|  |
| --- |
| **Сервис доставки push-сообщений** |
| **Информационный лист**  **Отображение уведомлений средствами библиотеки** |

# Введение

В версии 3.3.0 библиотек для приложений на базе ОС Android предоставлена возможность автоматического отображения push-уведомлений, с использованием средств библиотек edna.

# Интеграция

Для метода PushBroadcastReceiver.onShortPushRecieved добавлена реализация по умолчанию. Если метод не переопределён в наследнике класса, или в методе наследника происходит вызов super-метода – уведомление будет автоматически отображено библиотекой.

# Возможности

Отрисовка уведомлений библиотекой позволяет использовать все возможности Android по отображению уведомлений. Например, при использовании сервиса Красивый Чек автоматически будет отображен переданный сервисом логотип. Управлять отображением можно с помощью набора дополнительных параметров, передаваемых вместе с уведомлением.

Библиотекой при отображении может быть обработан следующий набор дополнительных параметров:

Таблица 1. Дополнительные параметры обрабатываемые библиотекой

|  |  |
| --- | --- |
| Наименование параметра | Описание |
| title | Заголовок уведомления. |
| logoURL | URI иконки для отображения с уведомлением. Допускается использование изображений в форматах jpg и png. Рекомендуемое соотношение сторон - 1:1. |
| bigContentText | Альтернативный текст, для отображения в расширенном виде уведомления. На некоторых устройствах отображается вместо основного текста для обоих видов уведомления. |
| bigContentTitle | Альтернативный заголовок, для отображения в расширенном виде уведомления. На некоторых устройствах отображается вместо основного заголовка для обоих видов уведомления. |
| bigContentLogoURL | URI изображения, для отображения в расширенном виде уведомления. Допускается использование изображений в форматах jpg и png. Рекомендованное соотношение сторон – 2:1. При другом соотношении сторон, изображение будет обрезано до него на большей части устройств. При использовании кнопок в уведомлении, изображение будет дополнительно обрезано по высоте. |
| NotificationChannel | Канал уведомлений Android, в котором должно быть отображено уведомление. При отсутствии указанного канала, он будет создан при отображении уведомления. Если не передан, используется канал по умолчанию – “Default channel”. |
| deeplink | Строка, передаваемая в Intent при переходе по уведомлению. |
| actions | Массив наименований кнопок, добавляемых к уведомлению, и информации, передаваемой при переходе по ним. Должен быть передан строкой содержащей JSON-массив |
| sound | Звуковой файл используемый для уведомления/канала уведомлений. Файл должен быть добавлен в папку raw ресурсов приложения. Наименование файла передаётся без указания расширения. |
| vibrate | Шаблон вибрации. Целочисленный массив. В массиве поочерёдно указываются продолжительности активности и бездействия вибромотора в миллисекундах, начиная с бездействия. |
| lights | Цвет светового индикатора. Передаётся как цвет RGB в шестнадцатеричном формате. При ограниченной палитре индикатора, системой будет подобран самый близкий доступный цвет. При отсутствии индикатора – проигнорирован. |

Важно! Параметры sound, vibrate и lights устанавливаются для каждого сообщения на Android 7 и ниже. На Android 8 и выше, параметры используются при создании канала уведомлений NotificationChannel. После создания канала уведомлений изменить настройки нельзя. Все последующие сообщения направленные в данный канал будут отображены с установленными при создании канала настройками. Для отображения уведомлений с новым набором параметров необходимо создать новый канал.

Все указанные параметры являются необязательными. При отсутствии части параметров, соответствующие возможности уведомлений не будут использованы.

Пример передачи всех доступных параметров сообщения в формате JSON:

|  |
| --- |
| {  "title": "Заголовок",  "logoURL": " https://.../logo.png",  "bigContentText": "Альтернативный текст уведомления",  "bigContentTitle": "Альтернативный заголовок уведомления",  "bigContentLogoURL": "https://.../picture.png",  "NotificationChannel": "Имя канала уведомлений",  "deeplink": "test://deeplink\_action",  "actions": "[{\"title\":\"Кнопка 1\", \"action\":\"test://button1\_action\"}, {\"title\":\"Кнопка 2\", \"action\":\"test://button2\_action\"}]",  "sound": "ring",  "vibrate": "[200,500,200,500,200]",  "lights": "#00ff00"  } |

При клике на уведомлении или на одной из кнопок в уведомлении, пользователь будет направлен в приложение. В стартовое Activity приложения будет направлен Intent с данными переданными в уведомлении. Необходимо обработать полученную информацию в методах onCreate или onNewIntent. В данных Intent будет передана либо строка, переданная с уведомлением как deeplink, если пользователь кликнул по уведомлению. Либо будут переданы наименование кнопки и строка, переданная для неё. Пример кода обработки перехода пользователя по уведомлению:

|  |
| --- |
| override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  // deeplink – строка действия уведомления, по которому пользователь перешёл в приложение  // actionTitle – наименование кнопки, по которой пользователь перешёл в приложение  // action – строка действия, переданная с кнопкой, по которой пользователь перешёл в приложение  val action = intent.action  val actionTitle = intent.extras?.getString(action)  val deeplink = intent.extras?.getString("deeplink")  if (!deeplink.isNullOrEmpty()) {  // TODO: Дальнейшая навигация клиента с учётом переданной строки (параметр deeplink)  } else if (!action.isNullOrEmpty()) {  // TODO: Дальнейшая навигация клиента с учётом переданной строки (параметр action)  }  } |