18.12.17

DIGITALGATE ANDROID

1. OVERVIEW

Digitalgate provides integration to Turkcell login systems. We have developed an SDK that is highly robust, secure, lightweight, configurable and very simple to embed.

You can initialize login , register, switch account flows.

The Digitalgate Android SDK is compatible with Android 4.3(API 18) and above.

2. QUICK START

2.1 Adding the SDK to your Project

The simplest way to integrate the SDK into your project is by using Gradle's Dependency Management.

Adding Digitalgate's Android SDK Dependency:

1 Open your project and then open your_app | build.gradle
2 Add this to Module-level /app/build.gradle before dependencies:

```
repositories {
    maven {
        url
'https://mymavenrepo.com/repo/ukAiuNSVkftQiB4kKUPH/'
        name 'Digital Gate Repo'
    }

}
```

Add the compile dependency with the latest version of the Digitalgate SDK in the build.gradle file:

```
compile ('com.turkcell.digitalgate:digitalgate-aar:2.0.0'){
    transitive = true
}
```

2.2 Setting the Required Permissions

The AndroidManifest.xml should include the following permissions:

```
<uses-permission
android:name="android.permission.READ_PHONE_STATE" />
<uses-permission
android:name="android.permission.GET_ACCOUNTS" />
```

3. SDK INITIALIZATION

Initialization of the SDK is completed in two stages. In the first stage the DGLoginCoordinator is created by using Builder Pattern. When building DGLoginCoordinator, appId is required where theme, language and environment parameters are optional. In the second stage the call to one of the main flows(e.g. call startForLogin, startForSwitchAccount, startForRegister) is needed.

Optional Parameters;
-theme: default theme of the application exists. See Section 10.
-language: default language is TR
-environment: default environment is PROD

To initialize the SDK, add the following code in your Application activity or fragment:

```
DGLoginCoordinator dg = new
DGLoginCoordinator.Builder().appId(app_id).build();
```

4. SDK START LOGIN

Having initialized the SDK, start login one of the main flows to call. It takes three boolean parameters;

-disableCellLogin: if true, cellular login functionality won't work.

-autoLoginOnly: if true, only cellular login and remember me will work

-disableAutoLogin: if true, login process is forced to user.

```
try {
        dg.startForLogin(this, disableCellLogin,
autoLoginOnly, disableAutoLogin);
} catch (DGException e) {
        //application error handling, e.g. required appId
}
```

For getting result, see Section 10.

5. SDK START REGISTER

Having initialized the SDK, start register one of the main flows to call. It runs without parameters;

```
try {
    dg.startForRegister(this);
} catch (DGException e) {
    //application error handling, e.g. required appId
}
```

For getting result, see Section 10.

6. SDK SWITCH ACCOUNT

Having initialized the SDK, switch account one of the main flows to call. It runs without parameters;

```
try {
    dg.startForSwitchAccount(this);
} catch (DGException e) {
    //application error handling, e.g. required appId
}
```

For getting result, see Section 10.

7. SDK START LOGIN WITH TRANSFER TOKEN

Having initialized the SDK, start login with transfer token one of the main flows to call. It takes three boolean parameters and one string parameter;

-disableCellLogin: if true, cellular login functionality won't work.

-autoLoginOnly: if true, only cellular login and remember me will work

-disableAutoLogin: if true, login process is forced to user.

-transferToken: transfer token that taken from the backend must be passed.

```
 try {
        dg.startForLoginWithTransferToken(this,
disableCellLogin, autoLoginOnly, disableAutoLogin,
transferToken);
} catch (DGException e) {
        //application error handling, e.g. required appId
}
```

For getting result, see Section 10.

8. SDK WIDGET LOGIN

Having initialized the SDK, widget login one of the main flows to call. It doesn't take any parameters. However, it only works if there is an active rememberme login in the relative application.

```
try {
    dg.startForWidgetLogin(getApplicationContext());
} catch (DGException e) {
    //application error handling, e.g. required appId
}
```

For getting result, see Section 9.

9. SDK WIDGET RESULT

To get the result from SDK, a class needed that is extended from BroadcastReceiver and relative action must be registered. See below example code:

```
IntentFilter filter = new IntentFilter();
filter.addAction(DGLoginCoordinator.DG_WIDGET_BROADCAST_RESULT);

WidgetReceiver myReceiver = new WidgetReceiver(this);
registerReceiver(myReceiver, filter);
```

After registration, DGResult can be taken like below example code;

```
if
(DGLoginCoordinator.DG_WIDGET_BROADCAST_RESULT.equals(intent.getAction())) {
    DGResult dgResult = DGLoginCoordinator.getDGResult(intent);

    //dgResult has the result, take action according to result
}
```

Result Type	Description	Action
DGResultType.SUCCESS_LOGIN	Successful Login	You can take loginToken and continue
DGResultType.SUCCESS_NO_LOGIN	(ONLY FOR WIDGET) Login needed in the application	For successful login to widget, there must be a logged in user. You can force user for application login
DGResultType.ERROR_APPLICATON	There is a system error in digitalgate	You can continue Non-Login or force user to login again
DGResultType.ERROR_SESSION_LOST	Session lost during digitalgate processes.	You can continue Non-Login or force user to login by starting digitalgate again

10. SDK RESULT

To get the result from SDK, onActivityResult method must be overridden in the application's Activity or Fragment.

To get the result, add the following code in your Application's fragment or activity:

```java
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    if (requestCode == DGLoginCoordinator.DG_REQUEST_CODE) {
        if (resultCode == Activity.RESULT_CANCELED) {
            //Unexpected Error Case
        }
        if (resultCode == Activity.RESULT_OK) {
            DGResult dgResult =
DGLoginCoordinator.getDGResult(data);
            //dgResult has the result, take action according
to result
        }
    }
}
```

These are digitalgate result types;

Result Type	Description	Action
DGResultType.SUCCESS_LOGIN	Successful Login	You can take loginToken and continue
DGResultType.ERROR_USER_QUIT	User intentionaly quits from Digitalgate	You can continue Non-Login or force user to login again by starting digitalgate again
DGResultType.ERROR_APPLICATON	There is a system error in digitalgate	You can continue Non-Login or force user to login again
DGResultType.ERROR_SESSION_LOST	Session lost during digitalgate processes.	You can continue Non-Login or force user to login by starting digitalgate again

11. SDK LOGOUT

To logout from the system, there is a static method to call. Re-initialization of DGLoginCoordinator is not needed. Add the following code in your Application:

```
DGLoginCoordinator.logout(this, appId);
```

For logout, there is no result. Having called the logout method is enough.

12. SDK STYLE CONFIGURATION

The configuration of the sdk can be achieved by creating DGTheme and passing DGTheme to DGLoginCoordinator. Builder pattern is used for creating the DGTheme.

Sample code for configuring the style:

```
DGTheme dgTheme = new DGTheme.Builder()
        .setBackgroundColor(android.R.color.holo_green_light)
        .setTitleLabelColor(android.R.color.holo_red_dark)
        .setDescriptionTextColor(android.R.color.holo_orange_dark)
        .setCheckBoxPassiveIcon(R.drawable.dg_checkbox_normal)
        .setPositiveButtonTextColor(android.R.color.black)
        .build();

DGLoginCoordinator dg = new
DGLoginCoordinator.Builder().theme(dgTheme).appId(app_id).build();
```

@DrawableRes int backButtonIcon	
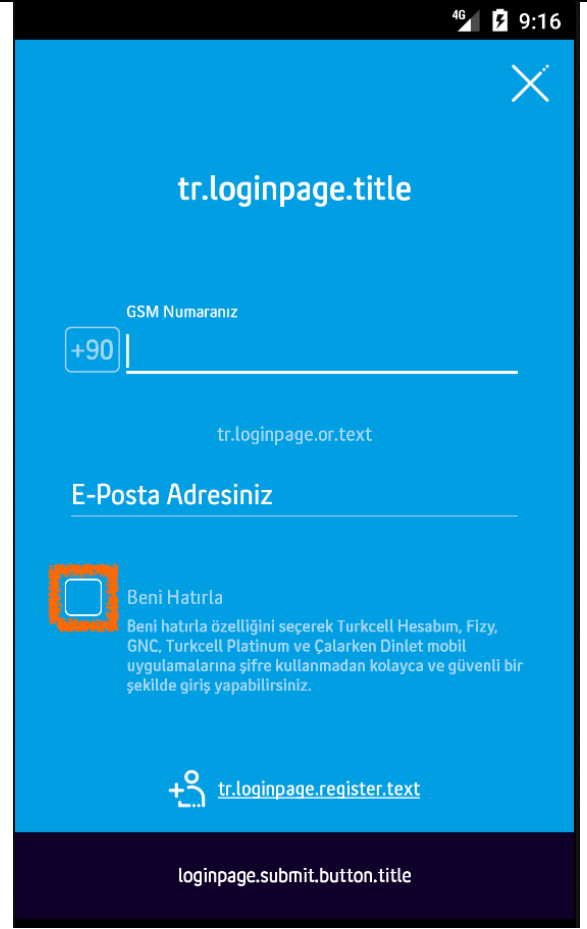
@DrawableRes int errorPopupImage	 **Hata** tr.fields.are.empty **Tamam**
@DrawableRes int infoPopupImage	 **Bilgi** Success **Tamam**
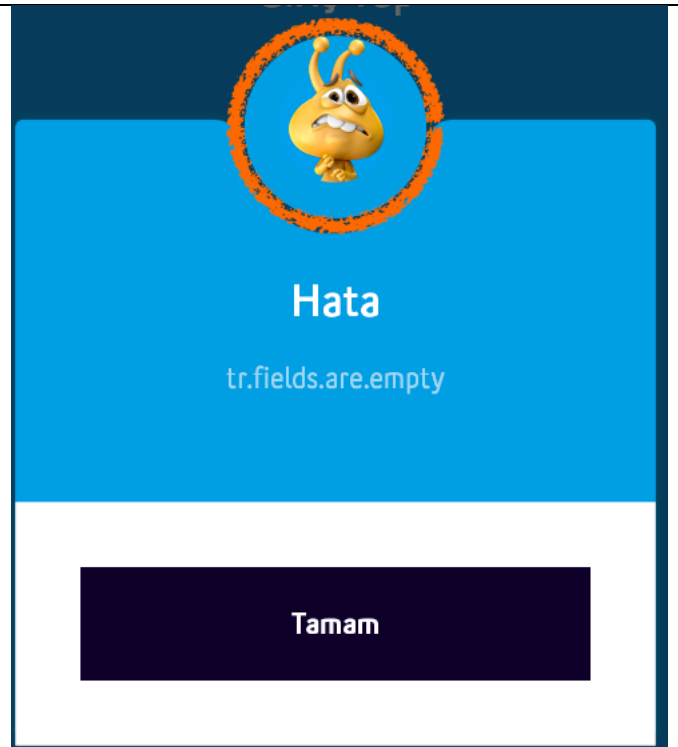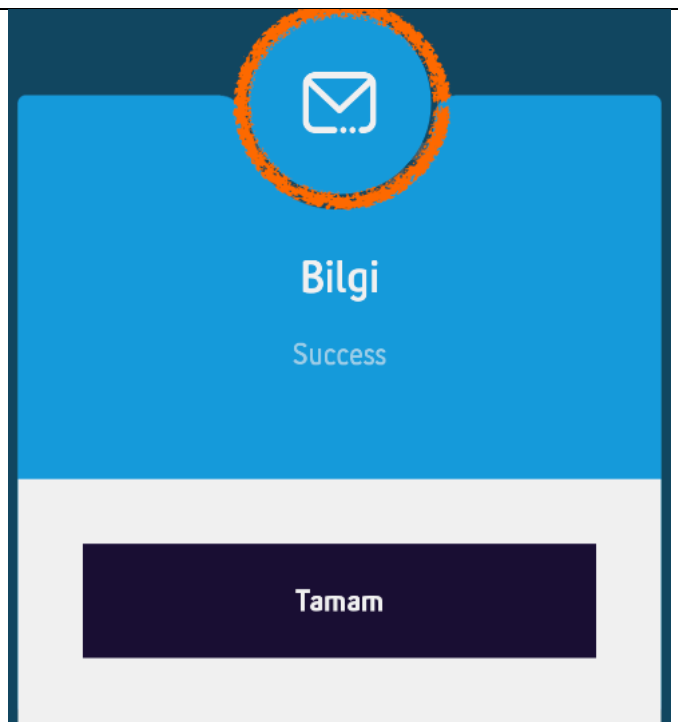
@DrawableRes int checkBoxActiveIcon	
@DrawableRes int checkBoxPassiveIcon	

@DrawableRes int crossButtonIcon	
@DrawableRes int registerIcon	

@DrawableRes int regionSelectIcon	
@ColorRes int backgroundColor	

@ColorRes int titleLabelColor	
@ColorRes int descriptionTextColor	

@ColorRes int textFieldSeperatorColor	
@ColorRes int positiveButtonBackgroundColor	

@ColorRes int positiveButtonTextColor	
@ColorRes int negativeButtonBackgroundColor	

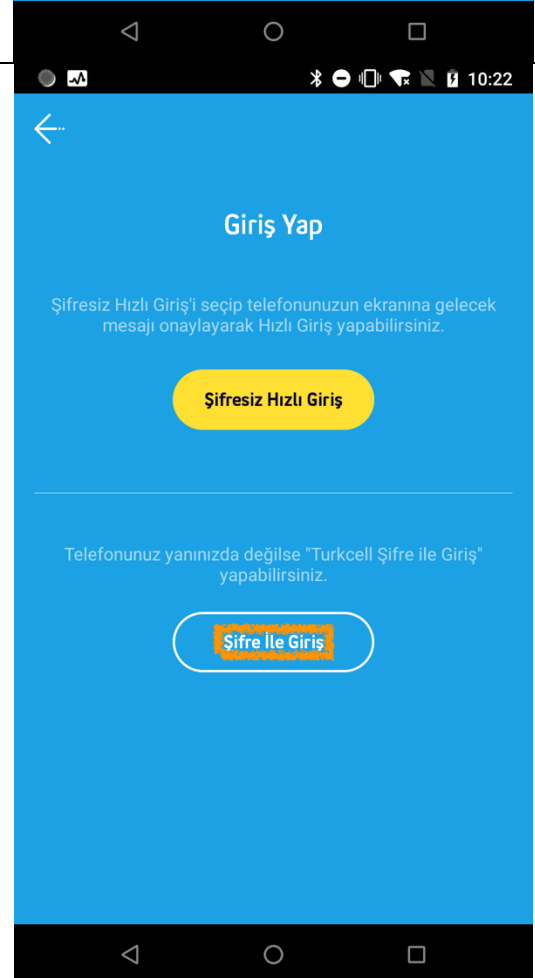|	
@ColorRes int negativeButtonTextColor	
@ColorRes int roundedFillButtonTextColor	
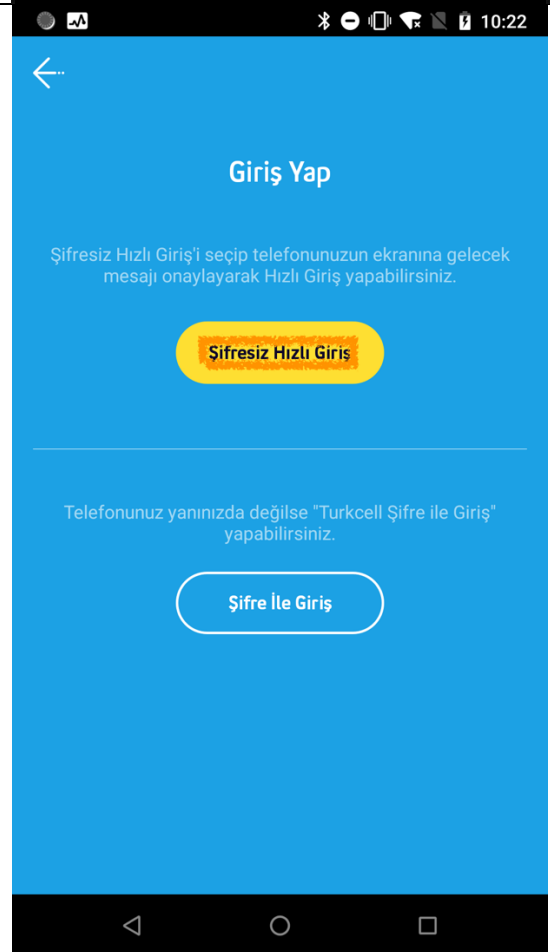
@ColorRes int roundedFillButtonBackgroundColor	
@ColorRes int roundedTransparentButtonTextColor	
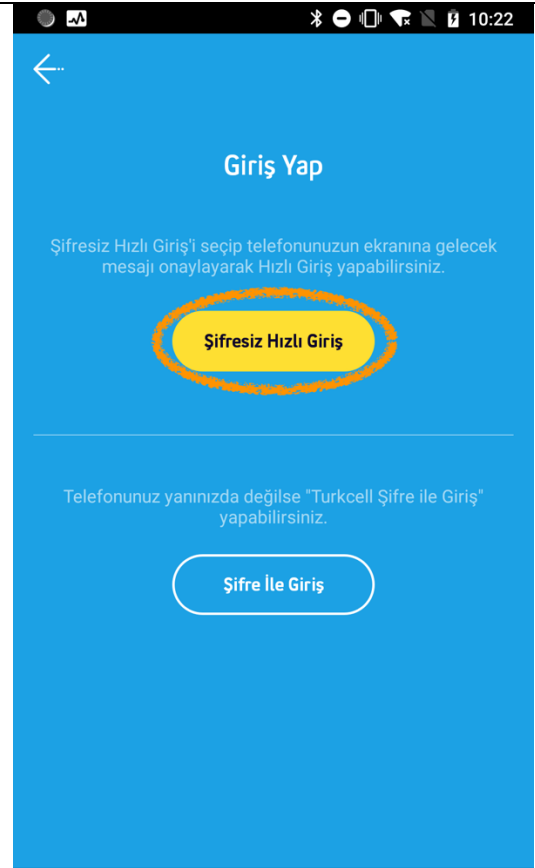
@ColorRes int roundedTransparentButtonBorderColor	
@ColorRes int popUpTopColor	
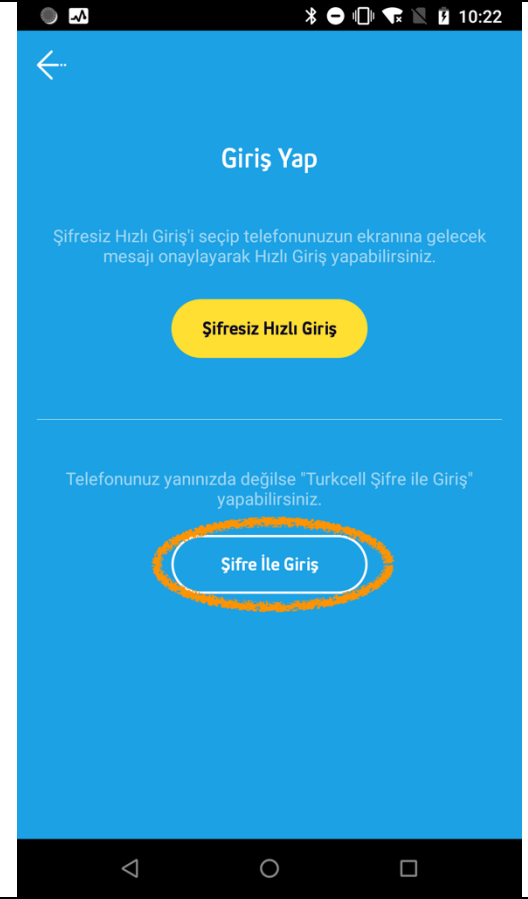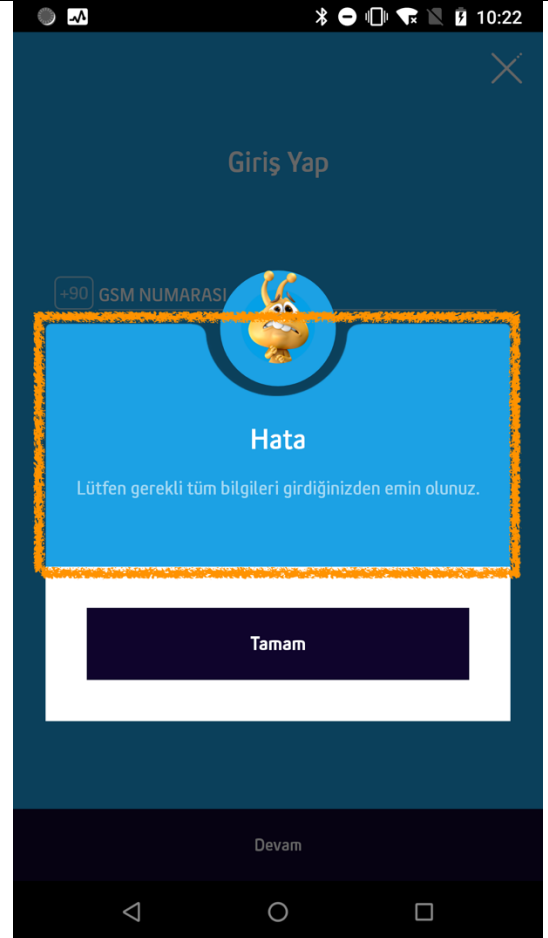
@ColorRes int popUpBottomColor	
@ColorRes int popupTitleLabelColor	
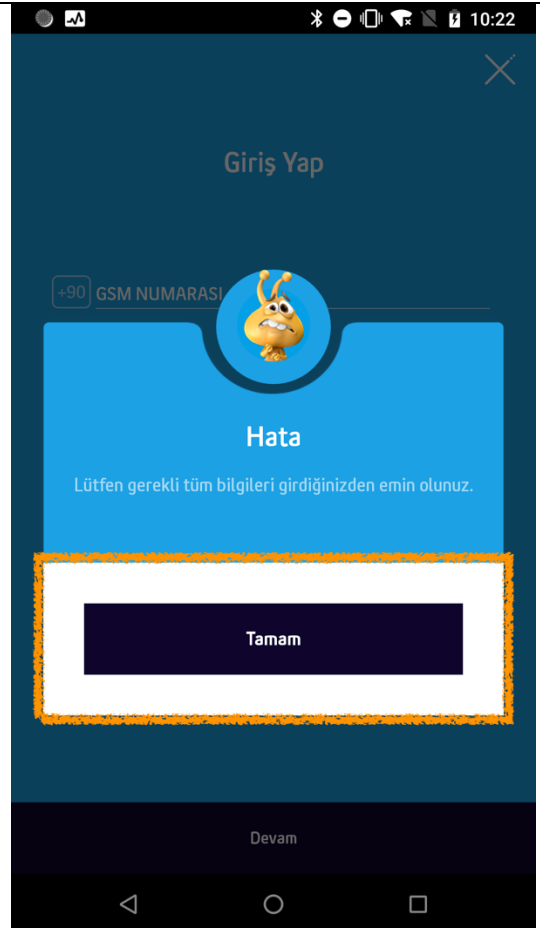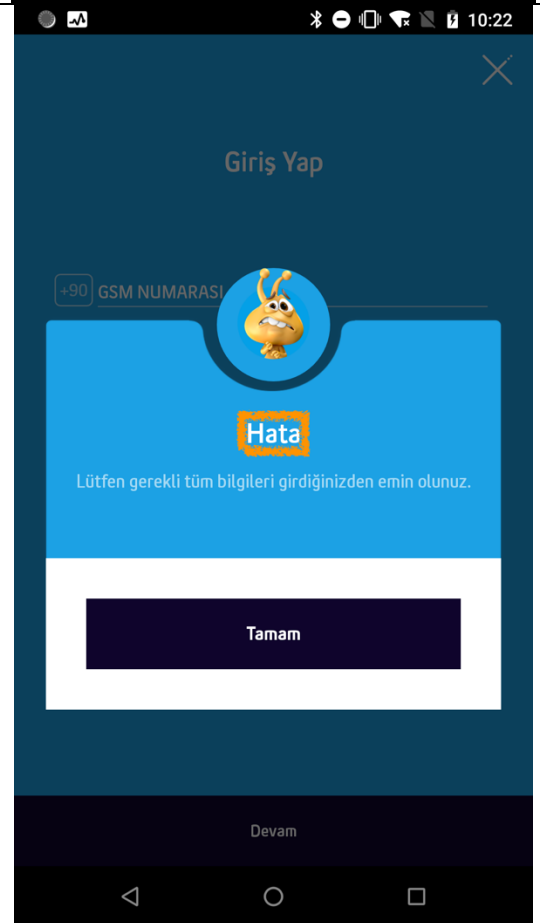
@ColorRes int popupDescriptionTextColor	
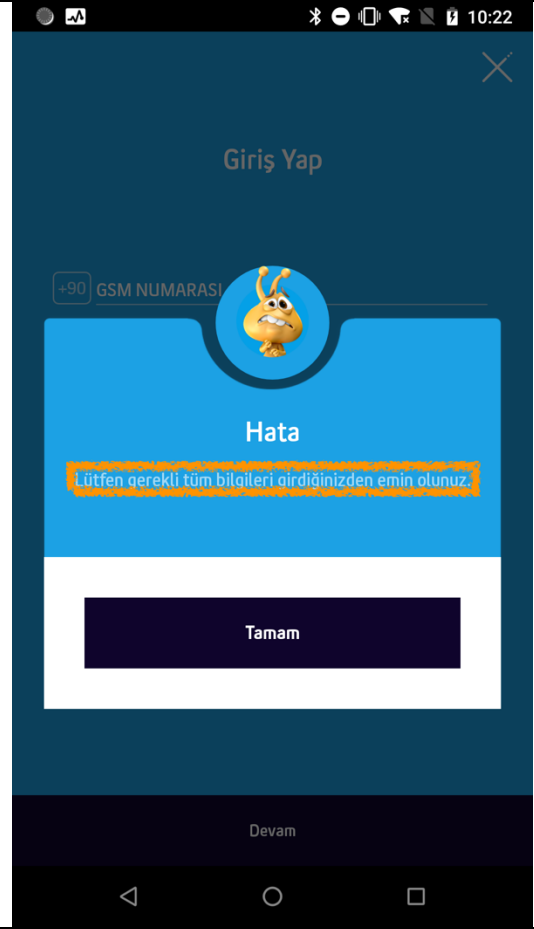
13. FAQ

Q: What is my app ID?

A: You can contact with Osman Kara(osman.kara@turkcell.com.tr) for your all questions.

Q: Do we need to set theme in the DGLoginCoordinator?

A: No you don't. There is a default theme in DigitalGate, if you don't need a special theme then you can continue with the default one by not setting or just setting null for theme method in DGLoginCoordinator.

Q: Is there a test Environment for Digitalgate?

A: Yes there are two environments for Digitalgate. One for prod, other for test. Default environment is prod. For test initialization use following code:

```
DGLoginCoordinator dg = new
DGLoginCoordinator.Builder().appId(app_id).
environment(DGEnv.TEST).build();
```

Q: Repo url starts with "http://mymavenrepo...", is this real repo?

A: Yes it is a real repository. Our need was a free public repo. This is the reason we are using it.

Q: I have other questions..

A: You can contact with Osman Kara(osman.kara@turkcell.com.tr) for your all questions.