//æøå

import \* as React from 'react';

import { RouteComponentProps } from 'react-router';

import { Cookie } from '../utilities/cookie'

import { User } from "../utilities/user";

import { Http } from "../utilities/http";

//the React component is for the user to browser the communities he is a part of

export class Communities extends React.Component<RouteComponentProps<{}>, {}> {

communities: GetMyCommunityResultCommunity[] | null = null;

constructor() {

super();

if (User.current == null) {

window.location.href = ("/Home");

}

Http.post<GetMyCommunityResult>('api/CommunityGateway/GetMy', {

userKey: User.current!.key

}).then(data => {

if (data.authorized && data.success) {

this.communities = data.communities;

this.setState({});

}

else {

window.location.href = "/Home";

}

}).catch(x => console.log(x));

}

public render() {

if (this.communities == null) {

return <div>loading</div>

}

else {

return <div>

<ul>

<li onClick={(e) => { this.props.history.push("newCommunity") }}>

<a style={{ cursor: "pointer" }}>Create new Community</a>

</li>

{(() => {

let re: JSX.Element[] = [];

for (let i: number = 0; i < this.communities.length; i++) {

re[i] = <li style={{ cursor: "pointer" }} onClick={() => {

this.props.history.push("community/" + this.communities![i].uniqueName)

}} key={i} title={this.communities[i].uniqueName}>

{this.communities[0].name}

</li>

}

return re;

})()}

</ul>

</div>

}

}

}

interface GetMyCommunityResultCommunity {

name: string;

uniqueName: string;

users: string[];

owner: string;

admins: string[];

description: string;

}

interface GetMyCommunityResult {

type: string;

authorized: boolean;

message: string;

success: boolean;

communities: GetMyCommunityResultCommunity[];

}