import React, { Component } from 'react';

import { Navbar, NavbarBrand } from 'reactstrap';

import Directory from './DirectoryComponent';

import CampsiteInfo from './CampsiteInfoComponent';

import Header from './HeaderComponent';

import Footer from './FooterComponent';

import Home from './HomeComponent';

import About from './AboutComponent';

import Contact from './ContactComponent';

import { Switch, Route, Redirect, withRouter } from 'react-router-dom';

import { connect } from 'react-redux';

import { actions } from 'react-redux-form';

import { postComment, fetchCampsites, fetchComments, fetchPromotions } from '../redux/ActionCreators';

const mapStateToProps = (state) => {

    return {

        campsites: state.campsites,

        comments: state.comments,

        partners: state.partners,

        promotions: state.promotions

    };

};

const mapDispatchToProps = {

    postComment: (campsiteId, rating, author, text) => postComment(campsiteId, rating, author, text),

    fetchCampsites: () => fetchCampsites(),

    resetFeedbackForm: () => actions.reset('feedbackForm'),

    fetchComments: () => fetchComments(),

    fetchPromotions: () => fetchPromotions()

};

class Main extends Component {

    componentDidMount() {

        this.props.fetchCampsites();

        this.props.fetchComments();

        this.props.fetchPromotions();

    }

    render() {

        const HomePage = () => {

            return (

                <Home

                    campsite={this.props.campsites.campsites.filter((campsite) => campsite.featured)[0]}

                    campsitesLoading={this.props.campsites.isLoading}

                    campsitesErrMess={this.props.campsites.errMess}

                    partner={this.props.partners.filter((partner) => partner.featured)[0]}

                    promotion={this.props.promotions.promotions.filter((promotion) => promotion.featured)[0]}

                    promotionLoading={this.props.promotions.isLoading}

                    promotionErrMess={this.props.promotions.errMess}

                />

            );

        };

        const CampsiteWithId = ({ match }) => {

            return (

                <CampsiteInfo

                    campsite={

                        this.props.campsites.campsites.filter((campsite) => campsite.id === +match.params.campsiteId)[0]

                    }

                    isLoading={this.props.campsites.isLoading}

                    errMess={this.props.campsites.errMess}

                    comments={this.props.comments.comments.filter(

                        (comment) => comment.campsiteId === +match.params.campsiteId

                    )}

                    commentsErrMess={this.props.comments.errMess}

                    postComment={this.props.postComment}

                />

            );

        };

        return (

            <div>

                <Header />

                <Switch>

                    <Route path="/home" component={HomePage} />

                    <Route exact path="/directory" render={() => <Directory campsites={this.props.campsites} />} />

                    <Route path="/directory/:campsiteId" component={CampsiteWithId} />

                    <Route

                        exact

                        path="/contactus"

                        render={() => <Contact resetFeedbackForm={this.props.resetFeedbackForm} />}

                    />

                    <Route exact path="/aboutus" render={() => <About partners={this.props.partners} />} />

                    <Redirect to="/home" />

                </Switch>

                <Footer />

            </div>

        );

    }

}

export default withRouter(connect(mapStateToProps, mapDispatchToProps)(Main));