package com.sharehub.sharehub.sharehub

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 com.sharehub.sharehub.sharehub.SaveTemplates.LoginSave

import com.sharehub.sharehub.sharehub.SaveTemplates.SavedData

import com.sharehub.sharehub.sharehub.SaveTemplates.SettingsSave

import org.json.JSONException

import org.json.JSONObject

import cz.msebera.android.httpclient.Header

class Main : ShareHubActivityBase() {

override fun onCreate(savedInstanceState: Bundle?) {

setContentView(R.layout.main)

super.onCreate(savedInstanceState)

try {

Statics.Settings = SavedData.Read<SettingsSave>(this, "SavedData")

if (Statics.Settings == null) {

val settingsSave = SettingsSave()

settingsSave.storeUserLoginData = true

Statics.Settings = settingsSave

SavedData.Write<SettingsSave>(this, "SettingsSave", Statics.Settings!!)

}

val loginSave = SavedData.Read<LoginSave>(this, "LoginSave")

if (loginSave?.key == null) {

goToLogin()

return

}

val client = AsyncHttpClient()

val params = RequestParams()

params.add("key", loginSave.key)

Statics.UserKey = loginSave.key;

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

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

try {

if (response!!.getBoolean("success")) {

Statics.Username = response.getString("username")

goToProfile()

} else {

if (loginSave.username != null && loginSave.password != null) {

val loginIntent = Intent(this@Main, Login::class.java)

loginIntent.putExtra("type", "autologin")

loginIntent.putExtra("username", loginSave.username)

loginIntent.putExtra("password", loginSave.password)

this@Main.replaceActivity(loginIntent)

} else {

goToLogin()

}

}

} catch (e: JSONException) {

Toast.makeText(this@Main, "unknown json error", Toast.LENGTH\_SHORT).show()

}

}

override fun onFailure(statusCode: Int, headers: Array<Header>?, res: String?, t: Throwable) {

val error: String

when (statusCode) {

404 -> error = "couldn't connect to server"

else -> error = "unknown error"

}

Toast.makeText(this@Main, statusCode.toString() + ", " + error, Toast.LENGTH\_SHORT).show()

}

})

} catch (e: Exception) {

Toast.makeText(this, e.message, Toast.LENGTH\_LONG).show()

}

}

fun goToLogin() {

val intent = Intent(this, Login::class.java)

replaceActivity(intent)

}

fun goToProfile() {

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

replaceActivity(intent)

}

}