

# Grails with React and Vue

by: [Firnaz Lutzian Adiansyah](#)

**Brief description:** This project is to showcase how grails can be used alongside the two most popular front end frameworks React and Vue.js by building simple CRUD application.

## Setup

Before diving in into the project it is **important** to check that these following are installed in your machine

- grails
- java sdk

```
# check by
grails -v

# if they are already installed it should return (version may vary)
| Grails Version: 4.0.2
| JVM Version: 1.8.0_241
```

Note: Grails Version 3 and Java 8 is used for the test

These repositories are consisted of grails application with React and Vue

```
# clone this repository
git clone https://github.com/firnazlutzian/grails-app.git

# to navigate to react
cd react-grails

# to navigate to vue
cd vue-grails
```

Inside the directory there will be 2 main folder which are the **server** and **client** folder. You will need to start the `server` and `client` applications separately by opening 2 separate terminals.

```
# to run the back end (grails) will open http://localhost:8080/
./gradlew server:bootRun

# to run the front end (React/Vue) will open http://localhost:3000/
./gradlew client:start

# or for React (preferable method) you may navigate to the client folder and
run it there by using command
cd client
yarn start

# or for NPM
npm start
```

## API

Note: if you are experiencing cors issue in your side please contact me or install some chrome plugin to resolve the issue. Otherwise, you are encouraged to test the API with a supported tools like:

[postman](#) or [insomnia](#)

### API for listing license, bid, and driver

GET `http://localhost:8080/license`

GET `http://localhost:8080/bid`

GET `http://localhost:8080/driver`

### API for posting license, bid, and driver

POST `http://localhost:8080/license`

```
{
  "classType": "CLASS_A",
  "licenseNum": 12345678,
  "bid": {
    "date": "2020-01-01",
    "amount": 5000
  },
  "driver": {
    "id": 1
  }
}
```

POST `http://localhost:8080/bid`

```
{
  "date": "2020-01-01",
  "amount": 9999,
}
```

POST `http://localhost:8080/driver`

```
{
  "name": "Firnaz"
}
```

### API for updating license, bid, and driver

PATCH `http://localhost:8080/license/${id}`

PATCH `http://localhost:8080/bid/${id}`

PATCH `http://localhost:8080/driver/${id}`

### API for deleting license, bid, and driver

DELETE `http://localhost:8080/license/${id}`

DELETE `http://localhost:8080/bid/${id}`

DELETE `http://localhost:8080/driver/${id}`

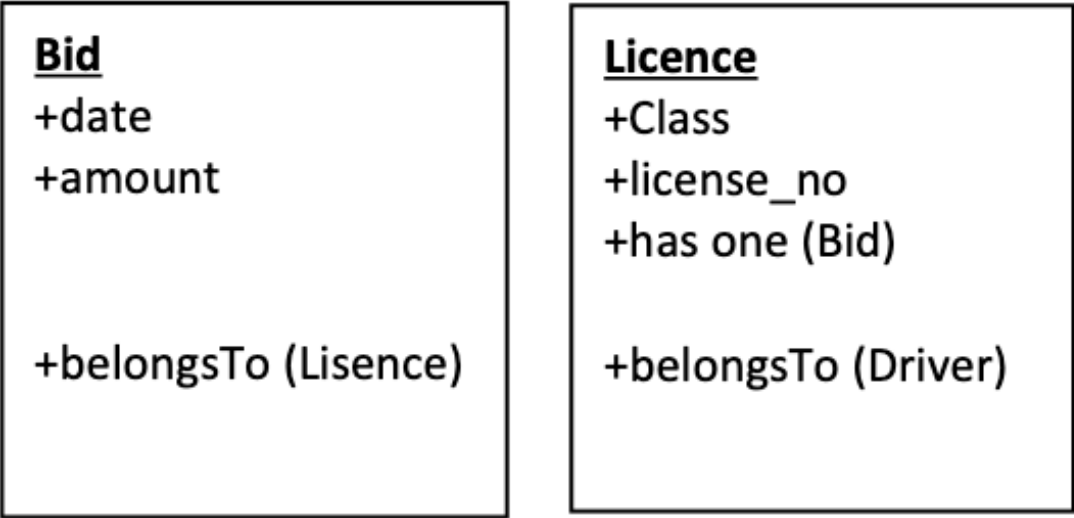
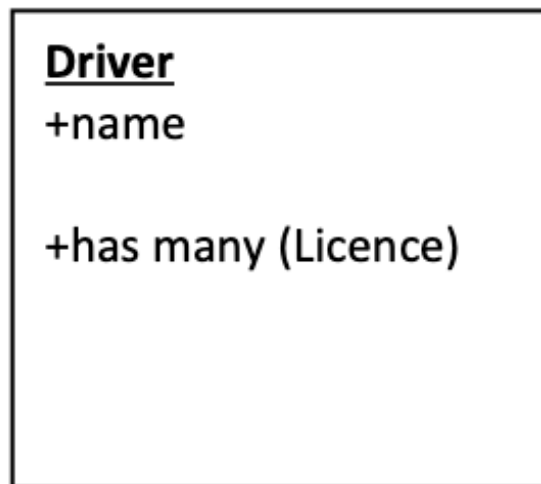
## Requirements

1. Create a SPA (Single Page Application) by using React and Vue  
you may refer to these template from grails to get you started

<https://guides.grails.org/building-a-vue-app/guide/index.html>

<https://guides.grails.org/building-a-react-app/guide/index.html>

2. Based on the COMPLETE solution on the repository referred in above link, create the object **Driver**, **Bid** and **Licence**



3. Develop a web app that displays the list of Bids and licence belonging to the driver

Name	Class	Licence no	Bid Date	Amount
Susan	Class_A	12345678	01/01/2020	\$5000
Joe	Class_A	24682468	02/02/2020	\$8000
Peter	Class_B	55555555	01/01/2020	\$5555

4. Develop the feature of edit and save the record

Susan

Class C

12341234

05/05/2020

\$8888

Save

Name	Class	Licence no	Bid Date	Amount
Susan	Class_A	12345678	01/01/2020	\$5000
Joe	Class_A	24682468	02/02/2020	\$8000
Peter	Class_B	55555555	01/01/2020	\$5555

What have been achieved in this project

1. Complete CRUD requests

CREATE license

**READ** license, driver, bid

**UPDATE** license

**DELETE** license

2. a simple validation to ensure fields are filled correctly or not empty
3. Toasts to improve user experience in exception or event handling
4. Build in using React hook functional component for better readability
5. Components to wrap the UI for possible reuses

dependencies: bootstrap, axios, Blueprint.js

## **What can be improved or added for future implementation**

1. more sophisticated validation
2. redux for a cleaner state management system or minimally use custom hooks