# Solutions-Lab: Routing and Architecture

# Team Manager

//index.html

<!doctype html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" type="text/css" href="css/index.css">

<link rel="stylesheet" type="text/css" href="css/team.css">

<link rel="stylesheet" type="text/css" href="css/details.css">

<!-- <link rel="stylesheet" type="text/css" href="node\_modules/bootstrap/dist/css/bootstrap.min.css"> -->

<link rel="stylesheet" type="text/css" href="./node\_modules/bootstrap/dist/css/bootstrap.css">

<!-- <script src="node\_modules/jquery/dist/jquery.min.js"></script> -->

<script src="./node\_modules/jquery/dist/jquery.js"></script>

<!-- <script src="node\_modules/bootstrap/dist/js/bootstrap.min.js"></script> -->

<script src="./node\_modules/bootstrap/dist/js/bootstrap.js"></script>

<!-- <script src="node\_modules/handlebars/dist/handlebars.min.js"></script> -->

<script src="./node\_modules/handlebars/dist/handlebars.js"></script>

<!-- <script src="node\_modules/sammy/lib/min/sammy-latest.min.js"></script> -->

<script src="./node\_modules/sammy/lib/sammy.js"></script>

<!-- <script src="node\_modules/sammy/lib/min/plugins/sammy.handlebars-latest.min.js"></script> -->

<script src="./node\_modules/sammy/lib/min/plugins/sammy.handlebars-latest.min.js"></script>

<!-- <script src="scripts/models/requestService.js"></script> -->

<script src="scripts/models/requester.js"></script>

<script src="scripts/models/notificationService.js"></script>

<!-- <script src="scripts/models/authService.js"></script> -->

<script src="scripts/models/auth.js"></script>

<!-- <script src="scripts/models/crudService.js"></script> -->

<script src="scripts/models/teamsService.js"></script>

<script src="scripts/app.js"></script>

<script src="scripts/viewHandlers/homeHandler.js"></script>

<script src="scripts/viewHandlers/authHandler.js"></script>

<script src="scripts/viewHandlers/crudHandler.js"></script>

<title>Team Managers App</title>

</head>

<body>

<div id="infoBox" class="notification" disabled></div>

<div id="errorBox" class="notification" disabled></div>

<div id="main">

</div>

</body>

</html>

//package-lock.json

{

"name": "name",

"version": "1.0.0",

"lockfileVersion": 1,

"requires": true,

"dependencies": {

"bootstrap": {

"version": "4.3.1",

"resolved": "https://registry.npmjs.org/bootstrap/-/bootstrap-4.3.1.tgz",

"integrity": "sha512-rXqOmH1VilAt2DyPzluTi2blhk17bO7ef+zLLPlWvG494pDxcM234pJ8wTc/6R40UWizAIIMgxjvxZg5kmsbag=="

},

"commander": {

"version": "2.19.0",

"resolved": "https://registry.npmjs.org/commander/-/commander-2.19.0.tgz",

"integrity": "sha512-6tvAOO+D6OENvRAh524Dh9jcfKTYDQAqvqezbCW82xj5X0pSrcpxtvRKHLG0yBY6SD7PSDrJaj+0AiOcKVd1Xg==",

"optional": true

},

"handlebars": {

"version": "4.1.1",

"resolved": "https://registry.npmjs.org/handlebars/-/handlebars-4.1.1.tgz",

"integrity": "sha512-3Zhi6C0euYZL5sM0Zcy7lInLXKQ+YLcF/olbN010mzGQ4XVm50JeyBnMqofHh696GrciGruC7kCcApPDJvVgwA==",

"requires": {

"neo-async": "^2.6.0",

"optimist": "^0.6.1",

"source-map": "^0.6.1",

"uglify-js": "^3.1.4"

}

},

"jquery": {

"version": "3.3.1",

"resolved": "https://registry.npmjs.org/jquery/-/jquery-3.3.1.tgz",

"integrity": "sha512-Ubldcmxp5np52/ENotGxlLe6aGMvmF4R8S6tZjsP6Knsaxd/xp3Zrh50cG93lR6nPXyUFwzN3ZSOQI0wRJNdGg=="

},

"minimist": {

"version": "0.0.10",

"resolved": "https://registry.npmjs.org/minimist/-/minimist-0.0.10.tgz",

"integrity": "sha1-3j+YVD2/lggr5IrRoMfNqDYwHc8="

},

"neo-async": {

"version": "2.6.0",

"resolved": "https://registry.npmjs.org/neo-async/-/neo-async-2.6.0.tgz",

"integrity": "sha512-MFh0d/Wa7vkKO3Y3LlacqAEeHK0mckVqzDieUKTT+KGxi+zIpeVsFxymkIiRpbpDziHc290Xr9A1O4Om7otoRA=="

},

"optimist": {

"version": "0.6.1",

"resolved": "https://registry.npmjs.org/optimist/-/optimist-0.6.1.tgz",

"integrity": "sha1-2j6nRob6IaGaERwybpDrFaAZZoY=",

"requires": {

"minimist": "~0.0.1",

"wordwrap": "~0.0.2"

}

},

"sammy": {

"version": "0.7.6",

"resolved": "https://registry.npmjs.org/sammy/-/sammy-0.7.6.tgz",

"integrity": "sha1-Rk/lTR+JZhzDqYgAYwVxEyHpqcE="

},

"source-map": {

"version": "0.6.1",

"resolved": "https://registry.npmjs.org/source-map/-/source-map-0.6.1.tgz",

"integrity": "sha512-UjgapumWlbMhkBgzT7Ykc5YXUT46F0iKu8SGXq0bcwP5dz/h0Plj6enJqjz1Zbq2l5WaqYnrVbwWOWMyF3F47g=="

},

"uglify-js": {

"version": "3.5.3",

"resolved": "https://registry.npmjs.org/uglify-js/-/uglify-js-3.5.3.tgz",

"integrity": "sha512-rIQPT2UMDnk4jRX+w4WO84/pebU2jiLsjgIyrCktYgSvx28enOE3iYQMr+BD1rHiitWnDmpu0cY/LfIEpKcjcw==",

"optional": true,

"requires": {

"commander": "~2.19.0",

"source-map": "~0.6.1"

}

},

"wordwrap": {

"version": "0.0.3",

"resolved": "https://registry.npmjs.org/wordwrap/-/wordwrap-0.0.3.tgz",

"integrity": "sha1-o9XabNXAvAAI03I0u68b7WMFkQc="

}

}

}

//package.json

{

"name": "name",

"description": "description",

"authors": "author",

"version": "1.0.0",

"main": "pathToMain",

"dependencies": {

"bootstrap": "^4.3.1",

"handlebars": "^4.1.1",

"jquery": "^3.3.1",

"sammy": "^0.7.6"

}

}

//css.details.css

.details-box p {

margin: 1.5em;

}

.member {

font-size: 1.5em;

margin: 0.25em;

margin-left: 1em;

list-style-type: none;

display: block;

}

.titlebar {

display: block;

font-size: 2em;

padding: 0.25em;

padding-left: 0.75em;

}

.spanner {

display: block;

padding: 0.25em;

padding-left: 1.5em;

background-color: antiquewhite;

font-weight: 600;

}

//css.index.css

body {

margin: 0;

padding: 0;

font-family: sans-serif;

}

#main {

text-align: center;

}

.header {

background-color: #222;

height: 150px;

padding: 20px;

color: white;

}

.footer {

text-align: center;

position: fixed;

}

.notification {

font-size: 1.5em;

float: right;

color: #ffffff;

padding: 0.5em 2em 0.5em 2em;

margin: 1em;

display: none;

position: fixed;

top: 0;

right: 0;

text-align: center;

border-radius: 3px;

}

#infoBox {

background: #393;

}

#errorBox {

background: #F50;

}

//css.team.css

.team-box {

float:left;

width: 20em;

margin: 1em;

border: 3px solid antiquewhite;

text-decoration: none;

color: black;

}

.team-box:hover {

border: 3px solid black;

text-decoration: none;

color: black;

}

.spanner {

display: block;

padding: 0.25em;

background-color: antiquewhite;

}

.title {

font-size: 1.5em;

display: block;

padding: 0.25em;

padding-left: 1.5em;

font-weight: 600;

}

.team-box p {

margin: 1.5em;

}

//scripts.app.js

// $(() => {

// const app = Sammy('#main', function () {

// this.use('Handlebars', 'hbs');

// this.get('#/home', function(){

// this.loggedIn = !!sessionStorage.getItem('authtoken');

// this.username = sessionStorage.getItem('username');

// this.loadPartials({

// header: './templates/common/header.hbs',

// footer: './templates/common/footer.hbs'

// }).then(function() {

// this.partial('./templates/home/home.hbs');

// });

// });

// this.get('#/about', function(){

// this.loadPartials({

// header: './templates/common/header.hbs',

// footer: './templates/common/footer.hbs'

// }).then(function() {

// this.partial('./templates/about/about.hbs');

// });

// });

// this.get('#/login', function(){

// this.loadPartials({

// header: './templates/common/header.hbs',

// footer: './templates/common/footer.hbs',

// loginForm: './templates/login/loginForm.hbs'

// }).then(function() {

// this.partial('./templates/login/loginPage.hbs');

// });

// });

// this.get('#/logout', async function() {

// let result = await auth.logout();

// sessionStorage.clear();

// auth.showInfo('You are logged out successfully!');

// this.redirect('#/home');

// });

// this.get('#/register', function(){

// this.loadPartials({

// header: './templates/common/header.hbs',

// footer: './templates/common/footer.hbs',

// registerForm: './templates/register/registerForm.hbs'

// }).then(function() {

// this.partial('./templates/register/registerPage.hbs');

// });

// });

// this.post('#/register', function(context){

// let that = this;

// let {username, password, repeatPassword} = context.params;

// auth.register(username, password, repeatPassword)

// .then(function(res) {

// auth.saveSession(res);

// auth.showInfo('You are registered successfully!');

// that.redirect('#/home');

// });

// });

// });

// app.run('#/home');

// });

const handlers = {};

$(() => {

const app = Sammy('#main', function() {

this.use('Handlebars', 'hbs');

this.get('#/index.html', handlers.homeHandler);

this.get('#/home', handlers.homeHandler);

this.get('#/about', handlers.aboutHandler);

this.get('#/register', handlers.registerGetHandler);

this.post('#/register', handlers.registerPostHandler);

this.get('#/login', handlers.loginGetHandler);

this.post('#/login', handlers.loginPostHandler);

this.get('#/logout', handlers.logoutHandler);

this.get('#/catalog', handlers.listTeamsHandler);

this.get('#/catalog/:teamId', handlers.teamDetailsHandler);

this.get('#/create', handlers.createTeamGetHandler);

this.post('#/create', handlers.createTeamPostHandler);

this.get('#/edit/:teamId', handlers.editTeamGetHandler);

this.post('#/edit/:teamId', handlers.editTeamPostHandler);

this.get('#/join/:teamId', handlers.joinTeamHandler);

this.get('#/leave', handlers.leaveTeamHandler);

});

app.run('#/home');

});

//scripts.models.auth.js

// let auth = (() => {

// function saveSession(userInfo) {

// let userAuth = userInfo.\_kmd.authtoken;

// sessionStorage.setItem('authtoken', userAuth);

// let userId = userInfo.\_id;

// sessionStorage.setItem('userId', userId);

// let username = userInfo.username;

// sessionStorage.setItem('username', username);

// sessionStorage.setItem('teamId', userInfo.teamId);

// }

// // user/login

// function login(username, password) {

// let userData = {

// username,

// password

// };

// return requester.post('user', 'login', 'basic', userData);

// }

// // user/register

// function register(username, password, repeatPassword) {

// let userData = {

// username,

// password

// };

// return requester.post('user', '', 'basic', userData);

// }

// // user/logout

// function logout() {

// let logoutData = {

// authtoken: sessionStorage.getItem('authtoken')

// };

// return requester.post('user', '\_logout', 'kinvey', logoutData);

// }

// function handleError(reason) {

// showError(reason.responseJSON.description);

// }

// function showInfo(message) {

// let infoBox = $('#infoBox');

// infoBox.text(message);

// infoBox.show();

// setTimeout(() => infoBox.fadeOut(), 3000);

// }

// function showError(message) {

// let errorBox = $('#errorBox');

// errorBox.text(message);

// errorBox.show();

// setTimeout(() => errorBox.fadeOut(), 3000);

// }

// return {

// login,

// register,

// logout,

// saveSession,

// showInfo,

// showError,

// handleError

// }

// })()

let authService = (() => {

const module = 'user';

const basicAuth = 'basic';

const kinveyAuth = 'kinvey';

const loginConst = 'login';

const logoutConst = '\_logout';

function isAuth() {

return sessionStorage.getItem('authToken') !== null;

}

function hasTeam() {

return sessionStorage.getItem('teamId') !== null;

}

function getUserId() {

return sessionStorage.getItem('userId');

}

function saveSession(res) {

sessionStorage.setItem('username', res.username);

sessionStorage.setItem('authToken', res.\_kmd.authtoken);

sessionStorage.setItem('userId', res.\_id);

if (res.teamId !== undefined) {

sessionStorage.setItem('teamId', res.teamId);

}

}

function clearSession() {

sessionStorage.clear();

}

function register(username, password) {

let body = {username: username, password: password};

return requestService.post(module, '', basicAuth, body);

}

function login(username, password) {

let body = {username: username, password: password};

return requestService.post(module, loginConst, basicAuth, body);

}

function logout() {

return requestService.post(module, logoutConst, kinveyAuth, {});

}

return {

isAuth,

hasTeam,

getUserId,

saveSession,

clearSession,

register,

login,

logout

};

})();

//scripts.models.notificationService.js

let notifications = (() => {

function handleError(reason) {

showError(reason.responseJSON.description);

}

function showInfo(message) {

let infoBox = $('#infoBox');

infoBox.text(message);

infoBox.show();

setTimeout(() => infoBox.fadeOut(), 3000);

}

function showError(message) {

let errorBox = $('#errorBox');

errorBox.text(message);

errorBox.show();

setTimeout(() => errorBox.fadeOut(), 3000);

}

return {

handleError,

showInfo,

showError

};

})();

//scrips.models.requester.js

// let requester = (() => {

// const kinveyBaseUrl = "https://baas.kinvey.com/";

// const kinveyAppKey = "kid\_ry7IR9WMe";

// const kinveyAppSecret = "095bdc1164c24d9d865cfad4086e4357";

// // Creates the authentication header

// function makeAuth(type) {

// return type === 'basic'

// ? 'Basic ' + btoa(kinveyAppKey + ':' + kinveyAppSecret)

// : 'Kinvey ' + sessionStorage.getItem('authtoken');

// }

// // Creates request object to kinvey

// function makeRequest(method, module, endpoint, auth) {

// return req = {

// method,

// url: kinveyBaseUrl + module + '/' + kinveyAppKey + '/' + endpoint,

// headers: {

// 'Authorization': makeAuth(auth)

// }

// };

// }

// // Function to return GET promise

// function get (module, endpoint, auth) {

// return $.ajax(makeRequest('GET', module, endpoint, auth));

// }

// // Function to return POST promise

// function post (module, endpoint, auth, data) {

// let req = makeRequest('POST', module, endpoint, auth);

// req.data = data;

// return $.ajax(req);

// }

// // Function to return PUT promise

// function update (module, endpoint, auth, data) {

// let req = makeRequest('PUT', module, endpoint, auth);

// req.data = data;

// return $.ajax(req);

// }

// // Function to return DELETE promise

// function remove (module, endpoint, auth) {

// return $.ajax(makeRequest('DELETE', module, endpoint, auth));

// }

// return {

// get,

// post,

// update,

// remove

// }

// })()

let requestService = (() => {

const appKey = 'kid\_ry7IR9WMe';

const appSecret = '095bdc1164c24d9d865cfad4086e4357';

const baseUrl = 'https://baas.kinvey.com';

function getBasicAuth() {

return {'Authorization': "Basic " + btoa(appKey + ":" + appSecret)};

}

function getKinveyAuth() {

return {'Authorization': "Kinvey " + sessionStorage.getItem('authToken')};

}

function constructRequest(method, module, endpoint, auth) {

return {

method: method,

url: `${baseUrl}/${module}/${appKey}/${endpoint}`,

headers: auth === 'basic' ? getBasicAuth() : getKinveyAuth()

};

}

function get(module, endpoint, auth) {

return $.ajax(constructRequest('GET', module, endpoint, auth));

}

function post(module, endpoint, auth, body) {

let request = constructRequest('POST', module, endpoint, auth);

request.contentType = 'application/json';

request.data = JSON.stringify(body);

return $.ajax(request);

}

function update(module, endpoint, auth, body) {

let request = constructRequest('PUT', module, endpoint, auth);

request.contentType = 'application/json';

request.data = JSON.stringify(body);

return $.ajax(request);

}

function remove(module, endpoint, auth) {

return $.ajax(constructRequest('DELETE', module, endpoint, auth));

}

return {

get,

post,

update,

remove

};

})();

//scripts.models.teamService.js

// let teamsService = (() => {

// function loadTeams() {

// // Request teams from db

// return requester.get('appdata', 'teams', 'kinvey');

// }

// function loadTeamDetails(teamId) {

// return requester.get('appdata', 'teams/' + teamId, 'kinvey');

// }

// function edit(teamId, name, description) {

// let teamData = {

// name: name,

// comment: description,

// author: sessionStorage.getItem('username')

// };

// return requester.update('appdata', 'teams/' + teamId, 'kinvey', teamData);

// }

// function createTeam(name, comment) {

// let teamData = {

// name: name,

// comment: comment

// };

// return requester.post('appdata', 'teams', 'kinvey', teamData);

// }

// function joinTeam(teamId) {

// let userData = {

// username: sessionStorage.getItem('username'),

// teamId: teamId

// };

// return requester.update('user', sessionStorage.getItem('userId'), 'kinvey', userData);

// }

// function leaveTeam() {

// let userData = {

// username: sessionStorage.getItem('username'),

// teamId: ''

// };

// return requester.update('user', sessionStorage.getItem('userId'), userData, 'kinvey');

// }

// return {

// loadTeams,

// loadTeamDetails,

// edit,

// createTeam,

// joinTeam,

// leaveTeam

// }

// })()

let crudService = (() => {

const appDataModule = 'appdata';

const userModule = 'user';

const auth = 'kinvey';

const entity = 'teams/';

function loadTeams() {

return requestService.get(appDataModule, entity, auth);

}

function loadTeamDetails(teamId) {

return requestService.get(appDataModule, entity + teamId, auth);

}

function loadTeamMembers(teamId) {

let query = `?query={"teamId":"${teamId}"}`;

return requestService.get(userModule, query, auth);

}

function editTeam(teamId, name, description) {

let teamData = {

name: name,

comment: description

};

return requestService.update(appDataModule, entity + teamId, auth, teamData);

}

function createTeam(name, comment) {

let teamData = {

name: name,

comment: comment

};

return requestService.post(appDataModule, entity, auth, teamData);

}

function joinTeam(teamId) {

let userData = {

username: sessionStorage.getItem('username'),

teamId: teamId

};

return requestService.update(userModule, sessionStorage.getItem('userId'), auth, userData);

}

function leaveTeam() {

let userData = {

username: sessionStorage.getItem('username')

};

return requestService.update(userModule, sessionStorage.getItem('userId'), auth, userData);

}

return {

loadTeams,

loadTeamDetails,

loadTeamMembers,

editTeam,

createTeam,

joinTeam,

leaveTeam

};

})();

//scripts.viewHandlers.authHandler.js

handlers.registerGetHandler = function(context) {

context.loggedIn = authService.isAuth();

context.username = sessionStorage.getItem('username');

context.loadPartials({

header: './templates/common/header.hbs',

footer: './templates/common/footer.hbs',

registerForm: './templates/register/registerForm.hbs'

}).then(function() {

this.partial('./templates/register/registerPage.hbs');

});

};

handlers.registerPostHandler = function(context) {

let username = context.params.username;

let password = context.params.password;

let repeatPassword = context.params.repeatPassword;

if (password !== repeatPassword){

notifications.showError('Passwords do not match!');

}

else {

authService.register(username, password, context)

.then(response => {

authService.saveSession(response);

notifications.showInfo('Registration Successfull.');

context.redirect('#/home');

})

.catch(notifications.handleError);

}

};

handlers.loginGetHandler = function(context) {

context.loggedIn = authService.isAuth();

context.username = sessionStorage.getItem('username');

context.loadPartials({

header: './templates/common/header.hbs',

footer: './templates/common/footer.hbs',

loginForm: './templates/login/loginForm.hbs'

}).then(function() {

this.partial('./templates/login/loginPage.hbs');

});

};

handlers.loginPostHandler = function(context) {

let username = context.params.username;

let password = context.params.password;

authService.login(username, password)

.then(response => {

authService.saveSession(response);

notifications.showInfo('Login successfull.');

context.redirect('#/home');

}).catch(notifications.handleError);

};

handlers.logoutHandler = function(context) {

authService.logout()

.then(() => {

authService.clearSession();

notifications.showInfo('Logout successfull.');

context.redirect('#/home');

}).catch(notifications.handleError);

};

//scripts.viewHandlers.crudHandler.js

handlers.listTeamsHandler = function(context) {

crudService.loadTeams().then(response => {

context.loggedIn = authService.isAuth();

context.username = sessionStorage.getItem('username');

context.hasNoTeam = !authService.hasTeam();

context.teams = response;

notifications.showInfo('Teams listed.');

context.loadPartials({

header: './templates/common/header.hbs',

footer: './templates/common/footer.hbs',

team: './templates/catalog/team.hbs'

}).then(function() {

this.partial('./templates/catalog/teamCatalog.hbs');

});

}).catch(notifications.handleError);

};

handlers.teamDetailsHandler = function(context) {

let teamId = context.params.teamId.slice(1);

let teamInfo;

let membersInfo;

getTeamInfo().then(() => {

context.loggedIn = authService.isAuth();

context.username = sessionStorage.getItem('username');

context.hasNoTeam = !authService.hasTeam();

context.isOnTeam = authService.hasTeam();

context.isAuthor = authService.getUserId() === teamInfo.\_acl.creator;

context.name = teamInfo.name;

context.comment = teamInfo.comment;

context.members = membersInfo;

context.teamId = teamInfo.\_id;

notifications.showInfo(`Details for ${context.name} listed.`);

context.loadPartials({

header: './templates/common/header.hbs',

footer: './templates/common/footer.hbs',

teamControls: './templates/catalog/teamControls.hbs',

teamMember: './templates/catalog/teamMember.hbs'

}).then(function() {

this.partial('./templates/catalog/details.hbs');

});

}).catch(notifications.handleError);

async function getTeamInfo() {

let [teamDetails, teamMembers] = await Promise.all([

crudService.loadTeamDetails(teamId),

crudService.loadTeamMembers(teamId)

]);

teamInfo = teamDetails;

membersInfo = teamMembers;

}

};

handlers.createTeamGetHandler = function(context) {

context.loggedIn = authService.isAuth();

context.username = sessionStorage.getItem('username');

context.loadPartials({

header: './templates/common/header.hbs',

footer: './templates/common/footer.hbs',

createForm: './templates/create/createForm.hbs'

}).then(function() {

this.partial('./templates/create/createPage.hbs');

});

};

handlers.createTeamPostHandler = function(context) {

let name = context.params.name;

let comment = context.params.comment;

crudService.createTeam(name, comment).then(response => {

crudService.joinTeam(response.\_id).then(() => {

sessionStorage.setItem('teamId', response.\_id);

notifications.showInfo(`Team ${name} successfully created.`);

context.redirect('#/home');

}).catch(notifications.handleError);

}).catch(notifications.handleError);

};

handlers.editTeamGetHandler = function(context) {

context.loggedIn = authService.isAuth();

context.username = sessionStorage.getItem('username');

context.teamId = context.params.teamId.slice(1);

context.loadPartials({

header: './templates/common/header.hbs',

footer: './templates/common/footer.hbs',

editForm: './templates/edit/editForm.hbs'

}).then(function() {

this.partial('./templates/edit/editPage.hbs');

});

};

handlers.editTeamPostHandler = function(context) {

let teamId = context.params.teamId.slice(1);

let name = context.params.name;

let comment = context.params.comment;

crudService.editTeam(teamId, name, comment).then(() => {

notifications.showInfo(`Team ${name} successfully edited.`);

context.redirect('#/home');

}).catch(notifications.handleError);

};

handlers.joinTeamHandler = function(context) {

let teamId = context.teamId = context.params.teamId.slice(1);

crudService.joinTeam(teamId).then(response => {

sessionStorage.setItem('teamId', response.teamId);

notifications.showInfo('Join successfull.');

context.redirect('#/home');

}).catch(notifications.handleError);

};

handlers.leaveTeamHandler = function(context) {

crudService.leaveTeam().then(() => {

sessionStorage.removeItem('itemId');

notifications.showInfo('Team successfully left.');

context.redirect('#/home');

}).catch(notifications.handleError);

};

//scripts.viewHandlers.homeHandler.js

handlers.homeHandler = function(context) {

context.loggedIn = authService.isAuth();

context.username = sessionStorage.getItem('username');

if (authService.hasTeam()){

context.hasTeam = authService.hasTeam();

context.teamId = sessionStorage.getItem('teamId');

}

context.loadPartials({

header: './templates/common/header.hbs',

footer: './templates/common/footer.hbs'

}).then(function() {

this.partial('./templates/home/home.hbs');

});

};

handlers.aboutHandler = function(context) {

context.loggedIn = authService.isAuth();

context.username = sessionStorage.getItem('username');

context.loadPartials({

header: './templates/common/header.hbs',

footer: './templates/common/footer.hbs'

}).then(function() {

this.partial('./templates/about/about.hbs');

});

};

//templates.about.about.hbs

{{!-- <div class="container">

{{> header}}

<h2>About Page</h2>

</div>

{{> footer}} --}}

<div class="container">

     {{> header}}

     <h1>Team Manager Homework</h1>

     <p>Task done, using: SammyJS, Handlebars, JQuery</p>

     <p>JavaScript Design Patterns: Revealing modules pattern, MVVM</p>

    </div>

    {{> footer}}

//templates.catalog.details.hbs

<div class="container">

{{> header}}

<div class="details-box">

<span class="titlebar">{{name}} details</span>

<span class="spanner">Team members</span>

<div>

{{#each members}}

{{> teamMember}}

{{else}}

<p>No member info</p>

{{/each}}

</div>

<span class="spanner">Description</span>

{{#if comment}}

<p>{{comment}}</p>

{{else}}

<p>No description</p>

{{/if}}

<span class="spanner">Team management</span>

{{> teamControls}}

</div>

</div>

{{> footer}}

//templates.catalog.team.hbs

<a href="#/catalog/:{{\_id}}" class="team-box">

<span class="spanner">Team name</span>

<span class="title">{{name}}</span>

<span class="spanner">Description</span>

{{#if comment}}

<p>{{comment}}</p>

{{else}}

<p>No description</p>

{{/if}}

</a>

//templates.catalog.teamCatalog.hbs

<div class="container">

{{> header}}

{{#if hasNoTeam}}

<a href="#/create" class="btn btn-default">Create Team</a>

{{/if}}

<h1>Catalog Page</h1>

<div>

{{#each teams}}

{{> team}}

{{/each}}

</div>

</div>

{{> footer}}

//templates.catalog.teamControls.hbs

{{#if isAuthor}}

<a href="#/edit/:{{teamId}}" class="btn btn-default">Edit Info</a>

{{/if}}

{{#if isOnTeam}}

<a href="#/leave" class="btn btn-default">Leave Team</a>

{{else}}

<a href="#/join/:{{teamId}}" class="btn btn-default">Join Team</a>

{{/if}}

//templates.catalog.teamMember.hbs

<div class="member">{{username}}</div>

//templates.common.footer.hbs

<footer>Team Manager SPA &copy; 2019</footer>

//templates.common.header.hbs

<header>

<div class="jumbotron">

<h2>Team Manager</h2>

{{#if loggedIn}}

<span>Welcome, {{username}}</span>

{{/if}}

<nav>

<a href="#/home" class="btn btn-default">Home</a>

<a href="#/about" class="btn btn-default">About</a>

{{#if loggedIn}}

<a href="#/catalog" class="btn btn-default">Catalog</a>

<a href="#/logout" class="btn btn-default">Logout</a>

{{else}}

<a href="#/login" class="btn btn-default">Login</a>

<a href="#/register" class="btn btn-default">Register</a>

{{/if}}

</nav>

</div>

</header>

//templates.create.createForm.hbs

<form action="#/create" method="post">

<div class="form-group">

<label for="name">Team Name:</label>

<input class="form-control" type="text" id="name" name="name"/>

</div>

<div class="form-group">

<label for="comment">Team Comment/Description: </label>

<input class="form-control" type="text" id="comment" name="comment"/>

</div>

<input class="btn btn-default" type="submit" value="Create Team"/>

</form>

//templates.create.createPage.hbs

<div class="container">

{{> header}}

<div>

<h2>Create Page</h2>

{{> createForm}}

</div>

</div>

{{> footer}}

//templates.edit.editForm.html

<form action="#/edit/:{{teamId}}" method="post">

<div class="form-group">

<label for="name">New Team Name:</label>

<input class="form-control" type="text" id="name" name="name" value="{{name}}"/>

</div>

<div class="form-group">

<label for="comment">New Team Comment/Description: </label>

<input class="form-control" type="text" id="comment" name="comment" value="{{comment}}"/>

</div>

<input class="btn btn-default" type="submit" value="Update Team"/>

</form>

//templates.edit.editPage.hbs

<div class="container">

{{> header}}

<div>

<h2>Edit Page</h2>

{{> editForm}}

</div>

</div>

{{> footer}}

//templates.home.home.hbs

<div class="container">

{{> header}}

<div>

<h2>Home Page</h2>

{{#if loggedIn}}

{{#if hasTeam}}

<a href="#/catalog/:{{teamId}}">Go To Team</a>

{{else}}

<p>You are currently not a member of a team. View the <a href="#/catalog">catalog</a> to join or create one.</p>

{{/if}}

{{else}}

<p>You are currently not logged in. <a href="#/login">Log in</a> / <a href="#/register">Register</a></p>

{{/if}}

</div>

</div>

{{> footer}}

//templates.login.loginForm.hbs

<form action="#/login" method="post">

<div class="form-group">

<label for="username">Username:</label>

<input class="form-control" type="text" id="username" name="username"/>

</div>

<div class="form-group">

<label for="password">Password:</label>

<input class="form-control" type="password" id="password" name="password"/>

</div>

<input class="btn btn-default" type="submit" value="Login"/>

</form>

//templates.login.loginPage.hbs

<div class="container">

{{> header}}

<h2>Login Page</h2>

{{> loginForm}}

</div>

{{> footer}}

//templates.register.registerForm.hbs

<form action="#/register" method="post">

<div class="form-group">

<label for="username">Username:</label>

<input class="form-control" type="text" id="username" name="username"/>

</div>

<div class="form-group">

<label for="password">Password:</label>

<input class="form-control" type="password" id="password" name="password"/>

</div>

<div class="form-group">

<label for="repeatPassword">Repeat Password:</label>

<input class="form-control" type="password" id="repeatPassword" name="repeatPassword"/>

</div>

<input class="btn btn-default" type="submit" value="Register"/>

</form>

//templates.register.registerPage.hbs

<div class="container">

{{> header}}

<h2>Register Page</h2>

{{> registerForm}}

</div>

{{> footer}}

## // ****index.html****

|  |
| --- |
| <!doctype html> |
|  | <html lang="en"> |
|  | <head> |
|  | <meta charset="utf-8"> |
|  | <title>Team Managers App</title> |
|  | <meta name="viewport" content="width=device-width, initial-scale=1"> |
|  | <link rel="stylesheet" type="text/css" href="css/index.css"> |
|  | <link rel="stylesheet" type="text/css" href="css/team.css"> |
|  | <link rel="stylesheet" type="text/css" href="css/details.css"> |
|  | <link rel="stylesheet" type="text/css" href="node\_modules/bootstrap/dist/css/bootstrap.min.css"> |
|  | <script src="node\_modules/jquery/dist/jquery.min.js"></script> |
|  | <script src="node\_modules/bootstrap/dist/js/bootstrap.min.js"></script> |
|  | <script src="node\_modules/handlebars/dist/handlebars.min.js"></script> |
|  | <script src="node\_modules/sammy/lib/min/sammy-latest.min.js"></script> |
|  | <script src="node\_modules/sammy/lib/min/plugins/sammy.handlebars-latest.min.js"></script> |
|  | <script src="scripts/models/requestService.js"></script> |
|  | <script src="scripts/models/notificationService.js"></script> |
|  | <script src="scripts/models/authService.js"></script> |
|  | <script src="scripts/models/crudService.js"></script> |
|  | <script src="scripts/app.js"></script> |
|  | <script src="scripts/viewHandlers/homeHandler.js"></script> |
|  | <script src="scripts/viewHandlers/authHandler.js"></script> |
|  | <script src="scripts/viewHandlers/crudHandler.js"></script> |
|  | </head> |
|  | <body> |
|  | <div id="infoBox" class="notification" disabled></div> |
|  | <div id="errorBox" class="notification" disabled></div> |
|  | <div id="main"></div> |
|  | </body> |
|  | </html> |

## // ****package-lock.json****

|  |
| --- |
| { |
|  | "name": "P01-Team-Manager", |
|  | "version": "0.0.1", |
|  | "lockfileVersion": 1, |
|  | "requires": true, |
|  | "dependencies": { |
|  | "align-text": { |
|  | "version": "0.1.4", |
|  | "resolved": "https://registry.npmjs.org/align-text/-/align-text-0.1.4.tgz", |
|  | "integrity": "sha1-DNkKVhCT810KmSVsIrcGlDP60Rc=", |
|  | "requires": { |
|  | "kind-of": "3.2.2", |
|  | "longest": "1.0.1", |
|  | "repeat-string": "1.6.1" |
|  | } |
|  | }, |
|  | "amdefine": { |
|  | "version": "1.0.1", |
|  | "resolved": "https://registry.npmjs.org/amdefine/-/amdefine-1.0.1.tgz", |
|  | "integrity": "sha1-SlKCrBZHKek2Gbz9OtFR+BfOkfU=" |
|  | }, |
|  | "async": { |
|  | "version": "1.5.2", |
|  | "resolved": "https://registry.npmjs.org/async/-/async-1.5.2.tgz", |
|  | "integrity": "sha1-7GphrlZIDAw8skHJVhjiCJL5Zyo=" |
|  | }, |
|  | "bootstrap": { |
|  | "version": "4.0.0", |
|  | "resolved": "https://registry.npmjs.org/bootstrap/-/bootstrap-4.0.0.tgz", |
|  | "integrity": "sha512-gulJE5dGFo6Q61V/whS6VM4WIyrlydXfCgkE+Gxe5hjrJ8rXLLZlALq7zq2RPhOc45PSwQpJkrTnc2KgD6cvmA==" |
|  | }, |
|  | "camelcase": { |
|  | "version": "1.2.1", |
|  | "resolved": "https://registry.npmjs.org/camelcase/-/camelcase-1.2.1.tgz", |
|  | "integrity": "sha1-m7UwTS4LVmmLLHWLCKPqqdqlijk=", |
|  | "optional": true |
|  | }, |
|  | "center-align": { |
|  | "version": "0.1.3", |
|  | "resolved": "https://registry.npmjs.org/center-align/-/center-align-0.1.3.tgz", |
|  | "integrity": "sha1-qg0yYptu6XIgBBHL1EYckHvCt60=", |
|  | "optional": true, |
|  | "requires": { |
|  | "align-text": "0.1.4", |
|  | "lazy-cache": "1.0.4" |
|  | } |
|  | }, |
|  | "cliui": { |
|  | "version": "2.1.0", |
|  | "resolved": "https://registry.npmjs.org/cliui/-/cliui-2.1.0.tgz", |
|  | "integrity": "sha1-S0dXYP+AJkx2LDoXGQMukcf+oNE=", |
|  | "optional": true, |
|  | "requires": { |
|  | "center-align": "0.1.3", |
|  | "right-align": "0.1.3", |
|  | "wordwrap": "0.0.2" |
|  | }, |
|  | "dependencies": { |
|  | "wordwrap": { |
|  | "version": "0.0.2", |
|  | "resolved": "https://registry.npmjs.org/wordwrap/-/wordwrap-0.0.2.tgz", |
|  | "integrity": "sha1-t5Zpu0LstAn4PVg8rVLKF+qhZD8=", |
|  | "optional": true |
|  | } |
|  | } |
|  | }, |
|  | "decamelize": { |
|  | "version": "1.2.0", |
|  | "resolved": "https://registry.npmjs.org/decamelize/-/decamelize-1.2.0.tgz", |
|  | "integrity": "sha1-9lNNFRSCabIDUue+4m9QH5oZEpA=", |
|  | "optional": true |
|  | }, |
|  | "handlebars": { |
|  | "version": "4.0.11", |
|  | "resolved": "https://registry.npmjs.org/handlebars/-/handlebars-4.0.11.tgz", |
|  | "integrity": "sha1-Ywo13+ApS8KB7a5v/F0yn8eYLcw=", |
|  | "requires": { |
|  | "async": "1.5.2", |
|  | "optimist": "0.6.1", |
|  | "source-map": "0.4.4", |
|  | "uglify-js": "2.8.29" |
|  | } |
|  | }, |
|  | "is-buffer": { |
|  | "version": "1.1.6", |
|  | "resolved": "https://registry.npmjs.org/is-buffer/-/is-buffer-1.1.6.tgz", |
|  | "integrity": "sha512-NcdALwpXkTm5Zvvbk7owOUSvVvBKDgKP5/ewfXEznmQFfs4ZRmanOeKBTjRVjka3QFoN6XJ+9F3USqfHqTaU5w==" |
|  | }, |
|  | "jquery": { |
|  | "version": "3.3.1", |
|  | "resolved": "https://registry.npmjs.org/jquery/-/jquery-3.3.1.tgz", |
|  | "integrity": "sha512-Ubldcmxp5np52/ENotGxlLe6aGMvmF4R8S6tZjsP6Knsaxd/xp3Zrh50cG93lR6nPXyUFwzN3ZSOQI0wRJNdGg==" |
|  | }, |
|  | "kind-of": { |
|  | "version": "3.2.2", |
|  | "resolved": "https://registry.npmjs.org/kind-of/-/kind-of-3.2.2.tgz", |
|  | "integrity": "sha1-MeohpzS6ubuw8yRm2JOupR5KPGQ=", |
|  | "requires": { |
|  | "is-buffer": "1.1.6" |
|  | } |
|  | }, |
|  | "lazy-cache": { |
|  | "version": "1.0.4", |
|  | "resolved": "https://registry.npmjs.org/lazy-cache/-/lazy-cache-1.0.4.tgz", |
|  | "integrity": "sha1-odePw6UEdMuAhF07O24dpJpEbo4=", |
|  | "optional": true |
|  | }, |
|  | "longest": { |
|  | "version": "1.0.1", |
|  | "resolved": "https://registry.npmjs.org/longest/-/longest-1.0.1.tgz", |
|  | "integrity": "sha1-MKCy2jj3N3DoKUoNIuZiXtd9AJc=" |
|  | }, |
|  | "minimist": { |
|  | "version": "0.0.10", |
|  | "resolved": "https://registry.npmjs.org/minimist/-/minimist-0.0.10.tgz", |
|  | "integrity": "sha1-3j+YVD2/lggr5IrRoMfNqDYwHc8=" |
|  | }, |
|  | "optimist": { |
|  | "version": "0.6.1", |
|  | "resolved": "https://registry.npmjs.org/optimist/-/optimist-0.6.1.tgz", |
|  | "integrity": "sha1-2j6nRob6IaGaERwybpDrFaAZZoY=", |
|  | "requires": { |
|  | "minimist": "0.0.10", |
|  | "wordwrap": "0.0.3" |
|  | } |
|  | }, |
|  | "repeat-string": { |
|  | "version": "1.6.1", |
|  | "resolved": "https://registry.npmjs.org/repeat-string/-/repeat-string-1.6.1.tgz", |
|  | "integrity": "sha1-jcrkcOHIirwtYA//Sndihtp15jc=" |
|  | }, |
|  | "right-align": { |
|  | "version": "0.1.3", |
|  | "resolved": "https://registry.npmjs.org/right-align/-/right-align-0.1.3.tgz", |
|  | "integrity": "sha1-YTObci/mo1FWiSENJOFMlhSGE+8=", |
|  | "optional": true, |
|  | "requires": { |
|  | "align-text": "0.1.4" |
|  | } |
|  | }, |
|  | "sammy": { |
|  | "version": "0.7.6", |
|  | "resolved": "https://registry.npmjs.org/sammy/-/sammy-0.7.6.tgz", |
|  | "integrity": "sha1-Rk/lTR+JZhzDqYgAYwVxEyHpqcE=" |
|  | }, |
|  | "source-map": { |
|  | "version": "0.4.4", |
|  | "resolved": "https://registry.npmjs.org/source-map/-/source-map-0.4.4.tgz", |
|  | "integrity": "sha1-66T12pwNyZneaAMti092FzZSA2s=", |
|  | "requires": { |
|  | "amdefine": "1.0.1" |
|  | } |
|  | }, |
|  | "uglify-js": { |
|  | "version": "2.8.29", |
|  | "resolved": "https://registry.npmjs.org/uglify-js/-/uglify-js-2.8.29.tgz", |
|  | "integrity": "sha1-KcVzMUgFe7Th913zW3qcty5qWd0=", |
|  | "optional": true, |
|  | "requires": { |
|  | "source-map": "0.5.7", |
|  | "uglify-to-browserify": "1.0.2", |
|  | "yargs": "3.10.0" |
|  | }, |
|  | "dependencies": { |
|  | "source-map": { |
|  | "version": "0.5.7", |
|  | "resolved": "https://registry.npmjs.org/source-map/-/source-map-0.5.7.tgz", |
|  | "integrity": "sha1-igOdLRAh0i0eoUyA2OpGi6LvP8w=", |
|  | "optional": true |
|  | } |
|  | } |
|  | }, |
|  | "uglify-to-browserify": { |
|  | "version": "1.0.2", |
|  | "resolved": "https://registry.npmjs.org/uglify-to-browserify/-/uglify-to-browserify-1.0.2.tgz", |
|  | "integrity": "sha1-bgkk1r2mta/jSeOabWMoUKD4grc=", |
|  | "optional": true |
|  | }, |
|  | "window-size": { |
|  | "version": "0.1.0", |
|  | "resolved": "https://registry.npmjs.org/window-size/-/window-size-0.1.0.tgz", |
|  | "integrity": "sha1-VDjNLqk7IC76Ohn+iIeu58lPnJ0=", |
|  | "optional": true |
|  | }, |
|  | "wordwrap": { |
|  | "version": "0.0.3", |
|  | "resolved": "https://registry.npmjs.org/wordwrap/-/wordwrap-0.0.3.tgz", |
|  | "integrity": "sha1-o9XabNXAvAAI03I0u68b7WMFkQc=" |
|  | }, |
|  | "yargs": { |
|  | "version": "3.10.0", |
|  | "resolved": "https://registry.npmjs.org/yargs/-/yargs-3.10.0.tgz", |
|  | "integrity": "sha1-9+572FfdfB0tOMDnTvvWgdFDH9E=", |
|  | "optional": true, |
|  | "requires": { |
|  | "camelcase": "1.2.1", |
|  | "cliui": "2.1.0", |
|  | "decamelize": "1.2.0", |
|  | "window-size": "0.1.0" |
|  | } |
|  | } |
|  | } |
|  | } |

## // ****package.json****

|  |
| --- |
| { |
|  | "name": "P01-Team-Manager", |
|  | "version": "0.0.1", |
|  | "dependencies": { |
|  | "jquery": "latest", |
|  | "handlebars": "latest", |
|  | "sammy": "latest", |
|  | "bootstrap": "latest" |
|  | } |
|  | } |

## // ****css**. **details.css****

|  |
| --- |
| .details-box p { |
|  | margin: 1.5em; |
|  | } |
|  |  |
|  | .member { |
|  | font-size: 1.5em; |
|  | margin: 0.25em 0.25em 0.25em 1em; |
|  | list-style-type: none; |
|  | display: block; |
|  | } |
|  |  |
|  | .titlebar { |
|  | display: block; |
|  | font-size: 2em; |
|  | padding: 0.25em 0.25em 0.25em 0.75em; |
|  | } |
|  |  |
|  | .spanner { |
|  | display: block; |
|  | padding: 0.25em 0.25em 0.25em 1.5em; |
|  | background-color: antiquewhite; |
|  | font-weight: 600; |
|  | } |

## ****//css.index.css****

|  |
| --- |
| body { |
|  | margin: 0; |
|  | padding: 0; |
|  | font-family: sans-serif; |
|  | } |
|  |  |
|  | #main { |
|  | text-align: center; |
|  | } |
|  |  |
|  | .header { |
|  | background-color: #222; |
|  | height: 150px; |
|  | padding: 20px; |
|  | color: white; |
|  | } |
|  |  |
|  | .footer { |
|  | text-align: center; |
|  | position: fixed; |
|  | } |
|  |  |
|  | .notification { |
|  | font-size: 1.5em; |
|  | float: right; |
|  | color: #ffffff; |
|  | padding: 0.5em 2em 0.5em 2em; |
|  | margin: 1em; |
|  | display: none; |
|  | position: fixed; |
|  | top: 0; |
|  | right: 0; |
|  | text-align: center; |
|  | border-radius: 3px; |
|  | } |
|  |  |
|  | #infoBox { |
|  | background: #393; |
|  | } |
|  |  |
|  | #errorBox { |
|  | background: #F50; |
|  | } |

## **//css.**team.css****

|  |
| --- |
| .team-box { |
|  | float: left; |
|  | width: 20em; |
|  | margin: 1em; |
|  | border: 3px solid antiquewhite; |
|  | text-decoration: none; |
|  | color: black; |
|  | } |
|  |  |
|  | .team-box:hover { |
|  | border: 3px solid black; |
|  | text-decoration: none; |
|  | color: black; |
|  | } |
|  |  |
|  | .spanner { |
|  | display: block; |
|  | padding: 0.25em; |
|  | background-color: antiquewhite; |
|  | } |
|  |  |
|  | .title { |
|  | font-size: 1.5em; |
|  | display: block; |
|  | padding: 0.25em 0.25em 0.25em 1.5em; |
|  | font-weight: 600; |
|  | } |
|  |  |
|  | .team-box p { |
|  | margin: 1.5em; |
|  | } |

## **//scripts.**app.js****

|  |
| --- |
| const HANDLERS = {}; |
|  |  |
|  | $(() => { |
|  | const APP = Sammy('#main', function () { |
|  | this.use('Handlebars', 'hbs'); |
|  |  |
|  | this.get('#/index.html', HANDLERS.homeHandler); |
|  | this.get('#/home', HANDLERS.homeHandler); |
|  | this.get('#/about', HANDLERS.aboutHandler); |
|  |  |
|  | this.get('#/register', HANDLERS.registerGetHandler); |
|  | this.post('#/register', HANDLERS.registerPostHandler); |
|  |  |
|  | this.get('#/login', HANDLERS.loginGetHandler); |
|  | this.post('#/login', HANDLERS.loginPostHandler); |
|  |  |
|  | this.get('#/logout', HANDLERS.logoutHandler); |
|  |  |
|  | this.get('#/catalog', HANDLERS.listTeamsHandler); |
|  | this.get('#/catalog/:teamId', HANDLERS.teamDetailsHandler); |
|  |  |
|  | this.get('#/create', HANDLERS.createTeamGetHandler); |
|  | this.post('#/create', HANDLERS.createTeamPostHandler); |
|  |  |
|  | this.get('#/edit/:teamId', HANDLERS.editTeamGetHandler); |
|  | this.post('#/edit/:teamId', HANDLERS.editTeamPostHandler); |
|  |  |
|  | this.get('#/join/:teamId', HANDLERS.joinTeamHandler); |
|  | this.get('#/leave', HANDLERS.leaveTeamHandler); |
|  | }); |
|  |  |
|  | APP.run(); |
|  | }); |

## **//scripts.models.**authService.js****

|  |
| --- |
| let authService = (() => { |
|  | const MODULE = 'user'; |
|  | const BASIC\_AUTH = 'basic'; |
|  | const KINVEY\_AUTH = 'kinvey'; |
|  | const LOGIN = 'login'; |
|  | const LOGOUT = '\_logout'; |
|  |  |
|  | function isAuth() { |
|  | return sessionStorage.getItem('authToken') !== null; |
|  | } |
|  |  |
|  | function hasTeam() { |
|  | return sessionStorage.getItem('teamId') !== null; |
|  | } |
|  |  |
|  | function getUserId() { |
|  | return sessionStorage.getItem('userId'); |
|  | } |
|  |  |
|  | function saveSession(res) { |
|  | sessionStorage.setItem('username', res.username); |
|  | sessionStorage.setItem('authToken', res.\_kmd.authtoken); |
|  | sessionStorage.setItem('userId', res.\_id); |
|  | if (res.teamId !== undefined) { |
|  | sessionStorage.setItem('teamId', res.teamId); |
|  | } |
|  | } |
|  |  |
|  | function clearSession() { |
|  | sessionStorage.clear(); |
|  | } |
|  |  |
|  | function register(username, password) { |
|  | let body = {username: username, password: password}; |
|  | return requestService.post(MODULE, '', BASIC\_AUTH, body); |
|  | } |
|  |  |
|  | function login(username, password) { |
|  | let body = {username: username, password: password}; |
|  | return requestService.post(MODULE, LOGIN, BASIC\_AUTH, body); |
|  | } |
|  |  |
|  | function logout() { |
|  | return requestService.post(MODULE, LOGOUT, KINVEY\_AUTH, {}); |
|  | } |
|  |  |
|  | return { |
|  | isAuth, |
|  | hasTeam, |
|  | getUserId, |
|  | saveSession, |
|  | clearSession, |
|  | register, |
|  | login, |
|  | logout |
|  | }; |
|  | })(); |

## **//scripts.models.**crudService.js****

|  |
| --- |
| let crudService = (() => { |
|  | const APPDATA\_MODULE = 'appdata'; |
|  | const USER\_MODULE = 'user'; |
|  | const AUTH = 'kinvey'; |
|  | const ENTITY = 'teams/'; |
|  |  |
|  | function loadTeams() { |
|  | return requestService.get(APPDATA\_MODULE, ENTITY, AUTH); |
|  | } |
|  |  |
|  | function loadTeamDetails(teamId) { |
|  | return requestService.get(APPDATA\_MODULE, ENTITY + teamId, AUTH); |
|  | } |
|  |  |
|  | function loadTeamMembers(teamId) { |
|  | let query = `?query={"teamId":"${teamId}"}`; |
|  | return requestService.get(USER\_MODULE, query, AUTH); |
|  | } |
|  |  |
|  | function editTeam(teamId, name, description) { |
|  | let teamData = { |
|  | name: name, |
|  | comment: description |
|  | }; |
|  |  |
|  | return requestService.update(APPDATA\_MODULE, ENTITY + teamId, AUTH, teamData); |
|  | } |
|  |  |
|  | function createTeam(name, comment) { |
|  | let teamData = { |
|  | name: name, |
|  | comment: comment |
|  | }; |
|  |  |
|  | return requestService.post(APPDATA\_MODULE, ENTITY, AUTH, teamData); |
|  | } |
|  |  |
|  | function joinTeam(teamId) { |
|  | let userData = { |
|  | username: sessionStorage.getItem('username'), |
|  | teamId: teamId |
|  | }; |
|  |  |
|  | return requestService.update(USER\_MODULE, sessionStorage.getItem('userId'), AUTH, userData); |
|  | } |
|  |  |
|  | function leaveTeam() { |
|  | let userData = { |
|  | username: sessionStorage.getItem('username') |
|  | }; |
|  |  |
|  | return requestService.update(USER\_MODULE, sessionStorage.getItem('userId'), AUTH, userData); |
|  | } |
|  |  |
|  | return { |
|  | loadTeams, |
|  | loadTeamDetails, |
|  | loadTeamMembers, |
|  | editTeam, |
|  | createTeam, |
|  | joinTeam, |
|  | leaveTeam |
|  | }; |
|  | })(); |

## **//scripts.models.**notificationService.js****

|  |
| --- |
| let notifications = (() => { |
|  | function handleError(reason) { |
|  | showError(reason.responseJSON.description); |
|  | } |
|  |  |
|  | function showInfo(message) { |
|  | let infoBox = $('#infoBox'); |
|  | infoBox.text(message); |
|  | infoBox.show(); |
|  | setTimeout(() => infoBox.fadeOut(), 3000); |
|  | } |
|  |  |
|  | function showError(message) { |
|  | let errorBox = $('#errorBox'); |
|  | errorBox.text(message); |
|  | errorBox.show(); |
|  | setTimeout(() => errorBox.fadeOut(), 3000); |
|  | } |
|  |  |
|  | return { |
|  | handleError, |
|  | showInfo, |
|  | showError |
|  | }; |
|  | })(); |

## **//scripts.models.**requestService.js****

|  |
| --- |
| let requestService = (() => { |
|  | const APP\_KEY = 'kid\_ry7IR9WMe'; |
|  | const APP\_SECRET = '095bdc1164c24d9d865cfad4086e4357'; |
|  | const BASE\_URL = 'https://baas.kinvey.com'; |
|  |  |
|  | function getBasicAuth() { |
|  | return {'Authorization': "Basic " + btoa(APP\_KEY + ":" + APP\_SECRET)}; |
|  | } |
|  |  |
|  | function getKinveyAuth() { |
|  | return {'Authorization': "Kinvey " + sessionStorage.getItem('authToken')}; |
|  | } |
|  |  |
|  | function constructRequest(method, module, endpoint, auth) { |
|  | return { |
|  | method: method, |
|  | url: `${BASE\_URL}/${module}/${APP\_KEY}/${endpoint}`, |
|  | headers: auth === 'basic' ? getBasicAuth() : getKinveyAuth() |
|  | }; |
|  | } |
|  |  |
|  | function get(module, endpoint, auth) { |
|  | return $.ajax(constructRequest('GET', module, endpoint, auth)); |
|  | } |
|  |  |
|  | function post(module, endpoint, auth, body) { |
|  | let request = constructRequest('POST', module, endpoint, auth); |
|  | request.contentType = 'application/json'; |
|  | request.data = JSON.stringify(body); |
|  | return $.ajax(request); |
|  | } |
|  |  |
|  | function update(module, endpoint, auth, body) { |
|  | let request = constructRequest('PUT', module, endpoint, auth); |
|  | request.contentType = 'application/json'; |
|  | request.data = JSON.stringify(body); |
|  | return $.ajax(request); |
|  | } |
|  |  |
|  | function remove(module, endpoint, auth) { |
|  | return $.ajax(constructRequest('DELETE', module, endpoint, auth)); |
|  | } |
|  |  |
|  | return { |
|  | get, |
|  | post, |
|  | update, |
|  | remove |
|  | }; |
|  | })(); |

## **//scripts.**viewHandlers.authHandler.js****

|  |
| --- |
| HANDLERS.registerGetHandler = function (context) { |
|  | context.loggedIn = authService.isAuth(); |
|  | context.username = sessionStorage.getItem('username'); |
|  |  |
|  | context.loadPartials({ |
|  | header: './templates/common/header.hbs', |
|  | footer: './templates/common/footer.hbs', |
|  | registerForm: './templates/register/registerForm.hbs', |
|  | }).then(function () { |
|  | this.partial('./templates/register/registerPage.hbs'); |
|  | }); |
|  | }; |
|  |  |
|  | HANDLERS.registerPostHandler = function (context) { |
|  | let username = context.params.username; |
|  | let password = context.params.password; |
|  | let repeatPassword = context.params.repeatPassword; |
|  |  |
|  | if (password !== repeatPassword) { |
|  | notifications.showError('Passwords do not match!'); |
|  | } else { |
|  | authService.register(username, password, context).then(res => { |
|  | authService.saveSession(res); |
|  | notifications.showInfo('Registration successful.'); |
|  | context.redirect('#/home'); |
|  | }).catch(notifications.handleError); |
|  | } |
|  | }; |
|  |  |
|  | HANDLERS.loginGetHandler = function (context) { |
|  | context.loggedIn = authService.isAuth(); |
|  | context.username = sessionStorage.getItem('username'); |
|  |  |
|  | context.loadPartials({ |
|  | header: './templates/common/header.hbs', |
|  | footer: './templates/common/footer.hbs', |
|  | loginForm: './templates/login/loginForm.hbs', |
|  | }).then(function () { |
|  | this.partial('./templates/login/loginPage.hbs'); |
|  | }); |
|  | }; |
|  |  |
|  | HANDLERS.loginPostHandler = function (context) { |
|  | let username = context.params.username; |
|  | let password = context.params.password; |
|  |  |
|  | authService.login(username, password).then(res => { |
|  | authService.saveSession(res); |
|  | notifications.showInfo('Login successful.'); |
|  | context.redirect('#/home'); |
|  | }).catch(notifications.handleError); |
|  | }; |
|  |  |
|  | HANDLERS.logoutHandler = function (context) { |
|  | authService.logout().then(() => { |
|  | authService.clearSession(); |
|  | notifications.showInfo('Logout successful.'); |
|  | context.redirect('#/home'); |
|  | }).catch(notifications.handleError); |
|  | }; |

## **//scripts.**viewHandlers.crudHandler.js****

|  |
| --- |
| HANDLERS.listTeamsHandler = function (context) { |
|  | crudService.loadTeams().then(res => { |
|  | context.loggedIn = authService.isAuth(); |
|  | context.username = sessionStorage.getItem('username'); |
|  | context.hasNoTeam = !authService.hasTeam(); |
|  | context.teams = res; |
|  |  |
|  | notifications.showInfo('Teams listed.'); |
|  | context.loadPartials({ |
|  | header: './templates/common/header.hbs', |
|  | footer: './templates/common/footer.hbs', |
|  | team: './templates/catalog/team.hbs' |
|  | }).then(function () { |
|  | this.partial('./templates/catalog/teamCatalog.hbs'); |
|  | }); |
|  |  |
|  | }).catch(notifications.handleError); |
|  | }; |
|  |  |
|  | HANDLERS.teamDetailsHandler = function (context) { |
|  | let teamId = context.params.teamId.slice(1); |
|  | let teamInfo; |
|  | let membersInfo; |
|  |  |
|  | getTeamInfo().then(() => { |
|  | context.loggedIn = authService.isAuth(); |
|  | context.username = sessionStorage.getItem('username'); |
|  | context.hasNoTeam = !authService.hasTeam(); |
|  | context.isOnTeam = authService.hasTeam(); |
|  | context.isAuthor = authService.getUserId() === teamInfo.\_acl.creator; |
|  | context.name = teamInfo.name; |
|  | context.comment = teamInfo.comment; |
|  | context.members = membersInfo; |
|  | context.teamId = teamInfo.\_id; |
|  |  |
|  | notifications.showInfo(`Details for ${context.name} listed.`); |
|  |  |
|  | context.loadPartials({ |
|  | header: './templates/common/header.hbs', |
|  | footer: './templates/common/footer.hbs', |
|  | teamControls: './templates/catalog/teamControls.hbs', |
|  | teamMember: './templates/catalog/teamMember.hbs' |
|  | }).then(function () { |
|  | this.partial('./templates/catalog/details.hbs'); |
|  | }); |
|  | }).catch(notifications.handleError); |
|  |  |
|  | async function getTeamInfo() { |
|  | let [teamDetails, teamMembers] = await Promise.all([ |
|  | crudService.loadTeamDetails(teamId), |
|  | crudService.loadTeamMembers(teamId), |
|  | ]); |
|  | teamInfo = teamDetails; |
|  | membersInfo = teamMembers; |
|  | } |
|  | }; |
|  |  |
|  | HANDLERS.createTeamGetHandler = function (context) { |
|  | context.loggedIn = authService.isAuth(); |
|  | context.username = sessionStorage.getItem('username'); |
|  |  |
|  | context.loadPartials({ |
|  | header: './templates/common/header.hbs', |
|  | footer: './templates/common/footer.hbs', |
|  | createForm: './templates/create/createForm.hbs' |
|  | }).then(function () { |
|  | this.partial('./templates/create/createPage.hbs'); |
|  | }); |
|  | }; |
|  |  |
|  | HANDLERS.createTeamPostHandler = function (context) { |
|  | let name = context.params.name; |
|  | let comment = context.params.comment; |
|  | crudService.createTeam(name, comment).then(res => { |
|  | crudService.joinTeam(res.\_id).then(() => { |
|  | sessionStorage.setItem('teamId', res.\_id); |
|  | notifications.showInfo(`Team ${name} successfully created.`); |
|  | context.redirect('#/home'); |
|  | }).catch(notifications.handleError); |
|  | }).catch(notifications.handleError); |
|  | }; |
|  |  |
|  | HANDLERS.editTeamGetHandler = function (context) { |
|  | context.loggedIn = authService.isAuth(); |
|  | context.username = sessionStorage.getItem('username'); |
|  | context.teamId = context.params.teamId.slice(1); |
|  |  |
|  | context.loadPartials({ |
|  | header: './templates/common/header.hbs', |
|  | footer: './templates/common/footer.hbs', |
|  | editForm: './templates/edit/editForm.hbs' |
|  | }).then(function () { |
|  | this.partial('./templates/edit/editPage.hbs'); |
|  | }); |
|  | }; |
|  |  |
|  | HANDLERS.editTeamPostHandler = function (context) { |
|  | let teamId = context.params.teamId.slice(1); |
|  | let name = context.params.name; |
|  | let comment = context.params.comment; |
|  |  |
|  | crudService.editTeam(teamId, name, comment).then(() => { |
|  | notifications.showInfo(`Team ${name} successfully edited.`); |
|  | context.redirect('#/home'); |
|  | }).catch(notifications.handleError); |
|  | }; |
|  |  |
|  | HANDLERS.joinTeamHandler = function (context) { |
|  | let teamId = context.teamId = context.params.teamId.slice(1); |
|  | crudService.joinTeam(teamId).then(res => { |
|  | sessionStorage.setItem('teamId', res.teamId); |
|  | notifications.showInfo(`Join successful.`); |
|  | context.redirect('#/home'); |
|  | }).catch(notifications.handleError); |
|  | }; |
|  |  |
|  | HANDLERS.leaveTeamHandler = function (context) { |
|  | crudService.leaveTeam().then(() => { |
|  | sessionStorage.removeItem('teamId'); |
|  | notifications.showInfo(`Team successfully left.`); |
|  | context.redirect('#/home'); |
|  | }).catch(notifications.handleError); |
|  | }; |

## **//**scripts.viewHandlers.homeHandler.js****

|  |
| --- |
| HANDLERS.homeHandler = function (context) { |
|  | context.loggedIn = authService.isAuth(); |
|  | context.username = sessionStorage.getItem('username'); |
|  |  |
|  | if (authService.hasTeam()) { |
|  | context.hasTeam = authService.hasTeam(); |
|  | context.teamId = sessionStorage.getItem('teamId'); |
|  | } |
|  |  |
|  | context.loadPartials({ |
|  | header: './templates/common/header.hbs', |
|  | footer: './templates/common/footer.hbs', |
|  | }).then(function () { |
|  | this.partial('./templates/home/home.hbs'); |
|  | }); |
|  | }; |
|  |  |
|  | HANDLERS.aboutHandler = function (context) { |
|  | context.loggedIn = authService.isAuth(); |
|  | context.username = sessionStorage.getItem('username'); |
|  |  |
|  | context.loadPartials({ |
|  | header: './templates/common/header.hbs', |
|  | footer: './templates/common/footer.hbs', |
|  | }).then(function () { |
|  | this.partial('./templates/about/about.hbs'); |
|  | }); |
|  | }; |

## **//**templates.about.about.hbs****

|  |
| --- |
| <div class="container"> |
|  | {{> header}} |
|  | <h1>Team Manager Homework</h1> |
|  | <p>Task done, using: SammyJS, Handlebars, JQuery</p> |
|  | <p>JavaScript Design Patterns: Revealing modules pattern, MVVM</p> |
|  | </div> |
|  | {{> footer}} |

## **//templates.**catalog.details.hbs****

|  |
| --- |
| <div class="container"> |
|  | {{> header}} |
|  | <div class="details-box"> |
|  | <span class="titlebar">{{name}} details</span> |
|  | <span class="spanner">Team members</span> |
|  | <div> |
|  | {{#each members}} |
|  | {{> teamMember}} |
|  | {{else}} |
|  | <p>No member info</p> |
|  | {{/each}} |
|  | </div> |
|  | <span class="spanner">Description</span> |
|  | {{#if comment}} |
|  | <p>{{comment}}</p> |
|  | {{else}} |
|  | <p>No description</p> |
|  | {{/if}} |
|  | <span class="spanner">Team management</span> |
|  | {{> teamControls}} |
|  | </div> |
|  | </div> |
|  | {{> footer}} |

## **//templates.**catalog.team.hbs****

|  |
| --- |
| <a href="#/catalog/:{{\_id}}" class="team-box"> |
|  | <span class="spanner">Team name</span> |
|  | <span class="title">{{name}}</span> |
|  | <span class="spanner">Description</span> |
|  | {{#if comment}} |
|  | <p>{{comment}}</p> |
|  | {{else}} |
|  | <p>No description</p> |
|  | {{/if}} |
|  | </a> |

## **//templates.**catalog.teamCatalog.hbs****

|  |
| --- |
| <div class="container"> |
|  | {{> header}} |
|  | {{#if hasNoTeam}} |
|  | <a href="#/create" class="btn btn-default">Create Team</a> |
|  | {{/if}} |
|  | <h1>Catalog Page</h1> |
|  | <div> |
|  | {{#each teams}} |
|  | {{> team}} |
|  | {{/each}} |
|  | </div> |
|  | </div> |
|  | {{> footer}} |

## **//templates.**catalog.teamControls.hbs****

|  |
| --- |
| {{#if isAuthor}} |
|  | <a href="#/edit/:{{teamId}}" class="btn btn-default">Edit Info</a> |
|  | {{/if}} |
|  | {{#if isOnTeam}} |
|  | <a href="#/leave" class="btn btn-default">Leave Team</a> |
|  | {{else}} |
|  | <a href="#/join/:{{teamId}}" class="btn btn-default">Join Team</a> |
|  | {{/if}} |

## **//templates.**catalog.teamMember.hbs****

<div class="member">{{username}}</div>

## **//templates.common.**footer.hbs****

<footer>Team Manager SPA &copy; 2017</footer>

## **//templates.common.**header.hbs****

|  |
| --- |
| <header> |
|  | <div class="jumbotron"> |
|  | <h1>Team Manager</h1> |
|  | {{#if loggedIn}} |
|  | <span>Welcome, {{username}}</span> |
|  | {{/if}} |
|  | <nav> |
|  | <a href="#/home" class="btn btn-default">Home</a> |
|  | <a href="#/about" class="btn btn-default">About</a> |
|  | {{#if loggedIn}} |
|  | <a href="#/catalog" class="btn btn-default">Catalog</a> |
|  | <a href="#/logout" class="btn btn-default">Logout</a> |
|  | {{else}} |
|  | <a href="#/login" class="btn btn-default">Login</a> |
|  | <a href="#/register" class="btn btn-default">Register</a> |
|  | {{/if}} |
|  | </nav> |
|  | </div> |
|  | </header> |

## **//templates.**create.createForm.hbs****

|  |
| --- |
| <form action="#/create" method="post"> |
|  | <div class="form-group"> |
|  | <label for="name">Team Name:</label> |
|  | <input class="form-control" type="text" id="name" name="name"/> |
|  | </div> |
|  | <div class="form-group"> |
|  | <label for="comment">Team Comment/Description: </label> |
|  | <input class="form-control" type="text" id="comment" name="comment"/> |
|  | </div> |
|  | <input class="btn btn-default" type="submit" value="Create Team"/> |
|  | </form> |

**//templates.create.createPage.hbs**

|  |
| --- |
| <div class="container"> |
|  | {{> header}} |
|  | <div> |
|  | <h1>Create Page</h1> |
|  | {{> createForm}} |
|  | </div> |
|  | </div> |
|  | {{> footer}} |

## **//templates.edit.**editForm.hbs****

|  |
| --- |
| <form action="#/edit/:{{teamId}}" method="post"> |
|  | <div class="form-group"> |
|  | <label for="name">New Team Name:</label> |
|  | <input class="form-control" type="text" id="name" name="name" value="{{name}}"/> |
|  | </div> |
|  | <div class="form-group"> |
|  | <label for="comment">New Team Comment/Description: </label> |
|  | <input class="form-control" type="text" id="comment" name="comment" value="{{comment}}"/> |
|  | </div> |
|  | <input class="btn btn-default" type="submit" value="Update Team"/> |
|  | </form> |

## **//templates.edit.**editPage.hbs****

|  |
| --- |
| <div class="container"> |
|  | {{> header}} |
|  | <div> |
|  | <h1>Edit Page</h1> |
|  | {{> editForm}} |
|  | </div> |
|  | </div> |
|  | {{> footer}} |

## **//templates.home.**home.hbs****

|  |
| --- |
| <div class="container"> |
|  | {{> header}} |
|  | <div> |
|  | <h1>Home Page</h1> |
|  | {{#if loggedIn}} |
|  | {{#if hasTeam}} |
|  | <a href="#/catalog/:{{teamId}}">Go To Team</a> |
|  | {{else}} |
|  | <p>You are currently not a member of a team. View the <a href="#/catalog">catalog</a> to join or create |
|  | one.</p> |
|  | {{/if}} |
|  | {{else}} |
|  | <p>You are currently not logged in. <a href="#/login">Log in</a> / <a href="#/register">Register</a></p> |
|  | {{/if}} |
|  | </div> |
|  | </div> |
|  | {{> footer}} |

## **//templates.login.**loginForm.hbs****

|  |
| --- |
| <form action="#/login" method="post"> |
|  | <div class="form-group"> |
|  | <label for="username">Username:</label> |
|  | <input class="form-control" type="text" id="username" name="username"/> |
|  | </div> |
|  | <div class="form-group"> |
|  | <label for="password">Password:</label> |
|  | <input class="form-control" type="password" id="password" name="password"/> |
|  | </div> |
|  | <input class="btn btn-default" type="submit" value="Login"/> |
|  | </form> |

**//templates.**login.**loginPage.hbs**

|  |
| --- |
| <div class="container"> |
|  | {{> header}} |
|  | <h1>Login Page</h1> |
|  | {{> loginForm}} |
|  | </div> |
|  | {{> footer}} |

## **//templates.register.**registerForm.hbs****

|  |
| --- |
| <form action="#/register" method="post"> |
|  | <div class="form-group"> |
|  | <label for="username">Username:</label> |
|  | <input class="form-control" type="text" id="username" name="username"/> |
|  | </div> |
|  | <div class="form-group"> |
|  | <label for="password">Password:</label> |
|  | <input class="form-control" type="password" id="password" name="password"/> |
|  | </div> |
|  | <div class="form-group"> |
|  | <label for="repeatPassword">Repeat Password:</label> |
|  | <input class="form-control" type="password" id="repeatPassword" name="repeatPassword"/> |
|  | </div> |
|  | <input class="btn btn-default" type="submit" value="Register"/> |
|  | </form> |

**//templates.register.registerPage.hbs**

|  |
| --- |
| <div class="container"> |
|  | {{> header}} |
|  | <h1>Register Page</h1> |
|  | {{> registerForm}} |
|  | </div> |
|  | {{> footer}} |