

Программистам

Инструкции по настройке сайта для работы в Converzilla

- [Интеграция с Push-уведомлениями](#)
- [Скрытие блоков сайта в приложении \(Через стили\)](#)
- [Настройка звука для push-уведомлений](#)
- [Интеграция с Converzilla](#)
- [Установка и настройка киоск-приложения через USB и ADB](#)

Интеграция с Push-уведомлениями

После запуска приложения, UserID из OneSignal будет автоматически передаваться на страницы сайта при загрузке через COOKIE. Вы можете перехватить этот UserID на стороне сайта и связать его с текущим пользователем.

Используя этот UserID, вы можете отправлять персональные push-уведомления вашим клиентам.

Пример получения UserID на PHP:

```
if(!isset($_COOKIE['ONESIGNAL_USER_ID'])) {  
    $userID = $_COOKIE['ONESIGNAL_USER_ID'];  
}
```

Стоит учесть что ONESIGNAL_USER_ID может появиться не сразу а например через 10-30 секунд после начала использования приложения при самом первом запуске.

<https://documentation.onesignal.com/reference/push-notification>

```
$fields = array(  
    'app_id' => 'ВАШ app_id',  
    'include_player_ids' => ["ONESIGNAL_USER_ID"]  
    'contents' => ["en" => "My Push"],  
    'headings' => ["en" => "Hi!"],  
    'data' => [  
        ['url'=>"https://site.com/my/link"  
        ]],  
    );  
  
$ch = curl_init();  
curl_setopt($ch, CURLOPT_URL, "https://api.onesignal.com/notifications");  
curl_setopt($ch, CURLOPT_HTTPHEADER, array(  
    'Content-Type: application/json; charset=utf-8',
```

```
'Authorization: Basic XXXXXXXXXX',
));
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_HEADER, false);
curl_setopt($ch, CURLOPT_POST, true);
curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($fields));
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
curl_exec($ch);
curl_close($ch);
```

Пример получения UserID в JS:

```
if(window.userId){
  console.log("USER ID", window.userId);
}
```

Аналогично userId может появиться не сразу, поэтому можно использовать функцию попытки получения userId, пока он не станет доступным:

```
function getUserId(callback){

  let tryGetUserIdInterval = setInterval(()=>{

    if(window.userId){
      callback(window.userId);
      clearInterval(tryGetUserIdInterval);
    }

  },3000);
}

getUserId((userId)=>{

  console.log("userId", userId);
})
```

Навигация через Push-уведомления

Если вы хотите, чтобы при нажатии на push-уведомление пользователь не только перешел в приложение, но и был направлен на одну из страниц сайта, передайте параметр url в параметре data запроса.

Если в приложении только 1 язык для уведомлений, например только русский, то используйте ключ en, так как он является ключом по умолчанию.

Пример на PHP:

⚠ **Следует учесть, что если ссылка будет не с вашего сайта, то она откроется в браузере.**

Отправка через теги

Вы можете отправлять push-уведомления не только по UserID, но и по тегу, который вы установите для пользователя. Это удобно, если вы хотите связать пользователя с его уникальным идентификатором, например email или phone.

Например, при авторизации на сайте вы можете установить тег email или phone для пользователя, а затем использовать его для отправки push-сообщений через OneSignal API.

Пример отправки уведомления по тегу (на PHP с использованием curl):

<https://documentation.onesignal.com/reference/push-notification>

```
$fields = [  
    'app_id' => 'ВАШ_APP_ID',  
    'filters' => [  
        ["field" => "tag", "key" => "email", "relation" => "=", "value" => "user@example.com"]  
    ],  
    'contents' => ["en" => "My Push"],  
    'headings' => ["en" => "Hi!"],  
];  
  
$ch = curl_init();  
curl_setopt($ch, CURLOPT_URL, "https://api.onesignal.com/notifications");  
curl_setopt($ch, CURLOPT_HTTPHEADER, [  
    'Content-Type: application/json',  
    'Authorization: Basic ВАШ_REST_API_KEY'
```

```
]);  
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);  
curl_setopt($ch, CURLOPT_POST, true);  
curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($fields));  
  
$response = curl_exec($ch);  
curl_close($ch);  
  
echo $response;
```

Скрытие блоков сайта в приложении (Через стили)

При конвертации вашего сайта в мобильное приложение с помощью сервиса Converzilla, в содержимое сайта автоматически добавляются определенные стили и классы. Эти изменения предназначены для оптимизации отображения вашего сайта в мобильном приложении.

Автоматическое добавление стилей

При конвертации вашего сайта в мобильное приложение с помощью Converzilla, в содержимое сайта добавляются следующие CSS классы:

1. **.converzilla-hide**: Этот класс скрывает элементы, устанавливая их свойство `display` в значение `none`:

```
.converzilla-hide {
  display: none !important;
}
```

2. **.converzilla-show**: Этот класс отображает элементы, устанавливая их свойство `display` в значение `block`:

```
.converzilla-show {
  display: block !important;
}
```

Кроме того, в тег `<body>` добавляется класс `converzilla`, который позволяет вам вносить изменения в стили сайта, чтобы он отображался по-другому в мобильном приложении.

Пример использования

Для изменения стилей элементов вашего сайта, когда он отображается в мобильном приложении, вы можете использовать класс `converzilla`, добавленный к тегу `<body>`. Например:

```
body.converzilla .your-element {
  background-color: #f0f0f0;
```

```
color: #333;  
}
```

Этот пример изменит фон и цвет текста элемента с классом `your-element` только при отображении в мобильном приложении.

Тестирование

Чтобы протестировать изменения в отображении вашего сайта при конвертации его в мобильное приложение с помощью Converzilla, вы можете использовать следующий JavaScript код. Этот код добавит необходимые стили и класс converzilla к тегу <body> на вашем сайте.

```
(function() {  
    // Функция для добавления стилей  
    function addConverzillaStyles() {  
        var styleElement = document.createElement('style');  
        styleElement.type = 'text/css';  
        styleElement.innerHTML = `  
            .converzilla-hide {  
                display: none !important;  
            }  
            .converzilla-show {  
                display: block !important;  
            }  
        `;  
        document.head.appendChild(styleElement);  
    }  
  
    // Функция для добавления класса к body  
    function addConverzillaClass() {  
        document.body.classList.add('converzilla');  
    }  
  
    // Проверка, если документ уже загружен  
    if (document.readyState === 'complete' || document.readyState === 'interactive') {  
        addConverzillaStyles();  
        addConverzillaClass();  
    } else {  
        // Если документ еще не загружен, добавляем слушатель события DOMContentLoaded  
        document.addEventListener('DOMContentLoaded', function() {
```

```
addConverzillaStyles();
addConverzillaClass();
});
}
})();
```

Инструкции по использованию

- Откройте ваш сайт в браузере.
- Откройте инструменты разработчика (обычно это F12 или Ctrl+Shift+I).
- Перейдите на вкладку “Консоль”.
- Вставьте код выше в консоль и нажмите Enter.

Полезные советы

- **Используйте специфичность:** Убедитесь, что ваши стили достаточно специфичны, чтобы переопределить существующие стили на сайте.
- **Тестируйте изменения:** Всегда проверяйте, как ваш сайт выглядит в мобильном приложении, чтобы убедиться, что все стили применяются корректно.

Конвертация сайта в мобильное приложение с помощью Converzilla предоставляет мощные инструменты для управления отображением вашего сайта. Используя предоставленные классы и стили, вы можете легко адаптировать внешний вид вашего сайта для оптимального взаимодействия пользователей в мобильном приложении.

Настройка звука для push-уведомлений

Подготовка файлов

Подготовьте набор звуковых файлов которые вы будете использовать для push-уведомлений.

Для использования кастомных звуков в уведомлениях на Android существуют определённые требования.

Вот основные из них:

Поддерживаемые форматы: для Android - `mp3`, для iOS - `wav`.

Имя файла: должно быть написано маленькими буквами и без пробелов. Например, файл должен называться как `audio_1.mp3` или `notification_sound.wav`.

Длительность: Нет строгих ограничений на длительность звукового файла, но рекомендуется, чтобы файл был коротким (например, 2-5 секунд), так как уведомления обычно не требуют долгих звуков.

Каналы: Рекомендуется использовать моно

Громкость: Звуковые файлы не должны быть слишком громкими.

Передайте архив с аудио-файлами для настройки и создания новых сборок мобильного приложения.

Для обновления (добавления) или замены звуковых файлов требуется полное обновление приложения в магазинах приложений.

Настройка в OneSignal

Android

В Android 8+ были введены категории уведомлений, которые необходимо настроить для кастомизации звуков уведомлений. OneSignal будет использовать звук, установленный в канале уведомлений для всех версий Android.

В личном кабинете откройте приложения для которого выполняете настройку.

Settings

- Platforms
- Integrations
- Roles
- Keys & IDs
- Messaging**
- Usage

Upgrade

Your Android audience may stop takes just a few minutes when y

Update Android Settings

Total Subscription Records

325

Subscription Trends

Keys & IDs

Messaging

Usage

Upgrade

Documentation

Support

Stop Tracking

Android Notification Channels

Notification updates in Android Or...

Notifications in Android O

No groups created. Create a group to add notification channels

Channels Documentation

Add Group

Create Notification Group

Name *

Notification Group

Submit

Android Notification Channels

Notification channels are an Android 8.0 (API level 26) feature which gives users finer control

Channels documentation.

Notification Group

Add Channel

Add Group

Не добавляйте расширение файла при указании звукового ресурса. Например, `audio_1`, а не audio_1.mp3

Нажмите на созданный канал, чтобы увидеть уникальный ID канала.

При отправке push-сообщений используйте свойство `android_channel_id` API для создания уведомления. Звук, установленный в категории, будет работать для всех версий Android.

iOS

Используйте свойство `ios_sound` в API Create Notification, **добавляйте расширение** файла при указании звукового ресурса, например `audio_1.wav`

Интеграция с Converzilla

Чтобы ваша платформа сайтов мог корректно взаимодействовать с мобильным приложением Converzilla через WebView, необходимо внести следующие изменения:

Добавьте стили для управления отображением элементов

Эти стили позволяют платформе Converzilla показывать или скрывать нужные элементы сайта из приложения:

```
/* Скрывает элемент */
.converzilla-hide {
  display: none !important;
}

/* Показывает элемент */
.converzilla-show {
  display: block !important;
}
```

Добавьте эти стили в основной CSS сайта.

Добавьте вызов передачи пользовательского идентификатора для пуш-уведомлений (OneSignal)

После загрузки страницы нужно передать в приложение уникальный идентификатор пользователя (**login**, **phone**, либо другой ключ).

Пример на чистом JavaScript:

```
<script>
document.addEventListener("DOMContentLoaded", function () {
  try {
    var login = getCurrentUserLogin(); // Получите логин пользователя с вашего сайта любым доступным
способом
    if (login && typeof window.oneSignalSendTag === 'function') {
      window.oneSignalSendTag('login', login);
    }
  }
});
```

```
    }  
  } catch (e) {  
    console.error('Ошибка при отправке oneSignalSendTag:', e);  
  }  
});
```

// Эта функция — пример, замените её на свой код получения логина/номера телефона

```
function getCurrentUserLogin() {  
  // Например:  
  // return document.querySelector('input[name="login"]').value;  
  return 'example_login_123';  
}  
</script>
```

📄 Пояснения:

- Метод `window.oneSignalSendTag(key, value)` отправляет уникальный идентификатор пользователя в приложение Converzilla для настройки пуш-уведомлений.
- Ключ (`login`, `phone`, и др.) и значение определяются разработчиком сайта.
- Вызов должен быть после загрузки страницы и (желательно) после того, как пользователь залогинился.

⚙️ Где вызывать?

Можно вызвать:

- На странице входа (`login`)
- На странице профиля (если там есть логин/номер телефона)
- На любой другой странице, где доступен уникальный пользовательский идентификатор.

Установка и настройка киоск-приложения через USB и ADB

Что понадобится

- Компьютер **Windows** или **Mac**
- USB-кабель
- Android-планшет (желательно **новый или сброшенный до заводских настроек**)
- Файл apk,
- ID приложения вида ru.site.apk (**обязательно в командах ниже замените на свой, в примере будет использован ru.site.app**)

1. Подготовка планшета

Если устройство не сброшено — сделайте сброс:

“ **△ Важно:** установить приложение в киоск-режиме можно **только на “чистое” устройство.** ”

Как сбросить:

1. Откройте **Настройки** → **Система** → **Сброс / Восстановление** → **Сброс до заводских настроек**
2. Подтвердите сброс. Планшет перезагрузится.

2. Включите «режим разработчика» и отладку по USB

1. Откройте **Настройки** → **О телефоне / Об устройстве**
2. Найдите пункт **Номер сборки** (Build number)
3. Нажмите по нему **7 раз**, пока не появится сообщение: «*Вы стали разработчиком*»
4. Назад → Зайдите в **Система** → **Для разработчиков (Developer options)**

5. Включите **Отладку по USB (USB Debugging)**

3. Установите ADB (на компьютере)

► Вариант А: Для Windows

1. Скачайте **ADB Platform Tools**:

- <https://developer.android.com/tools/releases/platform-tools>

2. Распакуйте архив в удобную папку, например: C:\adb

► Вариант В: Для macOS

1. Откройте **Терминал**

2. Установите ADB через Homebrew (если он установлен):

```
brew install android-platform-tools
```

Или скачайте вручную:

<https://developer.android.com/tools/releases/platform-tools>

4. Подключите планшет к компьютеру через USB

- Разрешите отладку по USB на планшете, если появится запрос.

5. Установите приложение

Перетащите .apk в папку с ADB. Например C:\adb

Затем в консоле:

► Windows:

```
cd C:\adb  
adb install -r app-release.apk
```

► Mac:

```
cd ~/Downloads/platform-tools  
./adb install -r app-release.apk
```

□ Должно появиться `Success`

6. Назначьте приложение как Device Owner (админ устройства)

△ Делается **один раз**, и только на сброшенном устройстве.

```
adb shell dpm set-device-owner ru.site.app/.MyDeviceAdminReceiver
```

ru.site.app замените на ваш ID приложения!

□ Если всё ок, появится сообщение `Success: Device owner set to package ru.lolapizza.kso`

7. Перезагрузите планшет

```
adb reboot
```

После включения:

- Приложение **автоматически запустится**
- Нельзя выйти из него
- Все кнопки и меню будут заблокированы
- Работает **Kiosk Mode**

□ ГОТОВО!

Как выйти из режима киоска (для администратора)

Только при необходимости, с компьютера:

```
adb shell am stop-user 0
```

```
adb shell dpm remove-active-admin ru.site.app/.MyDeviceAdminReceiver
```

ru.site.app замените на ваш ID приложения!

После этого можно удалить приложение вручную.

Как обновить приложение

Если планшет подключён к компьютеру, и у вас есть .apk новой версии:

1. Подключите планшет к компьютеру по USB
2. Убедитесь, что включена отладка по USB
3. Поместите app-release.apk в ту же папку, где adb.exe
4. Откройте терминал / командную строку
5. Выполните:

```
adb install -r app-release.apk
```