  }  
  // ...  
}
```

Controllers

- *controllers/_controller.js*
- *controllers/project.js*
- *controllers/creator.js*
- *controllers/backer.js*

Basic CRUD Controller: *_controller.js*

```
const express = require('express');
const DAO = require('../dao/postgresql');

module.exports = class Controller {
  constructor(Model) {
    this.Model = Model;
    this.dao = new DAO(Model);
  }

  withCreate() {}
  withGet(id) {}
  withGetAll() {}
  withUpdate() {}
  withRemove() {}

  router() {}
}
```


Basic CRUD Controller: *_controller.js*

```
// helper functions  
let handleErr = (err, res) => {  
  console.error(err);  
  res.status(500).send(err);  
};  
  
let handleRes = (data, res, statusCode) => {  
  s = statusCode || 200;  
  res.status(s).send(data);  
};
```

Basic CRUD Controller: *_controller.js*

```
// ...  
  
module.exports = class Controller {  
  // ...  
  
  withCreate() {  
    this.create = (req, res, next) => {  
      let model = req.body;  
      this.dao.create(model)  
        .then(data => handleRes(data, res))  
        .catch(err => handleErr(err, res));  
    }  
    return this;  
  }  
  
  // ...  
}
```

Project Controller: *project.js*

```
const Controller = require('./_controller');  
const Project = require('../models/project');  
  
const ctrl = new Controller(Project).withCreate();
```

Basic CRUD Controller: *_controller.js*

```
// ...  
  
module.exports = class Controller {  
  // ...  
  withCreate() { /* ... */ }  
  withGet(id) { /* ... */ }  
  // ...  
  
  router() {  
    const router = express.Router();  
    if (this.create) {  
      router.post('/', this.create.bind(this));  
    }  
    if (this.get) {  
      router.get('/:id', this.get.bind(this));  
    }  
    // ...  
    return router;  
  }  
}
```

Project Controller: *project.js*

```
const Controller = require('./_controller');
const Project = require('../models/project');

const router = new Controller(Project).withCreate()
  .withGet().withGetAll().withUpdate().withRemove()
  .router();

module.exports = router;
```

app.js

```
// ...  
  
// Import controllers (routes)  
const projects = require('./controllers/projects');  
  
// Set up express routes  
app.use('/projects', projects);  
  
// ...
```