POKKT SDK v5.0 Integration Guide for Marmalade (Android)

Contents:

1. Overview

Thank you for choosing **Pokkt SDK Plugin v5.0** for **Marmalade**. Pokkt SDK supports Reward Ad, Non-Reward Ad, Banner Ad and Offerwal campaigns feature. This document contains all the information which is needed to setup the SDK with project. Please follow these steps as per your integration requirement (Reward/Non-Reward/Banner/Offerwal). The current plugin supports mediation for various third party ad-networks. These are:

> 1. AdColony
> 2. AppLovin
> 3. Chartboost
> 4. Fyber
> 5. InMobi
> 6. Supersonic
> 7. UnityAds
> 8. Tapjoy
> 9. Vungle
> 10. AdMob
> 11. Facebook

A separate set of documents is provided for each of these, explaining the implementation process.

There is a sample app provided with the SDK. We will be referencing this app during the course of explanation in this document. It is suggested that you should check that app to understand the following process in detail.

2. Configuration steps

All we need is the file provided **PokktNativeExtension.zip** file which contains two files **PokktNativeExtension.zip** and **PokktSample.zip**. **PokktNativeExtension** is the extension project for Pokkt SDK which will be added in main project as sub project in .mkb file. Please check our given sample app **PokktSample.zip**.

Android Manifest xml update:

1. Add following permissions in your manifest xml if not added.

```
<!-- These permissions are mandatory to run Pokkt SDK -->
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<!-- These permissions are strongly recommended and will result in higher performance -->
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<!-- This permission is optional  but will improve SDK feature-->
<uses-permission android:name="android.permission.WRITE_CALENDAR" />
```

2. Following activity need to add in manifest xml but it is already added in our given extension project so you no need to do anything.

```
<activity android:name=“com.app.pokktsdk.ShowOfferwallActivity"
    android:configChanges=“keyboard|keyboardHidden|navigation|orientation|screenLayout|uiMode|screenSize"
    android:label=“@string/app_name"
    android:windowSoftInputMode="adjustPan" >
</activity>

<activity
    android:name=“com.app.pokktsdk.PlayVideoCampaignActivity”
    android:configChanges=“keyboard|keyboardHidden|navigation|orientation|screenLayout|uiMode|screenSize”
    android:label=“@string/app_name”
    android:screenOrientation=“landscape”
    android:windowSoftInputMode="stateAlwaysHidden|adjustUnspecified" >
</activity>

<activity
    android:name=“com.app.pokktsdk.VPAIDActivity”
    android:configChanges=“keyboard|keyboardHidden|navigation|orientation|screenLayout|uiMode|screenSize|smallestScreenSize”
    android:label=“@string/app_name”
    android:windowSoftInputMode="stateAlwaysHidden|adjustUnspecified"/>

<activity
    android:name=“com.app.pokktsdk. PokktInterstitialActivity”
    android:configChanges=“keyboard|keyboardHidden|navigation|orientation|screenLayout|uiMode|screenSize|smallestScreenSize”
    android:label=“@string/app_name”
    android:windowSoftInputMode="stateAlwaysHidden|adjustUnspecified"/>

<receiver
    android:name="com.app.pokktsdk.AppInstallBroadcastReceiver" >
    <intent-filter android:priority="1000" >
        <action android:name="android.intent.action.PACKAGE_INSTALL" />
        <action android:name="android.intent.action.PACKAGE_ADDED" />
        <data android:scheme="package" />
    </intent-filter>
</receiver>
```

Notes: You need to add **google-play-services-android** as subproject in your project’s .mkb file.

```
<meta-data
    android:name="com.google.android.gms.version"
    android:value="@integer/google_play_services_version" />
```

3. Add a meta data tag for offerwall campaign. you will have to implement the IOfferwallDelegate interface in your project to listen for all offerwall related events. (refer to Pokkt sample for example).

```
<meta-data
    android:name="offerwallDelegate
    android:value="<com.pokkt.plugin.common.PokktOfferwallDelegate>" />
```

4. Add following Service and receiver in manifest for google analytics (Optional).

```
<receiver
    android:name=“com.google.android.gms.analytics.AnalyticsReceiver"
    android:enabled=“true">
    <intent-filter>
        <action android:name="com.google.android.gms.analytics.ANALYTICS_DISPATCH"
 />
    </intent-filter>
</receiver>

<service
    android:name=“com.google.android.gms.analytics.AnalyticsService"
    android:enabled=“true"
    android:exported=“false"/>
```

5. Add following Service in manifest for in-app notification.

```
<service
    android:name=“com.app.pokktsdk.notification.NotificationService"
    android:label=“PokktNotificationService”
    android:exported="false"/>
```

Note: Please do not copy the code points from this PDF file as it may introduce unwanted characters and space in your code. Instead please refer to sample app source code provided with the sample app.

3. Implementation Steps

Common

1. For all invocation of Pokkt SDK developer will make use of methods available in **PCPokktManager** class which is static class.

2. In **PCPokktConfig**, you can set **ApplicationId**, **SecurityKey**. which are must for all type of integrations. Please check the sample app.

3. Before calling any other methods from the **PCPokktManager**, please make sure that you have called the **InitPokkt** with passing PCPokktConfig object.

4. If you are doing server to server integration with POKKT you can also set **thirdPartyUserId** in PCPokktConfig.

5. While in development, please call **setDebug(true)** to see Pokkt debug logs and toast messages. please make sure to change this to **setDebug(false)** for production build.

   ```
   PCPokktManager::setDebug(true/false);
   ```

6. Android **MinSDKVersion** should be >= **14**.

7. To use google analytics, please set **AnalyticsType** and **Analytics ID** in

   ```
   pokktConfig->selectedAnalyticsType = POKKT_ANALYTICS_TYPE_GOOGLE_ANALYTICS;
   pokktConfig->googleAnalyticsID = "Id";
   ```

8. To use **Flurry analytics** please set AnalyticsType and Flurry Application Key in PCPokktConfig.

   ```
   pokktConfig->selectedAnalyticsType = POKKT_ANALYTICS_TYPE_FLURRY;
   pokktConfig->flurryApplicationKey = "key";
   ```

9. To use **Mix Panel analytics** please set AnalyticsType and Mix PanelProject Token in PCPokktConfig:

   ```
   pokktConfig->selectedAnalyticsType = POKKT_ANALYTICS_TYPE_MIXPANEL;
   pokktConfig->mixPanelProjectToken = "token";
   ```

10. To use **Fabric analytics** please set AnalyticsType and Fabric Token in PCPokktConfig

    ```
    pokktConfig->selectedAnalyticsType = POKKT_ANALYTICS_TYPE_FABRIC;
    ```

11. Please call trackIAP(InAppPurchaseDetail) to send any in-app purchase information to Pokkt. Like below you can do this.

    ```
    // create object of IAPDetails and set properties for purchaseDetails and pass
     object
    PCPokktManager::trackIAP(IAPDetails);
    ```

Session

1. We have option to start session and end session for tracking in PCPokktManager.

   ```
   PCPokktManager::startSession();
   PCPokktManager::endSession();
   ```

2. You should call startSession at the start of his application if you want to use this but this is the optional

and call it after setting application id and security key.

3. You should call endSession at the end of his application.

Offerwall

1. In PCPokktConfig for OfferWall, you can set two additional parameters which are OfferWallAssetValue and closeOnSuccessFlag.
   OfferWallAssetValue is only required if you only want to show campaign of certain value on offerwall.
   closeOnSuccessFlag is required if you wish to close the OfferWall after user has completed one offer. It’s default value false.

2. Before calling another method for offerWall in **PCPokktManager**, please make sure that you have already called **initPokkt** first.

3. You need to add event listener as below or also please check the **OfferwallScreen** class in sample app.

   ```
   PCPokktManager::setOfferwallEventListener(COIN_RESPONSE_EVENT, PC_OFFERWALL_EVENT_SELECTOR(OfferwallScreen::handleCoinResponse), this);
   PCPokktManager::setOfferwallEventListener(COIN_RESPONSE_EVENT, PC_OFFERWALL_EVENT_SELECTOR(OfferwallScreen::handleCoinResponseWithTrId), this);
   PCPokktManager::setOfferwallEventListener(COIN_RESPONSE_EVENT, PC_OFFERWALL_EVENT_SELECTOR(OfferwallScreen::handleCoinResponseFailed), this);
   PCPokktManager::setOfferwallEventListener(COIN_RESPONSE_EVENT, PC_OFFERWALL_EVENT_SELECTOR(OfferwallScreen::handleCampaignAvailibity), this);
   PCPokktManager::setOfferwallEventListener(COIN_RESPONSE_EVENT, PC_OFFERWALL_EVENT_SELECTOR(OfferwallScreen::handleOfferwallclosed), this);
   ```

4. To show OfferWall you can call getCoins.

   ```
   PCPokktManager::getCoins(pokktConfig)
   ```

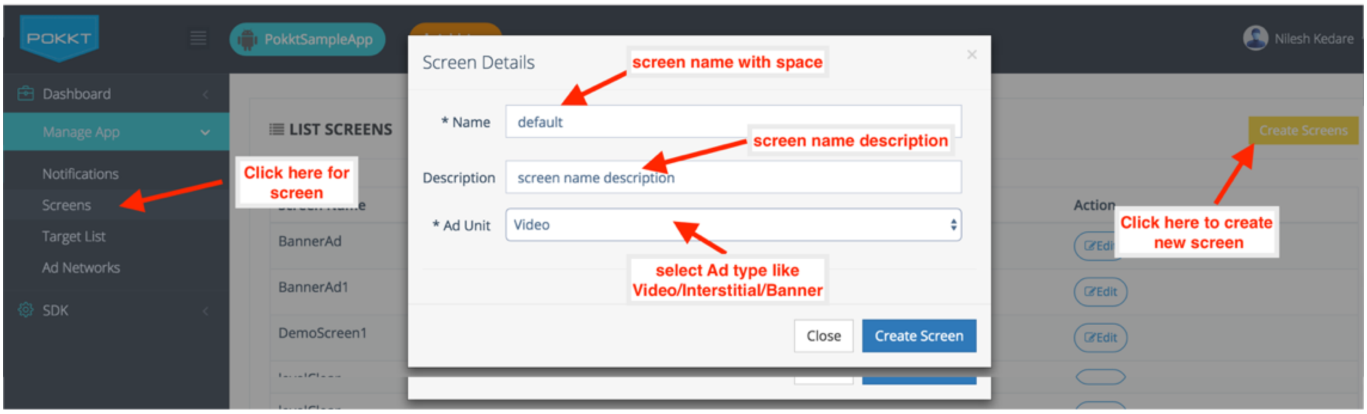
5. In the screen where you have button to show offerwall, in that activity onResume you should call **PCPokktManager::getPendingCoins(pokktConfig)** so that you get a callback to award points to the user after he has come back to your game after finishing with OfferWall. You will get a callback for this where you add the listener.

6. You can call **PCPokktManager::checkOfferWallCampaign(pokktConfig);** to check whether the campaigns are available before showing OfferWall button to user. You will get a callback for this call.

PCAdConfig (used for reward / non-reward ad)

1. In **PCAdConfig**, you need to provide **screenName**, **isRewarded** and **adFormat**. This screen name will be created on Pokkt dashboard.

Screen Name: For screen name, you will have to create on Pokkt dashboard. Please check below screen. This will help you to understand how to create new screen name in Pokkt dashboard.



* Name: default

Description: screen name description

* Ad Unit: Video

screen name with space

screen name description

select Ad type like Video/Interstitial/Banner

Click here for screen

Click here to create new screen

IsRewarded: You need to provide ad type. Pass it "false" if you are looking for non-reward ad otherwise "true" for reward ad.

AdFormat: you can set VIDEO / BANNER / INTERSTITIAL like below:

```
adConfig->adFormat = VIDEO;
adConfig->adFormat = BANNER;
adConfig->adFormat = INTERSTITIAL;
```

2. In **PCAdConfig** developer can also set shouldAllowSkip, defaultSkipTime, skipConfirmMessage, backButtonDisabled, shouldAllowMute, shouldSkipConfirm, skipConfirmYesLabel, skipConfirmNoLabel, skipTimerMessage and incentiveMessage. These values can be used to configure the behaviour of ad.

3. If you want to enable/disable the skip button on video screen please set shouldAllowSkip as true/false. The default value for shouldAllowSkip is true.

4. If you have enabled skipped button by setting shouldAllowSkip as true then you can control after how many seconds the skip button will be visible in video by setting defaultSkipTime to appropriate value. Since most videos will be 30 sec or less please set defaultSkipTime as 10 or less. You can also give your own skip message by setting skipConfirmMessage on PCAdConfig.

5. The screenName has default value "default" and can be used by you to give different screen name for different places in your app where you are showing ads. You will control ad targeting based on these screen names which should match exactly with screen names which is defined in dashboard. ScreenName can't contain white spaces and only special characters allowed are hyphen and underscore.

6. You can choose to show ad with or without incentive to user by setting isRewarded as true or false. Ad gratification will only happen for incentivised playback.

7. You can disable the back button while video is playing by setting backButtonDisabled on PCAdConfig.

8. You can configure the ad skip dialog yes/no labels by setting skipConfirmYesLabel and skipConfirmNoLabel.

9. You can configure the ad incentive message by setting incentiveMessage.

10. You can configure the ad skip timer message by setting skipTimerMessage. The message must contain a ## placeholder to show skip time value, which will keep changing as per the time.

Reward Ad / Non-Reward Ad

Rewarded ad and Non-Rewarded ad can be Video or Interstitial ad.

1. You need to set true/false for rewarded or non-rewarded ad like below in PCAdConfig object:

   ```
   adConfig->isRewarded = true/false;
   ```

2. You will have to call cache ad to start caching ads on device.

   ```
   PCPokktManager::cacheAd(adConfig);
   ```

3. You will need to register event for getting callback for Ad related like below and also please check given sample app AdScreen class and also keep in mind to remove listener at proper place. Please check given sample.

   ```
   PCPokktManager::setAdEventListener(AD_AVAILABILITY_EVENT, PC_AD_EVENT_SELECTOR(AdScreen::handleAdAvailability), this);
   PCPokktManager::setAdEventListener(AD_CACHING_COMPLETED_EVENT, PC_AD_EVENT_SELECTOR(AdScreen::handleAdCachingCompleted), this);
   PCPokktManager::setAdEventListener(AD_CACHING_FAILED_EVENT, PC_AD_EVENT_SELECTOR(AdScreen::handleAdCachingFailed), this);
   PCPokktManager::setAdEventListener(AD_DISPLAYED_EVENT, PC_AD_EVENT_SELECTOR(AdScreen::handleAdDisplayed), this);
   PCPokktManager::setAdEventListener(AD_SKIPPED_EVENT, PC_AD_EVENT_SELECTOR(AdScreen::handleAdSkipped), this);
   PCPokktManager::setAdEventListener(AD_COMPLETED_EVENT, PC_AD_EVENT_SELECTOR(AdScreen::handleAdCompleted), this);
   PCPokktManager::setAdEventListener(AD_CLOSED_EVENT, PC_AD_EVENT_SELECTOR(AdScreen::handleAdClosed), this);
   PCPokktManager::setAdEventListener(AD_GRATIFIED_EVENT, PC_AD_EVENT_SELECTOR(AdScreen::handleAdGratified), this);
   ```

4. You can call **checkAdAvailability** method to check if the campaign are available for a particular adConfig. Call this before you try to show ad.

   ```
   PCPokktManager::checkAdAvailability(adConfig);
   ```

5. You can call **showAd** method to show ad.

```
PCPokktManager::showAd(adConfig);
```

6. Please reward user only from the AD*GRATIFIED*EVENT event callback.

Banner Ad

Pokkt SDK allows to show banner ad on your screen. You can set any custom size or any position for banner. There are few fixed position already given in BannerPosition class. But you can customise that also.

1. **Load Banner:** Use loadBanner to show banner ad like below:

```
PCPokktManager::loadBanner(<screen name>, TOP_CENTER);

There is predefined positions are already given which can be used but there is
 separate method if you want to customise banner ad.

TOP_LEFT, TOP_CENTER, TOP_RIGHT, MIDDLE_LEFT, MIDDLE_CENTER, MIDDLE_RIGHT, BOT
TOM_LEFT, BOTTOM_CENTER, BOTTOM_RIGHT
```

2. **Remove Banner:** Use removeBanner method to remove banner ad from screen like below:

```
PCPokktManager::removeBanner(<screenName>);
```

3. **Auto Refresh Banner:** Use setBannerAutoRefresh method to disable or enable auto refresh. Default it is true and it will refresh automatically based on given time on Pokkt dashboard for particular screen name.

```
PCPokktManager::setBannerAutoRefresh(false/true);
```

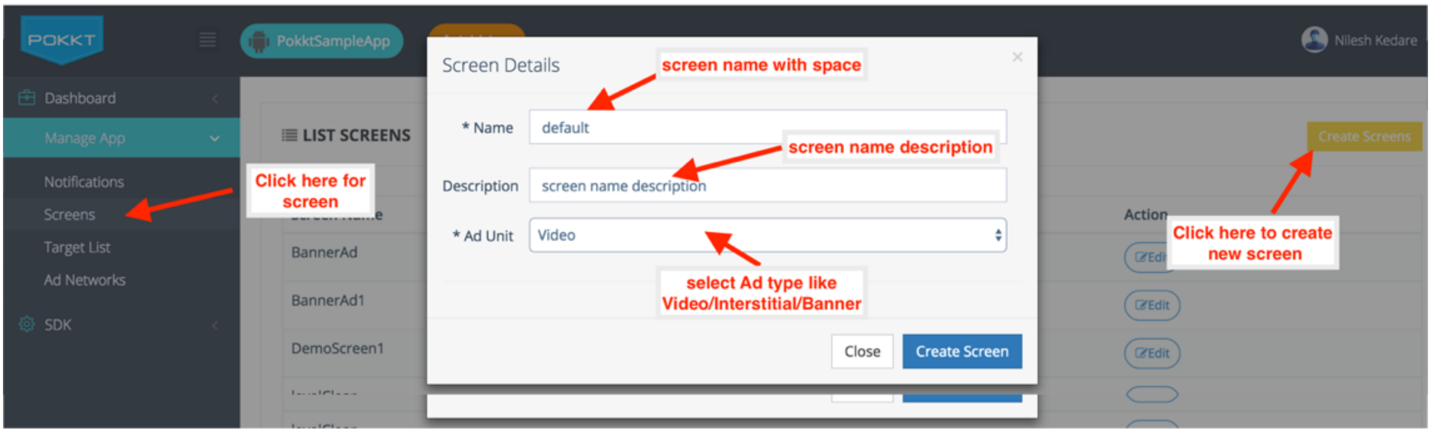
4. **Custom banner:** This method can be called to customize banner size and position by using below method:
~~~~ PCPokktManager::loadBannerWithRect(ScreenName, width, height, x, y); ~~~~

5. **Banner Event:**Register event to get banner related callback like below:
~~~~

```
// Register event
PCPokktManager::setBannerEventListener(BANNER_LOADED_EVENT, PC_BANNER_EVENT_SELECTOR(SelectorScreen::handleBannerLoaded), this);
PCPokktManager::setBannerEventListener(BANNER_LOAD_FAILED_EVENT, PC_BANNER_EVENT_SELECTOR(SelectorScreen::handleBannerLoadFailed), this);

// Handler method
void SelectorScreen::handleBannerLoaded(std::string screenName, std::string message) {
    //event.screenName: same scree name for what banner ad request was done
}
void SelectorScreen::handleBannerLoadFailed(std::string screenName, std::string errorMessage) {
    //event.screenName: same scree name for what banner ad request was done
    //event.message: Reason why banner load failed
}
//Please follow our given sample app
```

6. **Screen Name:** For screen name, you will have to create in Pokkt dashboard. Please check below screen. This will help you to understand how to create new screen name in Pokkt dashboard.



* Name: default

Description: screen name description

* Ad Unit: Video

screen name with space

screen name description

select Ad type like Video/Interstitial/Banner

Click here for screen

Click here to create new screen

Mediation Info

1. Pokkt SDK supports 11 ad networks which you can integrate in your application for better monetisation.

2. To integrate these networks through Pokkt, please visit the mediation menu on downloads page and download Marmalade mediation zip and documentation zip files.

3. Please follow the mediation integration documents shipped for each network.

4. You will need to create account on these networks and add the network details in your Pokkt dashboard after login into your account on Pokkt website.

5. You will also need to map Pokkt screens with the corresponding ad network's placement

ID/zoneID/AdUnitID etc in the dashboard.

Export Logs

1. Developer should call exportLog method to export the Pokkt SDK logs to folder of your choice.

   ```
   PCPokktManager::exportLog();
   ```

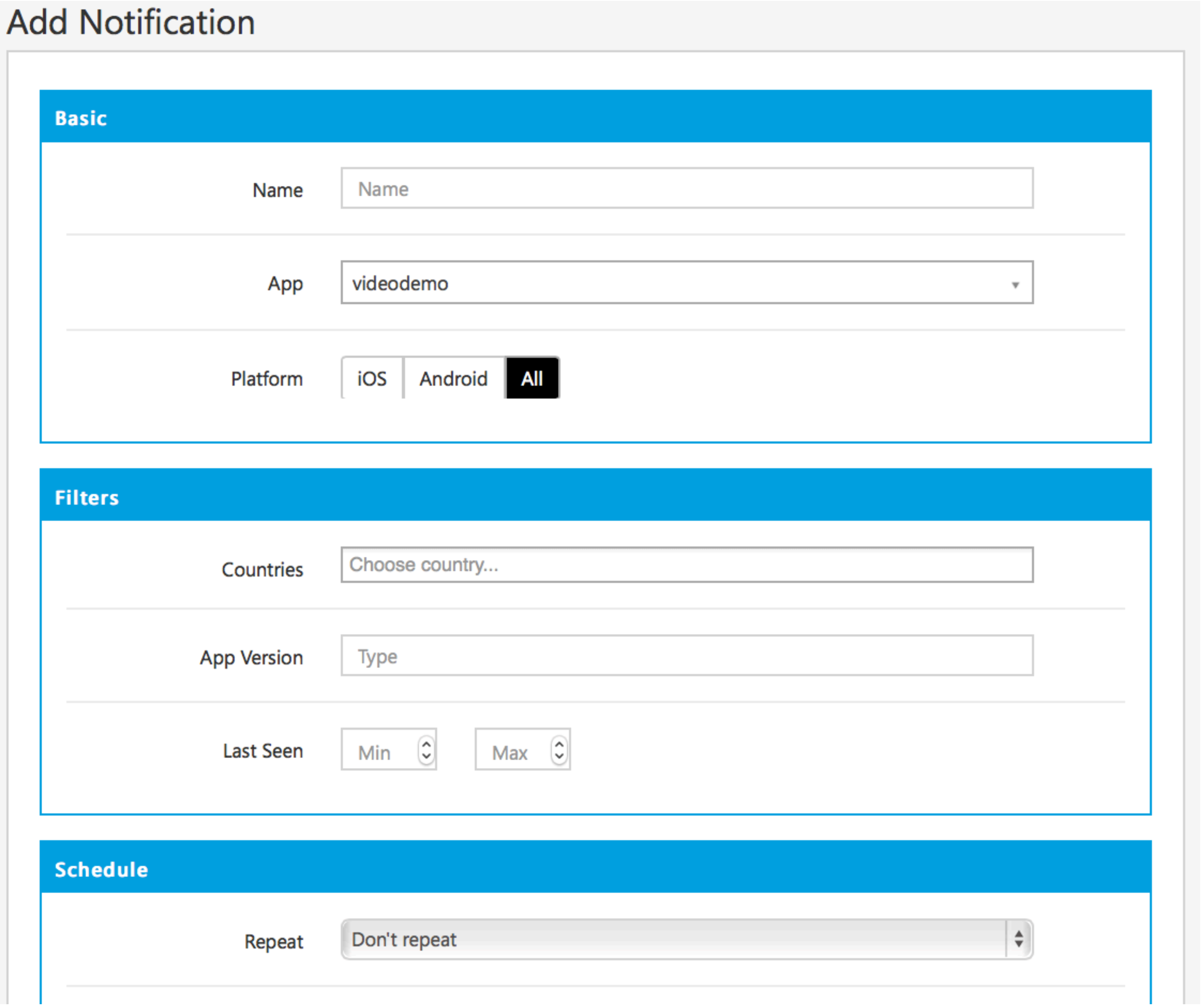
2. This API shows a folder chooser dialog where user can choose a particular folder.
3. User can also create a new folder where user wants to export the logs

Optional Parameters

PokktConfig also has provision for developers to provide extra user data available with them to pokkt. We currently support following data points: *name, age, sex, mobileNo, emailAddress, location, birthday, maritalStatus, facebookId, twitterHandle, setEducation, setNationality, setEmployment and setMaturityRating*.

In-App Notifications

Developer can add In-App notifications in their dashboard.



Add Notification

Basic

Name: Name

App: videodemo

Platform: iOS | Android | [✓ All]

Filters

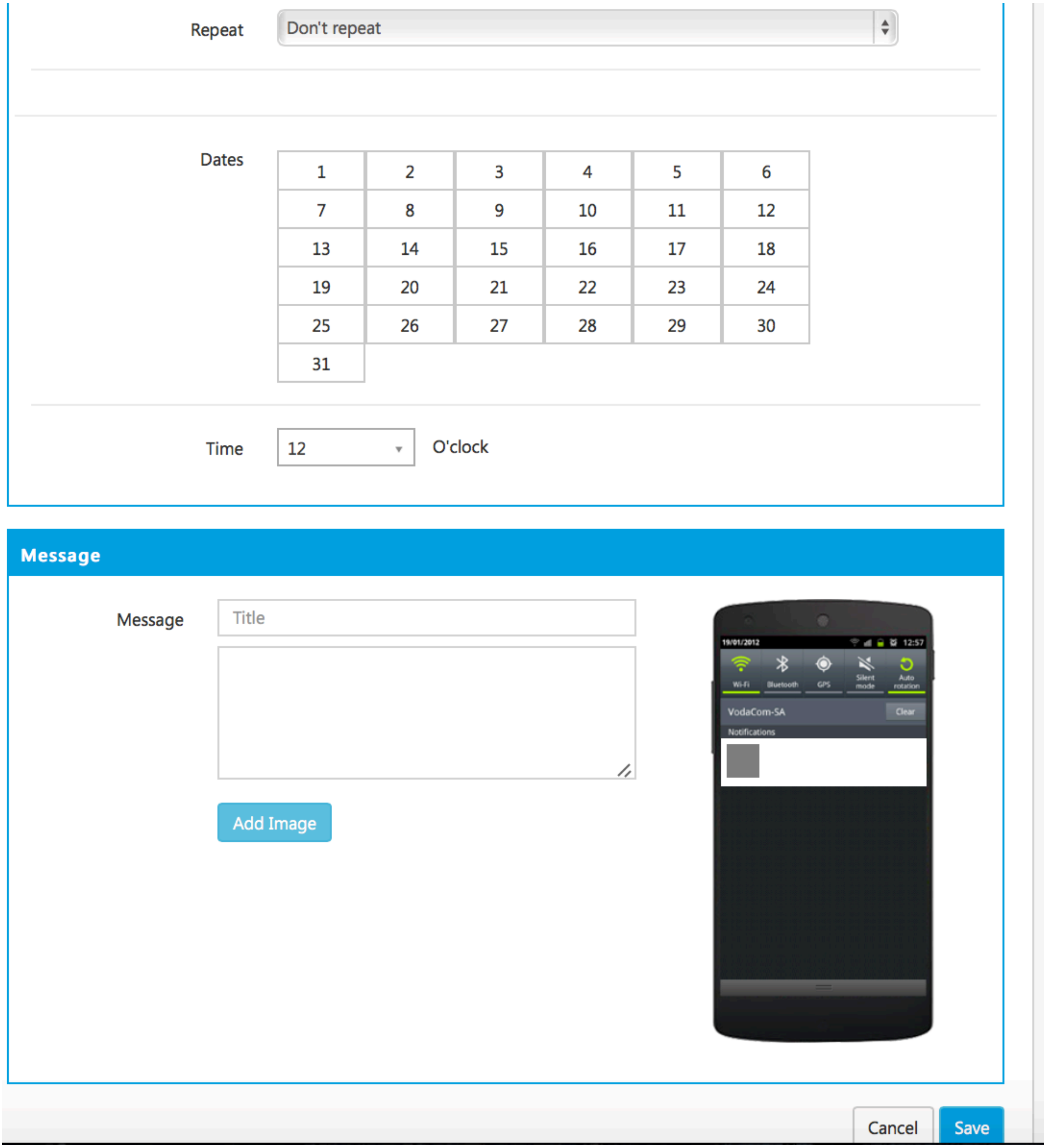
Countries: Choose country...

App Version: Type

Last Seen: Min | Max

Schedule

Repeat: Don't repeat



Repeat: Don't repeat

Dates

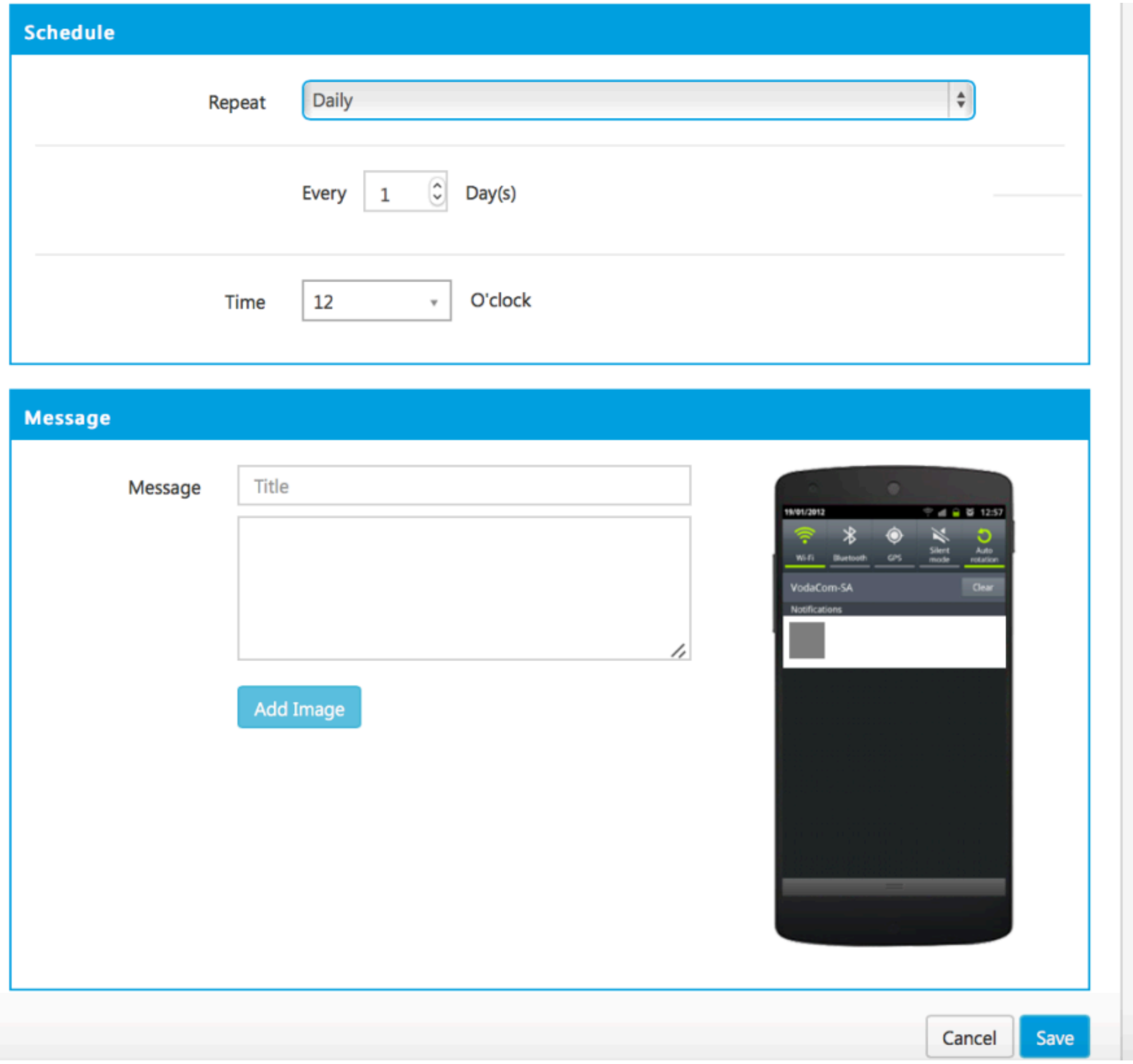
1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31					

Time: 12 O'clock

Message

Message: Title

Repeat schedule can be daily, weekly monthly. Daily Repeat can have options like frequency of repeat and time in hours of notification.



Schedule

Repeat: Daily

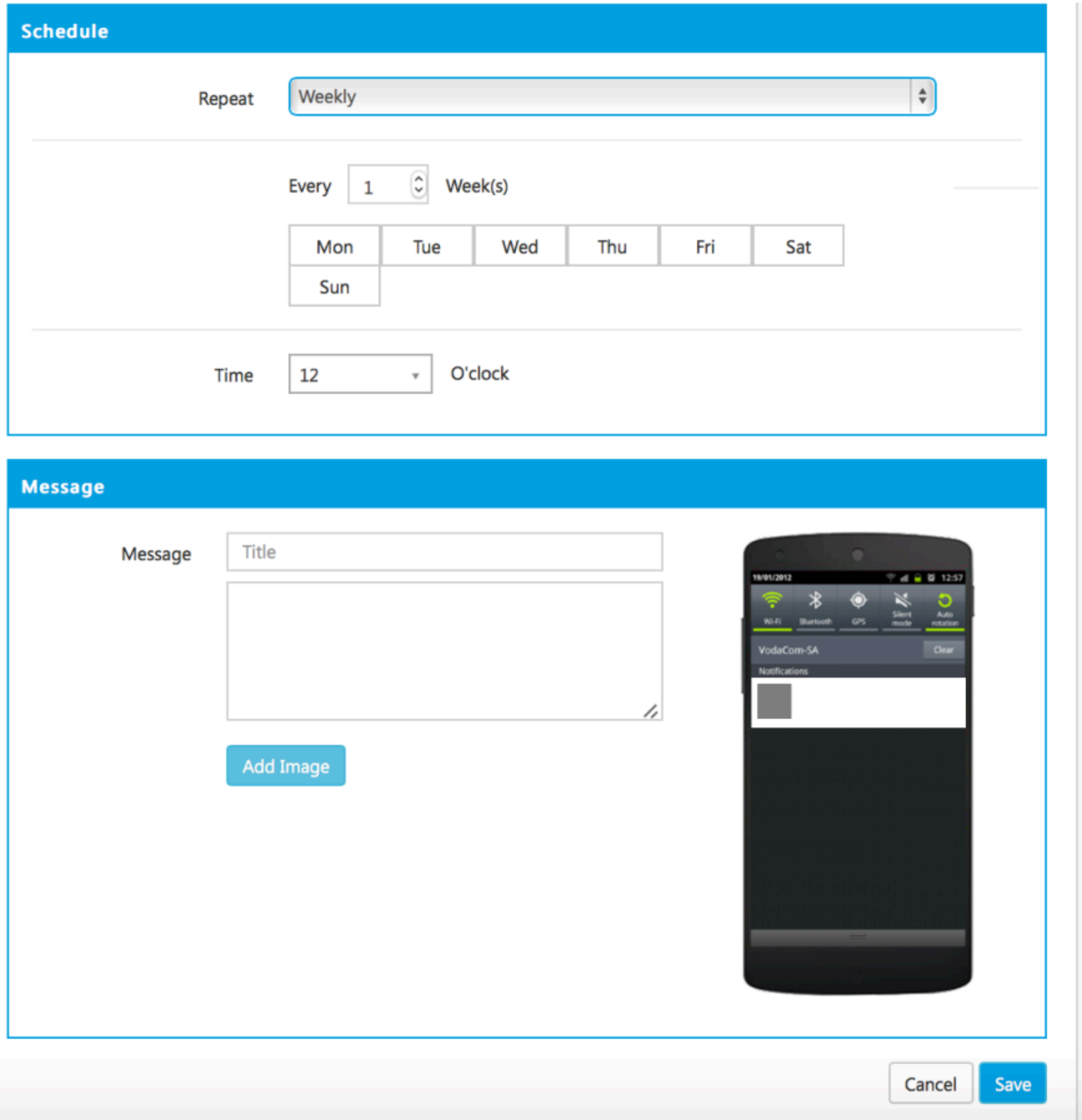
Every 1 Day(s)

Time: 12 O'clock

Message

Message: Title

Weekly repeat can have options like frequency of repeat in weeks, days of repeat and time in hours of notification.



Schedule

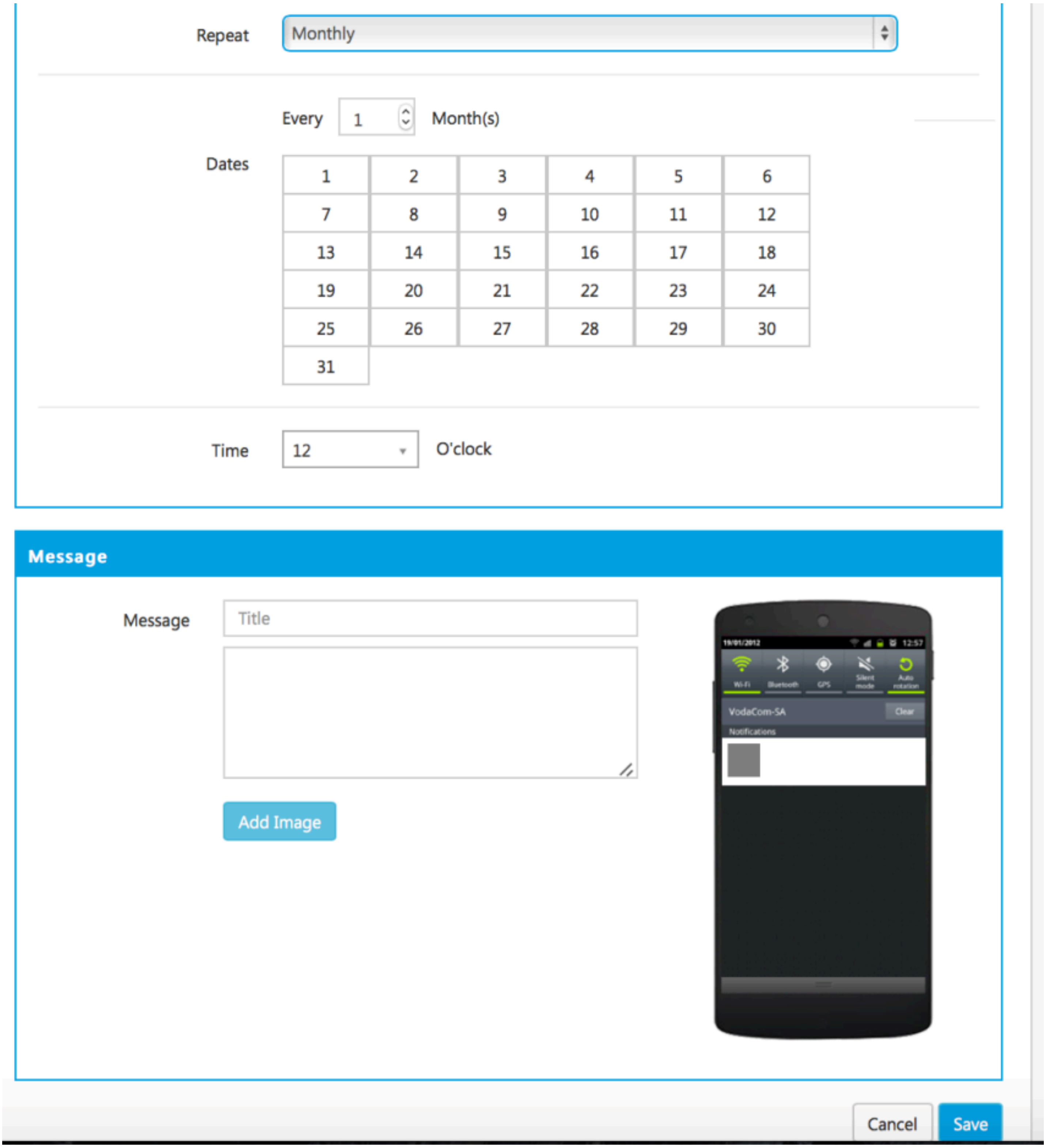
Repeat Weekly

Every 1 Week(s)

Time 12 O'clock

Message

Monthly repeat can have options like frequency of repeat in months dates of repeat and time in hours for notification .



Repeat: Monthly

Dates: Every 1 Month(s)

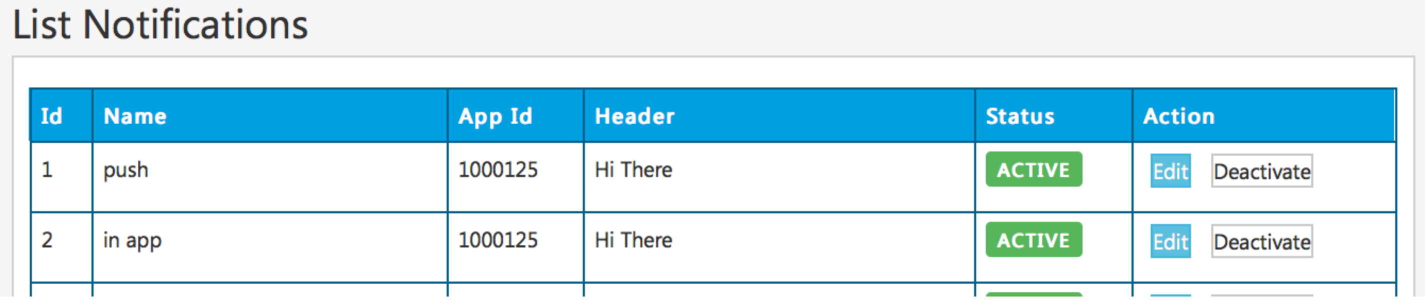
1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31					

Time: 12 O'clock

Message

Message: Title

For don't repeat case, there are options like dates and time in hour for notification. The notifications are listed and can be edited. Notifications can also be deactivated/activated.



List Notifications

Id	Name	App Id	Header	Status	Action
1	push	1000125	Hi There	ACTIVE	Edit Deactivate
2	in app	1000125	Hi There	ACTIVE	Edit Deactivate

4. Important Points

Please do not copy the code points from this pdf as it may introduce unwanted characters and space in your code. Instead please refer to sample app source code in Pokkt bundle. Please also refer to sample app source code for better understanding of implementation.