Санкт-Петербургский Национальный Исследовательский Университет Информационных Технологий, Механики и Оптики

ПИиКТ, кафедра ИПМ

Курсовая работа по дисциплине

«Программирование интернет-приложений»

Выполнили:

Каменский Валерий

Чудаков Матвей

P3219

Преподаватели:

Цопа Евгений Алексеевич

Харитонова Анастасия Евгеньевна

Санкт-Петербург 2017

**Описание разрабатываемой системы**

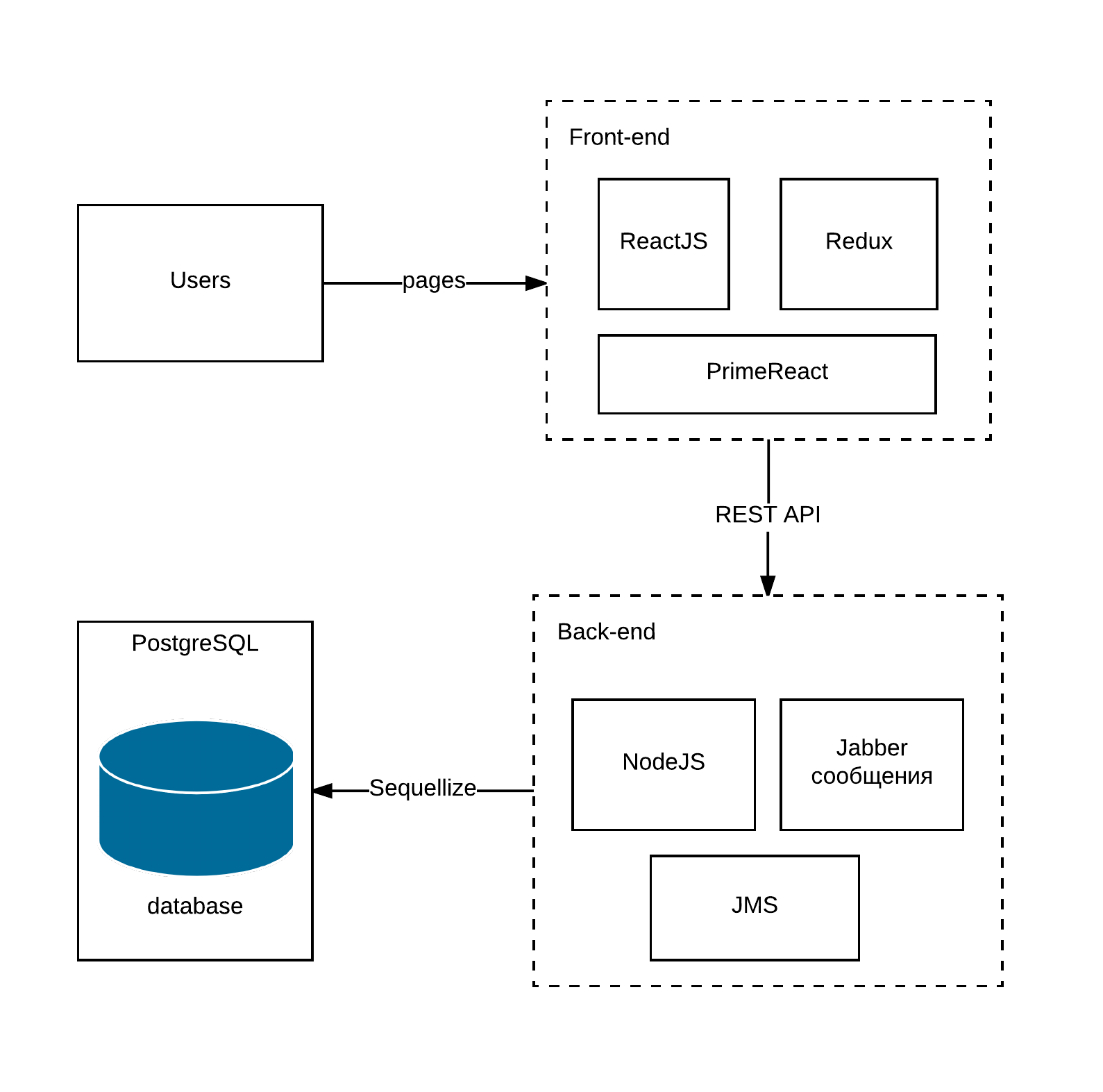
Предметом разработки является веб-сайт “PIPFUT” – ресурс, предоставляющий информацию об видеоигре FIFA. Данный сайт разрабатывается в первую очередь для людей, интересующихся игрой Fifa, а также для тех, кто увлекается футболом, как видом спорта. Он актуален, как и для опытных игроков, так и для новичков. Функционал сайта предоставляет удобный интерфейс для просмотра характеристик игроков, давая те возможности, которые отсутствуют в игре..

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Требование** | **Полезность** | **Трудоемкость** | **Стабильность** | **Риск** |
| **Функциональные требования** | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| F1 | Возможность просматривать информацию об игроках | Высокая | 10 | Высокая | Критический |
| F2 | Возможность регистрироваться и авторизовываться (для зарегистрированных пользователей) | Высокая | 10 | Высокая | Критический |
| F3 | Возможность сохранения игроков в избранное (для зарегистрированных пользователей) | Высокая | 10 | Высокая | Критический |
| F4 | Возможность комментировать содержимое сайта | Средняя | 10 | Высокая | Допустимый |
| F5 | Возможность участвовать в обсуждении команд | Средняя | 10 | Высокая | Допустимый |
| F6 | Возможность просматривать информацию о командах | Высокая | 10 | Высокая | Критический |
| F7 | Возможность сортировать игроков по командам | Высокая | 10 | Высокая | Критический |
| F8 | Возможность сортировать игроков по странам | Высокая | 10 | Высокая | Критический |
| F9 | Возможность удалять игроков из избранного | Высокая | 10 | Высокая | Критический |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Нефункциональные требования | | | | | |
| UF1 | Уровень back-end должен быть основан на NodeJS | Высокая | 10 | Высокая | Критический |
| UF2 | Уровень front-end должен быть построен на ReactJS + Redux (необходимо использовать ES6 и JSX) с использованием набора компонентов PrimeReact. | Высокая | 10 | Высокая | Критический |
| UF3 | Взаимодействие между уровнями back-end и front-end должно быть организовано посредством REST API. | Высокая | 10 | Высокая | Критический |
| UF4 | Интерфейс системы д.б. адаптирован для отображения в 3-х режимах: “Десктопный” – ширина экрана равна или превышает 1217 пикселей, 710<= “Планшетный” <1217, “Мобильный” < 710 | Высокая | 20 | Высокая | Критический |
| UF5 | Для доступа к БД необходимо использовать Sequelize | Высокая | 10 | Высокая | Критический |
| UF6 | Система оповещения пользователей о важных событиях д. б. реализована посредством jabber-сообщений. | Высокая | 5 | Высокая | Критический |
| UF7 | Уровень DataAccess должен содержать CRUD API ко всем сущностям базы данных | Высокая | 5 | Высокая | Критический |
| UF8 | В качестве провайдера очереди сообщений необходимо использовать RabbitMQ | Высокая | 5 | Высокая | Критический |
| UF9 | Пропускная способность 50 пользователей | Высокая | 25 | Средняя | Критический |
| UF10 | Обработка основных HTTP ошибок (403, 404, 502, 503) | Высокая | 5 | Высокая | Критический |

**Блок-схема архитектуры разрабатываемой системы**

****

**Исходный код**

/models/index.js

**const** fs = *require*(**'fs'**);  
**const** path = *require*(**'path'**);  
**const** Sequelize = *require*(**'sequelize'**);  
**const** basename = path.*basename*(module.**filename**);  
**const** env = ***process***.env.NODE\_ENV || **'development'**;  
**const** config = *require*(**`**${\_\_dirname}**/../config/config.json`**)[env];  
**const** db = {};  
  
**let** sequelize;  
**if** (config.use\_env\_variable) {  
 sequelize = **new** Sequelize(***process***.env[config.use\_env\_variable]);  
} **else** {  
 sequelize = **new** Sequelize(  
 config.**database**, config.**username**, config.**password**, config  
 );  
}  
  
fs  
 .*readdirSync*(\_\_dirname)  
 .filter(file =>  
 (file.indexOf(**'.'**) !== 0) &&  
 (file !== basename) &&  
 (file.slice(-3) === **'.js'**))  
 .forEach(file => {  
 **const** model = sequelize.import(path.*join*(\_\_dirname, file));  
 db[model.**name**] = model;  
 });  
  
***Object***.keys(db).forEach(modelName => {  
 **if** (db[modelName].associate) {  
 db[modelName].associate(db);  
 }  
});  
  
db.**sequelize** = sequelize;  
db.**Sequelize** = Sequelize;  
  
module.**exports** = db;

/models/players.js

module.**exports** = (sequelize, DataTypes) => {  
 **const** players = sequelize.define(**'players'**, {  
 **name**: {  
 **type**: DataTypes.**STRING**,  
 **allowNull**: **false**,  
 },  
 **surname**: {  
 **type**: DataTypes.**STRING**,  
 **allowNull**: **false**,  
 }  
 });  
 players.associate = (models) => {  
 players.hasMany(models.contracts, {  
 **foreignKey**: **'playerID'**,  
 **as**: **'contracts'**,  
 });  
 };  
 **return** players;  
};

/models/contracrs.js

module.**exports** = (sequelize, DataTypes) => {  
 **const** contracts = sequelize.define(**'contracts'**, {  
 **reiting**: {  
 **type**: DataTypes.**INTEGER**,  
 **allowNull**: **false**,  
 },  
  
 });  
 contracts.associate = (models) => {  
 contracts.**belongsTo**(models.players, {  
 **foreignKey**: **'playerID'**,  
 **onDelete**: **'CASCADE'**,  
 });  
 contracts.**belongsTo**(models.teams, {  
 **foreignKey**: **'teamID'**,  
 **onDelete**: **'CASCADE'**,  
 });  
 contracts.**hasOne**(models.characteristics, {  
 **foreignKey**: **'contractID'**,  
 **as**: **'characteristics'**,  
 });};  
 **return** contracts;  
};

/controller/playerController.js

**const** Players = *require*(**'../models'**).players;

list(req, res){  
 **return** Players  
 .all()  
 .then((player) => res.status(200).send(player))  
 .catch(error => res.status(400).send(error));  
},  
  
one(req, res){  
 **return** Players  
 .*findOne*({ **where**: {**name**: req.**params**.**name** }})  
 .then((player) => res.status(200).send(player))  
 .catch(error => res.status(400).send(error));  
},  
  
create(req, res){  
 **return** Players  
 .create({  
 **name**: req.**body**.**name**,  
 })  
 .then((player) => res.status(201).send(player))  
 .catch((error) => res.status(400).send(error));  
},  
  
update(req, res) {  
 **return** Players  
 .*findById*(req.**params**.**id**)  
 .then(player => {  
 **if** (!player) {  
 **return** res.status(404).send({  
 **message**: **'Player Not Found'**,  
 });  
 }  
 **return** player  
 .update({  
 **name**: req.**body**.**name** || player.**name**,  
 })  
 .then(() => res.status(200).send(player))  
 .catch((error) => res.status(400).send(error));  
 })  
 .catch((error) => res.status(400).send(error));  
},  
  
**delete**(req, res){  
 **return** Players  
 .*findById*(req.**params**.**id**)  
 .then(player => {  
 **if** (!player) {  
 **return** res.status(400).send({  
 **message**: **'Player Not Found'**,  
 });  
 }  
 **return** player  
 .destroy()  
 .then(() => res.status(204).send())  
 .catch(error => res.status(400).send(error));  
 })  
 .catch(error => res.status(400).send(error));  
}

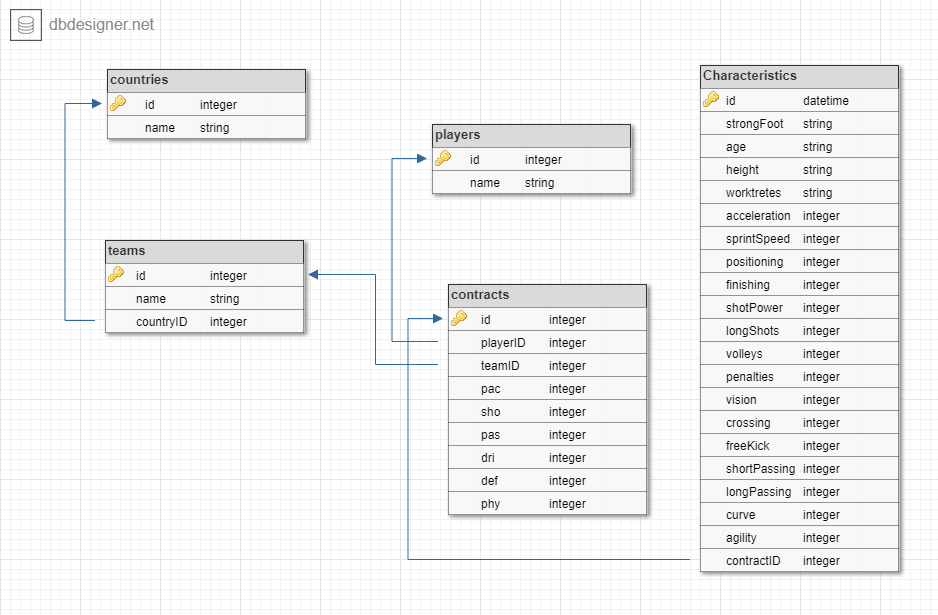
/controller/contractController.js

**const** Players = *require*(**'../models'**).players;  
**const** Contracts = *require*(**"../models"**).contracts;  
**const** Teams = *require*(**"../models"**).teams;  
**const** Countries = *require*(**'../models'**).countries;  
**const** CommentsPlayers = *require*(**'../models'**).commentsPlayers;  
**const** Characteristics = *require*(**'../models'**).characteristics;  
  
module.**exports** = {  
 list(req, res){  
 **return** Contracts  
 .*findAll*({  
 **include**:[{  
 **model**: Players, **as**: **'player'**,  
 **include**:[{  
 **model**: CommentsPlayers, **as**: **'commentsPlayers'**,  
 }]  
 },{  
 **model**: Teams, **as**: **'team'**,  
 **include**:[{  
 **model**: Countries, **as**: **'country'**,  
 }]  
 }]  
 })  
 .then(contracts => res.status(200).send(contracts))  
 .catch(error => res.status(400).send(error));  
 },  
  
 one(req, res){  
 **return** Contracts  
 .*findOne*({  
 **where**: {  
 **id**: req.**params**.**id**,  
 },  
 **include**: [{  
 **model**: Characteristics, **as**: **'characteristics'**,  
 },{  
 **model**: Players, **as**: **'player'**,  
 **include**: [{  
 **model**: CommentsPlayers, **as**: **'commentsPlayers'**,  
 }]  
 },{  
 **model**: Teams, **as**: **'team'**,  
 **include**: [{  
 **model**: Countries, **as**: **'country'**,  
 }]  
 }]  
 })  
 .then(contract => res.status(200).send(contract))  
 .catch(error => res.status(400).send(error));  
 },  
  
 listLimit(req, res){  
 **return** Contracts  
 .*findAll*({  
 **limit**: req.**params**.**limit**,  
 **offset**: (req.**params**.**limit** \* req.**params**.**offset**),  
 **include**:[{  
 **model**: Players, **as**: **'player'**,  
 },{  
 **model**: Teams, **as**: **'team'**,  
 **include**:[{  
 **model**: Countries, **as**: **'country'**,  
 }]  
 }]  
 })  
 .then(contracts => res.status(200).send(contracts))  
 .catch(error => res.status(400).send(error));  
 },  
  
 listLike(req, res){  
 ***console***.log(**"lol"**);  
 **return** Contracts  
 .*findAll*({  
 **limit**: req.**params**.**limit**,  
 **offset**: (req.**params**.**limit** \* req.**params**.**offset**),  
 **include**:[{  
 **model**: Players, **as**: **'player'**,  
 **where**: { **name**: { **$like**: **'%'** + req.**params**.**like** + **'%'** } }  
 },{  
 **model**: Teams, **as**: **'team'**,  
 **include**:[{  
 **model**: Countries, **as**: **'country'**,  
 }]  
 }]  
 })  
 .then(contracts => res.status(200).send(contracts))  
 .catch(error => res.status(400).send(error));  
 },  
  
 listByTeam(req, res){  
 **return** Contracts  
 .*findAll*({  
 **include**:[{  
 **model**: Players, **as**: **'player'**,  
 },{  
 **model**: Teams, **as**: **'team'**,  
 **where**: { **name**: req.**params**.**team** },  
 **include**:[{  
 **model**: Countries, **as**: **'country'**,  
 }]  
 }],  
 })  
 .then(contracts => res.status(200).send(contracts))  
 .catch(error => res.status(400).send(error));  
 },  
  
 listByCountry(req, res){  
 **return** Contracts  
 .*findAll*({  
 **include**:[{  
 **model**: Players, **as**: **'player'**,  
 **include**:[{  
 **model**: CommentsPlayers, **as**: **'commentsPlayers'**,  
 }]  
 },{  
 **model**: Teams, **as**: **'team'**,  
 **include**:[{  
 **model**: Countries, **as**: **'country'**,  
 **where**: { **name**: req.**params**.country }  
 }]  
 }]  
 })  
 .then(contracts => res.status(200).send(contracts))  
 .catch(error => res.status(400).send(error));  
 },  
};

/migrations/20180308113207-create-players.js

module.**exports** = {  
 up: (queryInterface, Sequelize) => {  
 **return** queryInterface.createTable(**'players'**, {  
 **id**: {  
 **allowNull**: **false**,  
 **autoIncrement**: **true**,  
 **primaryKey**: **true**,  
 **type**: Sequelize.*INTEGER*,  
 },  
 **name**: {  
 **type**: Sequelize.*STRING*,  
 **allowNull**: **false**,  
 },  
 **createdAt**: {  
 **allowNull**: **true**,  
 **type**: Sequelize.*DATE* },  
 **updatedAt**: {  
 **allowNull**: **true**,  
 **type**: Sequelize.*DATE* }  
 });  
 },  
 down: (queryInterface, Sequelize) => {  
 **return** queryInterface.dropTable(**'players'**);  
 }  
};

**DDL**



**Примеры запросов**

/main/contracts/2/0

[

{

"id": 1,

"reiting": 99,

"pac": 98,

"sho": 99,

"pas": 94,

"dri": 98,

"def": 50,

"phy": 95,

"refImage": "https://futhead.cursecdn.com/static/img/18/players\_alt/p67129665.png",

"createdAt": "2018-04-07T14:01:44.627Z",

"updatedAt": "2018-04-07T14:01:44.627Z",

"playerID": 1,

"teamID": 5,

"player": {

"id": 1,

"name": "Cristiano Ronaldo",

"createdAt": "2018-04-07T14:01:26.747Z",

"updatedAt": "2018-04-07T14:01:26.747Z"

},

"team": {

"id": 5,

"name": "Real Madrid CF",

"refClubs": "https://futhead.cursecdn.com/static/img/18/clubs/243.png",

"createdAt": "2018-04-07T14:01:17.612Z",

"updatedAt": "2018-04-07T14:01:17.612Z",

"countryId": 1,

"country": {

"id": 1,

"name": "Portugal",

"refNations": "https://futhead.cursecdn.com/static/img/18/nations/38.png",

"createdAt": "2018-04-07T14:01:08.480Z",

"updatedAt": "2018-04-07T14:01:08.480Z"

}

}

},

{

"id": 2,

"reiting": 98,

"pac": 96,

"sho": 97,

"pas": 98,

"dri": 99,

"def": 45,

"phy": 81,

"refImage": "https://futhead.cursecdn.com/static/img/18/players\_alt/p117598535.png",

"createdAt": "2018-04-07T14:01:44.628Z",

"updatedAt": "2018-04-07T14:01:44.628Z",

"playerID": 2,

"teamID": 4,

"player": {

"id": 2,

"name": "Lionel Messi",

"createdAt": "2018-04-07T14:01:26.748Z",

"updatedAt": "2018-04-07T14:01:26.748Z"

},

"team": {

"id": 4,

"name": "FC Barcelona",

"refClubs": "https://futhead.cursecdn.com/static/img/18/clubs/241.png",

"createdAt": "2018-04-07T14:01:17.612Z",

"updatedAt": "2018-04-07T14:01:17.612Z",

"countryId": 3,

"country": {

"id": 3,

"name": null,

"refNations": "https://futhead.cursecdn.com/static/img/18/nations/52.png",

"createdAt": "2018-04-07T14:01:08.483Z",

"updatedAt": "2018-04-07T14:01:08.483Z"

}

}

}

]

/main/contracts/1

{

"id": 1,

"reiting": 99,

"pac": 98,

"sho": 99,

"pas": 94,

"dri": 98,

"def": 50,

"phy": 95,

"refImage": "https://futhead.cursecdn.com/static/img/18/players\_alt/p67129665.png",

"createdAt": "2018-04-07T14:01:44.627Z",

"updatedAt": "2018-04-07T14:01:44.627Z",

"playerID": 1,

"teamID": 5,

"characteristics": {

"id": 1,

"strongFoot": "Right",

"age": "32 - 05/02/1985",

"height": "185cm | 6'1\"",

"workrates": "High / Low",

"acceleration": 96,

"sprintSpeed": 99,

"positioning": 99,

"finishing": 99,

"shotPower": 99,

"longShots": 97,

"volleys": 93,

"penalties": 90,

"vision": 97,

"crossing": 97,

"freeKick": 87,

"shortPassing": 95,

"longPassing": 88,

"curve": 92,

"agility": 96,

"balance": 68,

"reactions": 99,

"ballControl": 99,

"dribbling": 99,

"composure": 95,

"interceptions": 43,

"heading": 99,

"marking": 33,

"standingTackle": 46,

"slidingTackle": 34,

"jumping": 99,

"stamina": 99,

"strength": 95,

"aggression": 74,

"createdAt": "2018-04-08T14:13:59.296Z",

"updatedAt": "2018-04-08T14:13:59.296Z",

"contractID": 1

},

"player": {

"id": 1,

"name": "Cristiano Ronaldo",

"createdAt": "2018-04-07T14:01:26.747Z",

"updatedAt": "2018-04-07T14:01:26.747Z",

"commentsPlayers": []

},

"team": {

"id": 5,

"name": "Real Madrid CF",

"refClubs": "https://futhead.cursecdn.com/static/img/18/clubs/243.png",

"createdAt": "2018-04-07T14:01:17.612Z",

"updatedAt": "2018-04-07T14:01:17.612Z",

"countryId": 1,

"country": {

"id": 1,

"name": "Portugal",

"refNations": "https://futhead.cursecdn.com/static/img/18/nations/38.png",

"createdAt": "2018-04-07T14:01:08.480Z",

"updatedAt": "2018-04-07T14:01:08.480Z"

}

}

}

/main/teams/like/1/0/Real

[

{

"id": 5,

"name": "Real Madrid CF",

"refClubs": "https://futhead.cursecdn.com/static/img/18/clubs/243.png",

"createdAt": "2018-04-07T14:01:17.612Z",

"updatedAt": "2018-04-07T14:01:17.612Z",

"countryId": 1,

"country": {

"id": 1,

"name": "Portugal",

"refNations": "https://futhead.cursecdn.com/static/img/18/nations/38.png",

"createdAt": "2018-04-07T14:01:08.480Z",

"updatedAt": "2018-04-07T14:01:08.480Z"

}

}

]

**Исходный код**

/index.js

**const express** = *require*(**'express'**);  
**const** *logger* = *require*(**'morgan'**);  
**const bodyParser** = *require*(**'body-parser'**);  
**const** passport = *require*(**'passport'**);  
**const** *session* = *require*(**'express-session'**);  
**const flash** = *require*(**'connect-flash'**);  
  
**const load** = *require*(**'./load/load'**);  
**const** config = *require*(**'./config'**);  
  
**const** app = **express**();  
app.use(*logger*(**'dev'**));  
app.use(**bodyParser**.**json**());  
app.use(**bodyParser**.**urlencoded**({**extended**: **false**}));  
app.use(**flash**());  
  
app.use(*session*({**secret**: **'lol'**, **resave**: **true**, **saveUninitialized**: **true**}));  
app.use(passport.initialize());  
app.use(passport.session());  
*require*(**'./passport/index'**)(passport);  
  
app.use(**express**.**static**(\_\_dirname + **'/static'**));  
*require*(**'./routes/index'**)(app, passport);  
  
module.**exports** = app;

/routes/index.js

**const** playerController = *require*(**'../controller'**).**playerController**;  
**const** authorizationController = *require*(**'../controller'**).**authorizationController**;  
**const** teamController = *require*(**'../controller'**).**teamController**;  
**const** countriesController = *require*(**'../controller'**).**countriesController**;  
**const** contractsController = *require*(**'../controller'**).**contractsController**;  
**const** load = *require*(**'../load'**).**load**;  
**const** favoritesController = *require*(**'../controller'**).**favoritesController**;  
  
module.**exports** = (app, passport) => {  
  
 **const** *isLoggedIn* = (req, resp, next) => {  
 **if** (req.isAuthenticated())  
 **return** next();  
 resp.*redirect*(**'/signin'**);  
 };  
  
 **const** *isNotLoggedIn* = (req, resp, next) => {  
 **if** (!req.isAuthenticated())  
 **return** next();  
 resp.*redirect*(**'/main'**);  
 };  
  
 app.get(**'/'**, (req, res) =>{  
 res.redirect(**'/main'**);  
 });  
  
 app.get(**'/signin'**, *isNotLoggedIn*, authorizationController.signin);  
 app.get(**'/signup'**, *isNotLoggedIn*, authorizationController.signup);  
 app.get(**'/logout'**, *isLoggedIn*, authorizationController.logout);  
  
 app.post(**'/signin'**, passport.authenticate(**'local-signin'**, {  
 **badRequestMessage**: **'Заполните ВСЕ поля.'**,  
 **successRedirect**: **'/main'**,  
 **failureRedirect**: **'/signin.html?error=true'**,  
 **failureFlash**: **true** }));  
 app.post(**'/signup'**, passport.authenticate(**'local-signup'**, {  
 **badRequestMessage**: **'Заполните ВСЕ поля.'**,  
 **successRedirect**: **'/main'**,  
 **failureRedirect**: **'/signup.html?error=true'**,  
 **failureFlash**: **true**,  
 }));  
  
 app.get(**'/main'**, *isLoggedIn*, (req, res) => res.status(200).send({  
 **message**: **'hw'**,  
 }));  
  
 app.get(**'/main/contracts/:id'**, contractsController.one);  
 app.get(**'/main/contracts/team/:team'**, contractsController.listByTeam);  
 app.get(**'/main/contracts/:limit/:offset'**, contractsController.listLimit);  
 app.get(**'/main/contracts/like/:limit/:offset/:like'**, contractsController.listLike);  
 app.get(**'/main/contracts/country/:country'**, contractsController.listByCountry);  
  
 app.get(**'/main/favorites/players'**, favoritesController.**list**);  
  
 app.get(**'/main/teams/:limit/:offset'**, teamController.listLimit);  
 app.get(**'/main/teams/like/:limit/:offset/:like'**, teamController.listLike);  
 app.get(**'/main/teams/country/:country'**, teamController.listByCountry);  
  
 app.get(**'/main/countries/like/:limit/:offset/:like'**, countriesController.listLike);  
 app.get(**'/main/countries/:limit/:offset'**, countriesController.listLimit);  
};

**Front-end**

**/index.js**

**import** React **from 'react'**;  
**import** ReactDOM **from 'react-dom'**;  
**import './index.css'**;  
**import** { **Provider** } **from 'react-redux'**;  
**import** store **from './store'**;  
**import** { Router, Route, browserHistory } **from 'react-router'**;  
**import** { **syncHistoryWithStore** } **from 'react-router-redux'**;  
**import** ListOfContracts **from './components/listOfContracts'**;  
**import** ListOfTeam **from './components/listOfTeam'**;  
**import** ListOfCountries **from './components/listOfCountries'**;  
**import** Player **from './components/player'**;  
**import** Home **from './components/home'**;  
**import** { **Switch** } **from 'react-router-dom'**;  
**import** Favorite **from './components/favorite'**;  
  
**const** history = **syncHistoryWithStore**(browserHistory, store);  
  
ReactDOM.render(  
 <**Provider store=**{store}>  
 <**Router history=**{history}>  
 <**Switch**>  
 <**Route exact path="/" component=**{Home}/>  
 <**Route path="/players" component=**{ListOfContracts}/>  
 <**Route path="/players/team/:teamId" component=**{ListOfContracts}/>  
 <**Route path="/players/country/:countryId" component=**{ListOfContracts}/>  
 <**Route path="/player/:id" component=**{Player}/>  
 <**Route path="/teams" component=**{ListOfTeam}/>  
 <**Route path="/countries" component=**{ListOfCountries}/>  
 <**Route path="/favorite" component=**{Favorite}/>  
 <**Route path="\*" component=**{Home}/>  
 </**Switch**>  
 </**Router**>  
 </**Provider**>,  
 ***document***.getElementById(**'root'**)  
);

**import** React **from 'react'**;  
**import** ReactDOM **from 'react-dom'**;  
**import './index.css'**;  
**import** { **Provider** } **from 'react-redux'**;  
**import** store **from './store'**;  
**import** { Router, Route, browserHistory } **from 'react-router'**;  
**import** { **syncHistoryWithStore** } **from 'react-router-redux'**;  
**import** ListOfContracts **from './components/listOfContracts'**;  
**import** ListOfTeam **from './components/listOfTeam'**;  
**import** ListOfCountries **from './components/listOfCountries'**;  
**import** Player **from './components/player'**;  
**import** Home **from './components/home'**;  
**import** { **Switch** } **from 'react-router-dom'**;  
**import** Favorite **from './components/favorite'**;  
  
**const** history = **syncHistoryWithStore**(browserHistory, store);  
  
ReactDOM.render(  
 <**Provider store=**{store}>  
 <**Router history=**{history}>  
 <**Switch**>  
 <**Route exact path="/" component=**{Home}/>  
 <**Route path="/players" component=**{ListOfContracts}/>  
 <**Route path="/players/team/:teamId" component=**{ListOfContracts}/>  
 <**Route path="/players/country/:countryId" component=**{ListOfContracts}/>  
 <**Route path="/player/:id" component=**{Player}/>  
 <**Route path="/teams" component=**{ListOfTeam}/>  
 <**Route path="/countries" component=**{ListOfCountries}/>  
 <**Route path="/favorite" component=**{Favorite}/>  
 <**Route path="\*" component=**{Home}/>  
 </**Switch**>  
 </**Router**>  
 </**Provider**>,  
 ***document***.getElementById(**'root'**)  
);

**/listOfContract.js**

**import** React, { *Component* } **from 'react'**;  
**import** Img **from 'react-image'**;  
**import** { **connect** } **from 'react-redux'**;  
**import './css/listOfContracts.css'**;  
**import** $ **from "jquery"**;  
**import** Head **from './part/head'**;  
**import ScrollEvent from 'react-onscroll'**;  
**import** { Link } **from 'react-router'**;  
  
**const** urlOnDataLoad = **"/main/contracts/"**;  
**const** urlOnUserName = **"/main/user/"**;  
  
**class** ListOfContract **extends** *Component* {  
 constructor() {  
 **super**();  
 **this**.**state** = {  
 **offset**: 0,  
 **limit**: 10,  
 **urlOnFind**: urlOnDataLoad + **'like/'**,  
 **urlOnDataLoad**: urlOnDataLoad,  
 **isAuth**: **false**,  
 **login**: **''**,  
 **isFavorite**: **false**,  
 };  
 **this**.loadName = **this**.loadName.bind(**this**);  
 **this**.handleScrollCallback = **this**.handleScrollCallback.bind(**this**);  
 **this**.findContracts = **this**.findContracts.bind(**this**);  
 }  
  
 componentDidMount() {  
 **this**.**props**.onDeleteContracts();  
 **this**.loadName();  
 **if** (**this**.**props**.**params**.teamId != **undefined**) {  
 **this**.**state**.**urlOnDataLoad** = urlOnDataLoad + **'team/'** + **this**.**props**.**params**.teamId + **'/'**;  
 **this**.**state**.**urlOnFind** = urlOnDataLoad + **'team/like/'** + **this**.**props**.**params**.teamId + **'/'**;  
 }  
 **else if** (**this**.**props**.**params**.**countryId** != **undefined**) {  
 **this**.**state**.**urlOnDataLoad** = urlOnDataLoad + **'country/'** + **this**.**props**.**params**.**countryId** + **'/'**;  
 **this**.**state**.**urlOnFind** = urlOnDataLoad + **'country/like/'** + **this**.**props**.**params**.**countryId** + **'/'**;  
 }  
 **this**.addContracts();  
 };  
  
 handleScrollCallback() {  
 **const** scroll\_height = $(***document***).height();  
 **const** scroll\_position = $(***window***).height() + $(***window***).**scrollTop**();  
 **if** (scroll\_position === scroll\_height) {  
 **if** (**this**.**findInput**.**value** === **""** || **this**.**findInput**.**value** === **null**) {  
 **this**.addContracts();  
 }  
 **else** {  
 **this**.findContracts();  
 }  
 }  
 }  
  
 getUserFavorite() {  
  
 }  
  
 inputChange() {  
 **this**.**state**.**offset** = 0;  
 **this**.**props**.onDeleteContracts();  
 **if** (**this**.**findInput**.**value** === **""** || **this**.**findInput**.**value** === **null**) {  
 **this**.addContracts();  
 }  
 **else** {  
 **this**.findContracts();  
 }  
 }  
  
 addContracts() {  
 $.ajax({  
 **url**: (**this**.**state**.**urlOnDataLoad** + **this**.**state**.**limit** + **'/'** + **this**.**state**.**offset**),  
 **dataType**: **'json'**,  
 **cache**: **false**,  
 **success**: **function** (data) {  
 data.sort((contract1, contract2) => {  
 **return** contract2.**reiting** - contract1.**reiting**;  
 });  
 data.forEach((contract) => {  
 **this**.**props**.onAddContracts(contract);  
 });  
 **this**.**state**.**offset**++;  
 }.bind(**this**),  
 error: **function** (xhr, status, err) {  
 ***console***.error(urlOnDataLoad, status, err.toString());  
 }  
 });  
 }  
  
 findContracts() {  
 $.ajax({  
 **url**: (**this**.**state**.**urlOnFind** + **this**.**state**.**limit** + **'/'** + **this**.**state**.**offset** + **'/'** + **this**.**findInput**.**value**),  
 **dataType**: **'json'**,  
 **cache**: **false**,  
 **success**: **function** (data) {  
 **if** (**this**.**findInput**.**value** === **""** || **this**.**findInput**.**value** === **null**) {  
 **this**.**state**.**offset** = 0;  
 **this**.addContracts();  
 **return**;  
 }  
 data.forEach((contract) => {  
 **this**.**props**.onAddContracts(contract);  
 });  
 **this**.**state**.**offset**++;  
 }.bind(**this**),  
 error: **function** (xhr, status, err) {  
 ***console***.error(**this**.**state**.**urlOnFind**, status, err.toString());  
 }  
 });  
 }  
  
 loadName() {  
 $.ajax({  
 **url**: (urlOnUserName),  
 **dataType**: **'json'**,  
 **cache**: **false**,  
 **success**: **function** (data) {  
 **if** (data.**login** != **null**) {  
 **this**.setState({**isAuth**: **true**});  
 **this**.setState({**login**: data.**login**});  
 }  
 }.bind(**this**),  
 error: **function** (xhr, status, err) {  
 ***console***.error(urlOnUserName, status, err.toString());  
 }  
 });  
 }  
  
 addFavorite(id){  
 $.ajax({  
 **type**: **"POST"**,  
 **url**: **'/main/add/favorite/'** + id,  
 **cache**: **false**,  
 **success**: **function** (data) {  
 **this**.setState({**isAuth**: **true**});  
 }.bind(**this**),  
 error: **function** (xhr, status, err) {  
 ***console***.error(**'/main/add/comment/'** + id, status, err.toString());  
 }  
 });  
 }  
  
 render() {  
 **return** (  
 <**div className="ListOfContract"**>  
 <**ScrollEvent handleScrollCallback=**{**this**.handleScrollCallback}/>  
 <**Head**/>  
 <**div class="content-header"**>  
 <**div class="content-header-left slash futhead"**>  
 <**h1 class="player-list-header"**>  
 <**small**>FIFA 18 Database</**small**>  
  
 Top Players  
  
 <**small**>Best FUT 18 Players</**small**>  
 </**h1**>  
 </**div**>  
 </**div**>  
 <**div class="content-table"**>  
 <**ul class="list-group"**>  
 <**li class="advansed-search padding-0"**>  
 <**div class="row obscure padding-15"**>  
 <**div class="col-md-10"**>  
 <**input class="filter form-control" type="text" placeholder="Filter By Name"  
 ref=**{(input) => {  
 **this**.**findInput** = input;  
 }}  
 **onChange=**{**this**.inputChange.bind(**this**)}/>  
 </**div**>  
 </**div**>  
 </**li**>  
 <**li**>  
 <**div class="player-item"**>  
 <**span class="stream-col-40"**>RAT</**span**>  
 <**span class="stream-col-80"**></**span**>  
 <**span class="stream-col-60"**>Name</**span**>  
 <**span class="pull-right padding-l-r-14 hidden-1050"**>  
 <**span class="stream-col-60"**>PAC</**span**>  
 <**span class="stream-col-60"**>SHO</**span**>  
 <**span class="stream-col-60"**>PAS</**span**>  
 <**span class="stream-col-60"**>DRI</**span**>  
 <**span class="stream-col-60"**>DEF</**span**>  
 <**span class="stream-col-60"**>PHY</**span**>  
 <**span class="stream-col-60" hidden=**{!**this**.**state**.**isAuth**}>FAV</**span**>  
 </**span**>  
 </**div**>  
 </**li**>  
 {**this**.**props**.**contracts**.map((contract) =>  
 <**li key=**{contract.**id**.toString()}>  
 <**div class="display-block padding-0"**>  
 <**Link class="player-item" to=**{**'/player/'** + contract.**id**}>  
 <**span class="player-rating stream-col-50 text-center"**>{contract.**reiting**}</**span**>  
 <**span class="player-info"**>  
 <**Img class="player-image" src=**{contract.**refImage**}/>  
 <**Img class="player-club" src=**{contract.**team**.**refClubs**}/>  
 <**Img class="player-nation" src=**{contract.**player**.**country**.**refNations**}/>  
 <**span class="player-name"**>{contract.**player**.**name**}</**span**>  
 </**span**>  
 </**Link**>  
 <**span class="pull-right padding-l-r-14 hidden-1050"**>  
 <**span class="player-stat stream-col-60"**>  
 <**span class="value"**>{contract.**pac**}</**span**>  
 <**span class="hover-label"**>PAC</**span**>  
 </**span**>  
 <**span class="player-stat stream-col-60"**>  
 <**span class="value"**>{contract.**sho**}</**span**>  
 <**span class="hover-label"**>SHO</**span**>  
 </**span**>  
 <**span class="player-stat stream-col-60"**>  
 <**span class="value"**>{contract.**pas**}</**span**>  
 <**span class="hover-label"**>PAS</**span**>  
 </**span**>  
 <**span class="player-stat stream-col-60"**>  
 <**span class="value"**>{contract.**dri**}</**span**>  
 <**span class="hover-label"**>DRI</**span**>  
 </**span**>  
 <**span class="player-stat stream-col-60"**>  
 <**span class="value"**>{contract.**def**}</**span**>  
 <**span class="hover-label"**>DEF</**span**>  
 </**span**>  
 <**span class="player-stat stream-col-60"**>  
 <**span class="value"**>{contract.**phy**}</**span**>  
 <**span class="hover-label"**>PHY</**span**>  
 </**span**>  
 <**span class="player-stat stream-col-60" hidden=**{!**this**.**state**.**isAuth**}>  
 <**button class="fav-btn" onClick=**{() =>{  
 $.ajax({  
 **type**: **"POST"**,  
 **url**: **'/main/add/favorite/'** + contract.**player**.**id**,  
 **cache**: **false**,  
 **success**: **function** (data) {  
 **this**.setState({**isAuth**: **true**});  
 *alert*(**'Player added to favorites!'**);  
 }.bind(**this**),  
 error: **function** (xhr, status, err) {  
 ***console***.error(**'/main/add/comment/'** + **id**, status, err.toString());  
 }  
 });  
 }}>  
 <**span class="value"**>  
 <**div hidden=**{**this**.**state**.**isFavorite**}><**i  
 class="material-icons"**>favorite\_border</**i**></**div**>  
 <**div hidden=**{!**this**.**state**.**isFavorite**}><**i  
 class="material-icons"**>favorite</**i**></**div**>  
 </**span**>  
 </**button**>  
 </**span**>  
 </**span**>  
 </**div**>  
 </**li**>  
 )}  
 <**li**>  
 <**button class="more-button" onClick=**{**this**.addContracts.bind(**this**)}>More</**button**>  
 </**li**>  
 </**ul**>  
 </**div**>  
 </**div**>  
 );  
 }  
}  
  
**export default connect**(  
 state => ({  
 **contracts**: state.**contracts** }),  
 dispatch => ({  
 onAddContracts: (contracts) => {  
 dispatch({ **type**: **'ADD\_CONTRACTS'**, **payload**: contracts });  
 },  
 onDeleteContracts: () => {  
 dispatch({ **type**: **'DELETE\_CONTRACTS'** });  
 }  
 })  
)(ListOfContract);