import \* as ActionTypes from './ActionTypes';

import { baseUrl } from '../shared/baseUrl';

export const fetchCampsites = () => (dispatch) => {

    dispatch(campsitesLoading());

    return fetch(baseUrl + 'campsites')

        .then(

            (response) => {

                if (response.ok) {

                    return response;

                } else {

                    const error = new Error(`Error ${response.status}: ${response.statusText}`);

                    error.response = response;

                    throw error;

                }

            },

            (error) => {

                const errMess = new Error(error.message);

                throw errMess;

            }

        )

        .then((response) => response.json())

        .then((campsites) => dispatch(addCampsites(campsites)))

        .catch((error) => dispatch(campsitesFailed(error.message)));

};

export const campsitesLoading = () => ({

    type: ActionTypes.CAMPSITES\_LOADING

});

export const fetchComments = () => (dispatch) => {

    return fetch(baseUrl + 'comments')

        .then(

            (response) => {

                if (response.ok) {

                    return response;

                } else {

                    const error = new Error(`Error ${response.status}: ${response.statusText}`);

                    error.response = response;

                    throw error;

                }

            },

            (error) => {

                var errMess = new Error(error.message);

                throw errMess;

            }

        )

        .then((response) => response.json())

        .then((comments) => dispatch(addComments(comments)))

        .catch((error) => dispatch(commentsFailed(error.message)));

};

export const commentsFailed = (errMess) => ({

    type: ActionTypes.COMMENTS\_FAILED,

    payload: errMess

});

export const addComments = (comments) => ({

    type: ActionTypes.ADD\_COMMENTS,

    payload: comments

});

export const addComment = (comment) => ({

    type: ActionTypes.ADD\_COMMENT,

    payload: comment

});

export const postComment = (campsiteId, rating, author, text) => (dispatch) => {

    const newComment = {

        campsiteId: campsiteId,

        rating: rating,

        author: author,

        text: text

    };

    newComment.date = new Date().toISOString();

    return fetch(baseUrl + 'comments', {

        method: 'POST',

        body: JSON.stringify(newComment),

        headers: {

            'Content-Type': 'application/json'

        }

    })

        .then(

            (response) => {

                if (response.ok) {

                    return response;

                } else {

                    const error = new Error(`Error ${response.status}: ${response.statusText}`);

                    error.response = response;

                    throw error;

                }

            },

            (error) => {

                throw error;

            }

        )

        .then((response) => response.json())

        .then((response) => dispatch(addComment(response)))

        .catch((error) => {

            console.log('post comment', error.message);

            alert('Your comment could not be posted\nError: ' + error.message);

        });

};

export const fetchPromotions = () => (dispatch) => {

    dispatch(promotionsLoading());

    return fetch(baseUrl + 'promotions')

        .then(

            (response) => {

                if (response.ok) {

                    return response;

                } else {

                    const error = new Error(`Error ${response.status}: ${response.statusText}`);

                    error.response = response;

                    throw error;

                }

            },

            (error) => {

                const errMess = new Error(error.message);

                throw errMess;

            }

        )

        .then((response) => response.json())

        .then((promotions) => dispatch(addPromotions(promotions)))

        .catch((error) => dispatch(promotionsFailed(error.message)));

};

export const promotionsLoading = () => ({

    type: ActionTypes.PROMOTIONS\_LOADING

});

export const promotionsFailed = (errMess) => ({

    type: ActionTypes.PROMOTIONS\_FAILED,

    payload: errMess

});

export const addPromotions = (promotions) => ({

    type: ActionTypes.ADD\_PROMOTIONS,

    payload: promotions});