package com.sharehub.sharehub.sharehub

import android.content.Context

import android.content.ContextWrapper

import android.content.Intent

import android.os.Bundle

import android.widget.Toast

import com.loopj.android.http.AsyncHttpClient

import com.loopj.android.http.JsonHttpResponseHandler

import com.loopj.android.http.RequestParams

import cz.msebera.android.httpclient.Header

import org.json.JSONArray

import org.json.JSONObject

import java.io.Serializable

@Suppress("UNUSED\_EXPRESSION")

class ProfileLauncher : ShareHubActivityBase() {

class ProfileData : Serializable {

var username: String? = null

var communitiesName: Array<String>? = null

var communitiesUniqueName: Array<String>? = null

var myself: Boolean? = null

var boxes: Array<String>? = null

var boxesKey: Array<String>? = null

var nfcId: String? = null

}

companion object {

val profileDataMap: HashMap<String, ProfileData> = HashMap()

fun loadProfile(user: String, context: ShareHubActivityBase, callback: (profileData: ProfileData) -> Unit) {

if (profileDataMap.containsKey(user)) {

callback(profileDataMap[user]!!)

return

}

val client = AsyncHttpClient()

val params = RequestParams()

params.add("key", Statics.UserKey)

params.add("username", user)

client.post(Constants.UserGateway.ProfileData, params, object : JsonHttpResponseHandler() {

override fun onSuccess(statusCode: Int, headers: Array<out Header>?, response: JSONObject) {

try {

if (response["success"] as Boolean) {

val profileData = ProfileData()

profileData.boxes = response.castArray("boxes")

profileData.boxesKey = response.castArray("boxesKey")

profileData.communitiesName = response.castArray("communitiesName")

profileData.communitiesUniqueName = response.castArray("communitiesUniqueName")

profileData.username = response.cast("username")

profileData.myself = response.cast("myself")

profileData.nfcId = response.cast("nfcId")

profileDataMap[user] = profileData

callback(profileData)

}

else {

Toast.makeText(context, "you have been logged out", Toast.LENGTH\_LONG).show()

Statics.logout(context)

context.replaceActivity(Intent(context, Login::class.java))

}

}

catch (e: Exception) {

Toast.makeText(context, "an exception occurred: " + e.message, Toast.LENGTH\_LONG).show()

}

}

override fun onFailure(statusCode: Int, headers: Array<out Header>?, responseString: String?, throwable: Throwable?) {

Toast.makeText(context, "1\_http request error: " + throwable?.message, Toast.LENGTH\_LONG).show()

}

})

}

}

override fun onCreate(savedInstanceState: Bundle?) {

setContentView(R.layout.profile)

super.onCreate(savedInstanceState)

if (Statics.UserKey == null) {

replaceActivity(Intent(this, Login::class.java))

return

}

var user = ""

run {

var \_user: String? = intent.getStringExtra("user")

if (\_user == null) {

\_user = Statics.Username

}

user = \_user!!

}

loadProfile(user, this, {profileData -> goToProfile(profileData)})

}

fun goToProfile(profileData: ProfileData) {

val intent = Intent(this@ProfileLauncher, Profile::class.java)

val bundle = Bundle()

bundle.putSerializable("profileData", profileData)

intent.putExtras(bundle)

replaceActivity(intent)

}

}