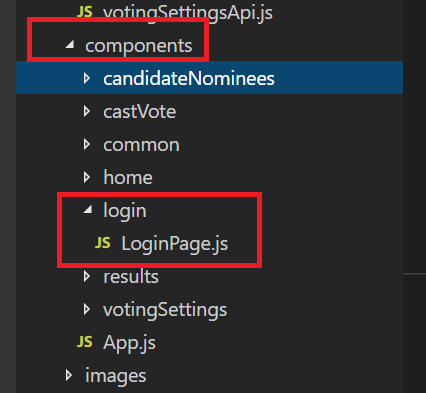
1. Create a folder login under components folder and create a js file LoginPage.js



1. Copy the below code in LoginPage.js

import React from 'react';

import PropTypes from 'prop-types';

import { connect } from 'react-redux';

import { bindActionCreators } from 'redux';

import { hashHistory } from 'react-router-dom';

import PageHeader from '../common/PageHeader';

import { BootstrapTable, TableHeaderColumn } from 'react-bootstrap-table';

import { DefaultButton, PrimaryButton, IButtonProps } from 'office-ui-fabric-react/lib/Button';

import { DatePicker, DayOfWeek, IDatePickerStrings } from 'office-ui-fabric-react/lib/DatePicker';

import Utilities from '../../utilities/utilities';

import { IfAnyGranted } from 'react-authorization';

import ErrorPage from '../common/ErrorPage';

import \* as sharedActions from '../../actions/sharedActions';

import \* as ajaxActions from '../../actions/ajaxStatusActions';

import Config from '../../api/config';

class LoginPage extends React.Component {

constructor(props, context) {

super(props, context);

this.state = {

userID: ''

};

this.login = this.login.bind(this);

this.onChangeUserID = this.onChangeUserID.bind(this);

}

login() {

let userID= this.state.userID;

let promise = this.props.actions.loadCustomUser(userID);

promise.then(users => {

document.location = "#";

})

.catch(reason => {

this.props.actions.ajaxCallError();

});

}

onChangeUserID(e) {

this.setState({ userID: e.target.value });

}

render() {

return (

<IfAnyGranted expected={[true]} actual={Config.SHOW\_LOGIN} unauthorized={<ErrorPage />}>

<div id="page-wrapper">

<div className="container-fluid">

<div className="row">

<div className="col-md-12 col-lg-12 col-sm-12">

<div className="white-box">

<form className="form-horizontal">

<div className="row">

<label className="col-md-2 col-lg-2 col-sm-2">User ID:</label>

<div className=" col-md-4 col-lg-4 col-sm-4">

<input id="userID" className="form-control" onChange={this.onChangeUserID} value={this.state.userID} type="text" />

</div>

</div>

<div className="row">

<div className=" col-md-4 col-lg-4 col-sm-4">

<PrimaryButton

text="Login" onClick={this.login}

/>

</div>

</div>

</form>

</div>

</div>

</div>

</div>

</div>

</IfAnyGranted>

);

}

}

LoginPage.propTypes = {

actions: PropTypes.object.isRequired,

loadCustomUser: PropTypes.func

};

function mapStateToProps(state, ownProps) {

return {

};

}

function mapDispatchToProps(dispatch) {

return {

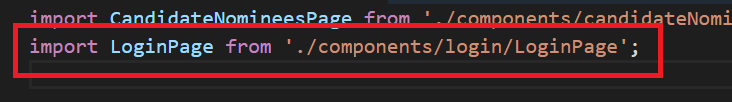
actions: bindActionCreators(Object.assign({}, sharedActions, ajaxActions), dispatch)

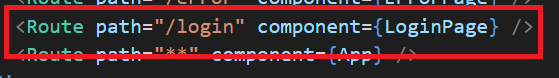
};

}

export default connect(mapStateToProps, mapDispatchToProps)(LoginPage);

1. In index.js add the following codes marked in red below





1. Add the following methods in sharedActions.js

export function loadCustomUser(userID) {

return function (dispatch) {

dispatch(beginAjaxCall());

return SharedApi.getCustomUserProperties(userID).then(userProps => {

dispatch(loadMenuOptions(userProps.user.role));

const SchoolYearTypespromise = SharedApi.getSchoolYearTypes();

SchoolYearTypespromise.then(res => {

userProps.schoolYearDescription = res.value[0].Description;// "2016-2017";// res.value[0].Description;

dispatch(loadVotingSettingsActiveYear(res.value[0].Description));

const promise = AdminApi.getHomeMessage();

promise.then(res => {

userProps.homePage = { messageHeader: res.value[0].HomeMessageHeader, messageBody: res.value[0].HomeMessageBody };

dispatch(loadUserSuccess(userProps));

});

});

if (userProps.user.role == 'Admin')

dispatch(loadAllCampuses());

else if (userProps.user.role == 'Principal')

dispatch(loadCampusesByManager(userProps.user.staffNaturalKey));

else if (userProps.user.role == 'SC')

dispatch(loadCampusesByManager(userProps.user.staffNaturalKey));

}).catch(error => {

throw (error);

});

};

}

1. Add the below methods in sharedApi.js

static getCustomUserProperties(userID) {

return new Promise((resolve, reject) => {

let userProps = {};

let user = { firstName: '', lastName: '', fullName: '', employeeID: '', role: '', roles: [], campusID: '', campusName: '' };

let currentLoginPromise = this.getCustomCurrentUserLogin(userID);

currentLoginPromise.then(

function (currentUserLogin) {

user.loginId = currentUserLogin.loginName.replace('AD\\', '');

user.fullName = currentUserLogin.name;

user.roles = getRoles(currentUserLogin.roles);

user.role = user.roles[0];

let staffInfoPromise = SharedApi.getUserByLoginID(user.loginId);

staffInfoPromise.then(

function (staffInfo) {

if (staffInfo.value && staffInfo.value.length > 0) {

user.firstName = staffInfo.value[0].FirstName;

user.lastName = staffInfo.value[0].LastSurName;

user.employeeID = staffInfo.value[0].EmployeeNumber;

user.staffNaturalKey = staffInfo.value[0].StaffNaturalKey,

user.campusID = staffInfo.value[0].EducationOrgNaturalKey;

user.campusName = staffInfo.value[0].EducationOrganization.NameOfInstitution;

}

userProps.user = user;

resolve(userProps);

}

);

});

});

}

static getCustomCurrentUserLogin(userID) {

let serviceUrl = Config.REST\_URL + 'common/api/SelectedUserInfo?username=' + userID + '';

return new Promise((resolve, reject) => {

fetch(serviceUrl, {

method: "get",

mode: 'cors',

cache: 'no-cache',

credentials: 'include',

body: undefined

})

.then(function (response) {

resolve(response.json());

})

.catch(function (error) {

reject(error);

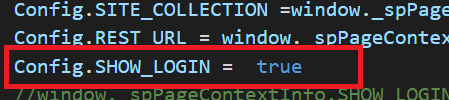
console.log("error----", error);

});

});

}

1. Add the following line marked in red in config.js under api folder



1. Update the following line marked in red in the already existing **getNavigationItems** method in sharedApi.js



1. Run the application using npm start

