

Создание мобильной версии сайта

На текущий момент при построении мобильных версий сайтов существуют 3 основных способа их построения:

- Адаптивный дизайн;
- Отдельная мобильная версия сайта;
- RESS (Responsive Design + Server Side).

Каждый из способов обладает своими плюсами и минусами:

Адаптивный дизайн

Для реализации адаптивного дизайна обычно используются CSS3 Media Queries. В зависимости от размера экрана, пользователь может видеть разные картинки:

```
@media screen and (max-width: 1600px) {  
  div.for-example {width: 1500px;}  
}  
@media screen and (max-width: 1280px) {  
  div.for-example {width: 1100px;}  
}  
@media screen and (max-width: 1024px) {  
  div.for-example {width: 980px;}  
}
```

Преимущества адаптивного дизайна

- *Удобство разработки* — при адаптивной верстке вся структура сайта автоматически подстраивается под различную ширину экрана. Для того, чтобы получить рабочий продукт не нужно писать все с нуля — достаточно лишь подправить CSS и HTML. С учетом наличия фреймворков, подобных [Bootstrap](#) подобная разработка не является очень сложной при стандартной реализации. К тому же, поддержка такого продукта будет относительно простой задачей.

- *Один URL* — избавляет нас от ненужных редиректов, и надобности пользователю запоминать адрес мобильной версии (пусть даже это просто приставка m.). Так же наличие единого адреса позитивно скажется на продвижении сайта, поскольку поисковику будет «удобнее» работать.

Недостатки адаптивного дизайна

- *Разные задачи* - типовые задачи «мобильных» пользователей крупных сайтов обычно отличаются от задач пользователей ПК. Если вы являетесь клиентом банка, то, скорее всего, в мобильной версии сайта вас будет интересовать очень ограниченный спектр информации — адреса ближайших отделений, банкоматов и т.д.

- В целом, при адаптивной верстке самый распространенный подход — делать копию обычного сайта, реализовать в верстке под телефоны потребности всех групп целевой аудитории. Но тогда о юзабилити можно забыть. Нужные пяти процентам посетителей второстепенные разделы создадут неудобства для основной массы клиентов.

- *Медленная загрузка* - «вес» сайтов остается серьезным препятствием для пользователей мобильных телефонов. Это означает, что некоторые типичные для десктопных сайтов активные элементы, в том числе embedded-карты, ролики, кредитные калькуляторы и меню с анимацией на мобильных сайтах должны быть заменены на более легкие альтернативы. Может ли адаптивный дизайн дать нам такую возможность? В популярной реализации пользователь с маленьким экраном должен загрузить страницу целиком, чтобы увидеть лишь ее часть. Например, если десктопная версия верстки главной весит 200 Кб, а мобильная — еще 50 Кб, для просмотра придется скачать 250 Кб. Конечно, можно использовать сжатие кода страницы, но лишние запросы к серверу все равно будут идти.

- *Безысходность* - Одно из неоспоримых преимуществ мобильной версии: если не нравится, ее можно отключить, перейти на обычный домен. Сайты с адаптивным дизайном не дают этого простого, но важного выбора. Если адаптированная верстка неудобна, забагована или если она скрывает важный элемент навигации, пишете пропало: вы ничего не можете сделать, чтобы увидеть его снова. Придется бежать искать десктоп или сайт конкурента. Можно придумать «костыли» для обхода этого ограничения (использовать cookie и подключать разные таблицы стилей). Но такой подход усложняет разработку.

Идея разработки мобильной версии в адаптивном дизайне достаточно популярна, несмотря на вышеуказанные минусы. В частности, эту концепцию целиком поддерживают такие гиганты как, например, Google.

Отдельная мобильная версия сайта

Чтобы сделать сайт удобным для мобильных пользователей часто создают так же отдельные версии сайтов — специально ориентированные на пользователя со смартфоном/планшетом. Наиболее распространена практика — перенаправление мобильных пользователей на специальный поддомен (m.example.com, mobile.example.com и т.д.). Наверное, в 99% случаев, мобильная версия представляет собой урезанную основную версию — лишь с тем функционалом, который, по мнению разработчиков, будет необходим и полезен пользователям мобильных устройств и планшетов.

Преимущества мобильной версии

- *Легкость в изменениях* — поскольку сайт существует, де-факто, отдельно от основной версии, вносить в него правки, связанные только с мобильной версией значительно проще, поскольку мобильная версия чаще всего не предоставляет избыточного, ненужного функционала.

- *Удобство для пользователя* — мобильная версия обычно сильно упрощена по сравнению с десктопной версией, поэтому пользователю не нужно будет далеко ходить за нужной ему информацией.

- *Быстрота* — из-за того же упрощения сайта, мобильная версия грузится быстрее. Это существенно для пользователей, которые все ещё ходят через GPRS или слабый 3G.

- *Выбор* — чаще всего, в мобильной версии есть возможность перейти на основную версию сайта.

Недостатки мобильной версии

- *Несколько адресов* — для десктопной и мобильной версии разные адреса. Для одних это может оказаться плюсом, для других же — крайне раздражительным фактором, когда для того, что бы удобно посмотреть сайт, нужно запоминать ещё один адрес. Так же есть проблемы и с поисковиками: во избежание дублированного контента SEO-специалистам нужно использовать мета-теги `rel=«alternative»` и `rel=«canonical»`. Кроме того, когда пользователь мобильного поиска Google кликает по ссылке в результатах, он попадет на десктопную версию или перенаправлен на мобильную. Но если мобильная версия этой страницы не существует, он получит сообщение об ошибке.

- *Неудобство для пользователя* — для десктопной и мобильной версии разные адреса. Для одних это может оказаться плюсом, для других же — крайне раздражительным фактором, когда для того, что бы удобно посмотреть сайт, нужно запоминать ещё один адрес. Так же есть проблемы и с поисковиками: во избежание дублированного контента SEO-специалистам нужно использовать мета-теги `rel=«alternative»` и `rel=«canonical»`. Кроме того, когда пользователь мобильного поиска Google кликает по ссылке в результатах, он попадет на десктопную версию или перенаправлен на мобильную. Но если мобильная версия этой страницы не существует, он получит сообщение об ошибке.

- *Ограниченность* — создание отдельного мобильного сайта означает избавление от части содержания и функциональности. Кроме того, у вас может быть два различных набора контента, который может негативно сказаться на общей информационной картине.

В целом, создание мобильных версий сайтов оправдывает себя, в частности, для крупных проектов. Как например, Amazon использует у себя специальную, мобильную версию сайта.

RESS

Сам Google, хоть и поддерживает использование веб-мастерами адаптивного дизайна, однако в своих продуктах использует иную систему. Если зайти, к примеру, на главную страницу под разными User-Agent-ами, то можно увидеть различный HTML для различных устройств. RESS — Responsive Design + Server Side. Пример реализации:

```
$DS = DIRECTORY_SEPARATOR;
require_once( dirname(__FILE__) . $DS . 'libraries' . $DS . 'browser.php');
$device = BBrowser::detectDevice();
if($device == DEVICE_TYPE_MPHONE){
    $tpl = 'template.m.php';
}
else if($device == DEVICE_TYPE_TABLET){
    $tpl = 'template.t.php';}
else{
    $tpl = 'template.php';
}
include( dirname(__FILE__) . $DS . 'templates' . $DS . $tpl);
```

Плюсы RESS

Фактически, метод может включать в себя преимущества как отдельной мобильной, так и адаптивной версии сайтов, в зависимости от реализации. Из того, что будет новым:

- *Минимизация трафика* — Ненужные JavaScript могут быть удалены из HTML, который освобождает CPU, память и кэш на мобильном устройстве. Так же может быть специально оптимизирована html и css.

- *Возможно использовать таргетирование* — например, для Android-устройств предлагать скачать приложение с GooglePlay, а для Apple – с iTunes. Для каждого устройства можно сделать свою вёрстку.

Минусы RESS

- *Сложность в разработке* — подобный метод потребует соответствующей настройки сервера и труда большего количества программистов. Так же необходимо будет делать несколько различных вариантов верстки.

- *Механизм определения устройств* — к сожалению, даже в наше время ещё не доведен до совершенства. Истории с тем, как чей-то редкий телефон не определился как мобильное устройство, всплывают довольно часто.

В целом же, RESS — самый лучший из трех предложенных вариантов, однако требует намного больших трудозатрат при разработке.

На мой личный взгляд — идеального варианта, который бы стоило использовать всем, нет. Лучший вариант для меня — это RESS. Однако, это один из непростых вариантов, поскольку требует больших усилий для реализации. В целом же все 3 варианта имеют свои плюсы и минусы, причем в зависимости от сути и направленности сайта.

Реализация одного из вариантов мобильной версии сайта

Итак, поддомен создали, разместили на нем требуемые нам странички. Основные задачи, стоящие перед нами после этого:

1. Правильный редирект мобильных и десктопных устройств на соответствующие версии сайта
2. Возможность просмотра полной версии с мобильных устройств
3. Так как на урезанной версии сайта некоторые страницы из полной отсутствуют, то надо обрабатывать эти случаи, не теряя посетителей
4. Удобство мобильной версии сайта для пользователей
5. Избегание всевозможных сеошных проблем — появления дублей и т.д.

Первую задачу с редиректом решаем следующим образом:

В `htaccess` **полной версии** добавляем код:

```
RewriteEngine on
```

```
RewriteCond %{HTTP_USER_AGENT} ((.*iPhone.*)|(.*iPod.*)|(.*BlackBerry.*  
)|(.*Android.*Mobile.*)|(.*Windows\ CE.*)|(.*IEMobile.*)|(.*Opera\ Mini.*)|(.*Opera\  
Mobi.*))
```

```
RewriteCond %{REQUEST_FILENAME} !\.(jpg|gif|png|css|js|txt|ico|pdf|bmp|tif|  
mp3|wav|wma|asf|mp4|flv|mpg|avi|csv|doc|docx|xls|xlsx|ppt|pptx|zip|rar|tar|gz|dmg|iso)$ [NC]
```

```
RewriteCond %{HTTP_REFERER} !^http://yourdomain\.ru(/)?
```

```
RewriteCond %{QUERY_STRING} !no_redirect=true [NC]
```

```
RewriteRule ^(.*)$ http://m.yourdomain.ru/$1 [L,R=302]
```

В `htaccess` **мобильной версии** пишем следующее:

```
RewriteEngine on
```

```
RewriteCond %{HTTP_USER_AGENT} !Windows\ NT.+Touch [NC]
```

```
RewriteCond %{HTTP_USER_AGENT} Windows\ NT\ 6|Macintosh|Ubuntu|Lin  
ux\ (x86_64|i686)|CrOS [NC]
```

```
RewriteCond %{QUERY_STRING} !no_redirect=true [NC]
```

```
RewriteCond %{HTTP_REFERER} !^http://m\.yourdomain\.ru(/)?
```

```
RewriteRule ^(.*)$ http://yourdomain.ru/$1 [L,R=302]
```

Примерная расшифровка:

Десктопные пользователи, попавшие на мобильную версию (вообще говоря они туда никак не должны попадать, но на всякий случай) редиректятся на полную версию, мобильные пользователи с полной версии — на мобильную. При этом, используются следующие исключения:—при наличии в УРЛе параметра `no_redirect=true` (неважно у какого пользователя и на какой версии) — редирект не происходит;

— если реферером пользователя является та версия сайта, на которой он находится сейчас — редирект не происходит;

— если мобильный пользователь делает запрос к конкретному файлу на полной версии сайта — редирект не происходит.

Причина для последнего исключения очевидна, а вот первые два относятся уже ко второму пункту нашей повестки дня — возможности просматривать полную версию сайта с мобильных устройств.

Третий пункт. Ошибки 404 на мобильной версии.

На полной версии сайта может быть примерно 70+ страниц. Но для мобильной необходимо около 8-10. Соответственно, мобильные пользователи, придя с поисковика, часто натыкались на 404 ошибку. Разместим там информацию, о том, что, нужная страничка в полной версии, но % отказов все равно может быть очень высок. Поэтому в случае: если на мобильном сайте получаем 404-ю ошибку, то редиректим пользователя на полную версию с тем же урлом, добавив незабвенный `no_redirect=true`. Как это сделано в `htaccess` мобильной версии:

```
ErrorDocument 404 /404.php
```

В этом файле:

```
<?
$ref=$_SERVER['REQUEST_URI'];
if ($ref!="") $ref=$ref.'?no_redirect=true';
header('Location: http://yourdomain.ru'.$ref);
?>
```

Четвертая задача: удобство для мобильных пользователей

Решается с помощью всем известного инструмента: developers.google.com/speed/pagespeed/insights

Все рекомендации расписаны,— в `header` мобильной версии добавляем `<meta name=viewport content="width=device-width, initial-scale=1">`

И следим за правильным расположением и масштабированием контента.

В стили добавляем `body{max-width:900px;margin:0 auto;}` для того, чтобы при горизонтальном просмотре сайт сильно не растягивался в ширину.

И, наконец, проблемы SEO

Есть [замечательное руководство](#) гугла по сео-оптимизации для мобильных устройств. На обычной странице (`http://www.example.com/page-1`) добавьте следующий код:

```
<link rel="alternate" media="only screen and (max-width: 640px)"
href="http://m.example.com/page-1" >
```

а на странице для мобильных устройств (<http://m.example.com/page-1>) используйте такие атрибуты:

```
<link rel="canonical" href="http://www.example.com/page-1" >
```

URL, который размещен на странице мобильного сайта и указывает на аналог этой страницы для обычных компьютеров, обязательно нужно добавить тег `rel=«canonical»`.

Помимо этого, указываем мобильность сайта в файле `sitemap`:

```
<?xml version="1.0" encoding="UTF-8" ?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
  xmlns:mobile="http://www.google.com/schemas/sitemap-mobile/1.0">
<url>
<loc>http://m.photocrew.ru/</loc>
<lastmod>2015-05-26T18:07:54+06:00</lastmod>
<changefreq>never</changefreq>
<priority>1.0</priority>
<mobile:mobile/>
</url>
```

И так для каждого url.

Яндекс утверждает, что он сам распознает стандартные поддомены типа `m.example.com`, `pda.example.com` и т.п.

Для пущей надежности можно еще каждой мобильной страничке указать соответствующий доктайп:

```
<!DOCTYPE html PUBLIC
"-//WAPFORUM//DTD XHTML Mobile 1.2//EN"
"http://www.openmobilealliance.org/tech/DTD/xhtml-mobile12.dtd">
```

Думаю, уж после такого комплекса мер, поисковики должны адекватно разобраться, где какая версия сайта.

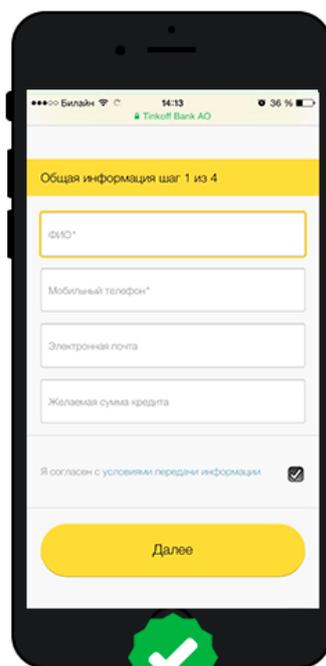
При работе над приложениями бывают ошибки, которые необходимо предусмотреть:

1. *Слишком длинные формы захвата.* Это настоящий убийца конверсии на любом сайте! Особенно к этому чувствительны мобильные версии, так как заполнять анкеты на телефоне, крайне, не удобно. Избавляйтесь от них.



CITYBANK.RU

Убийственные 10 полей, которые не умещаются на одном экране. Незаметная кнопка СТА.



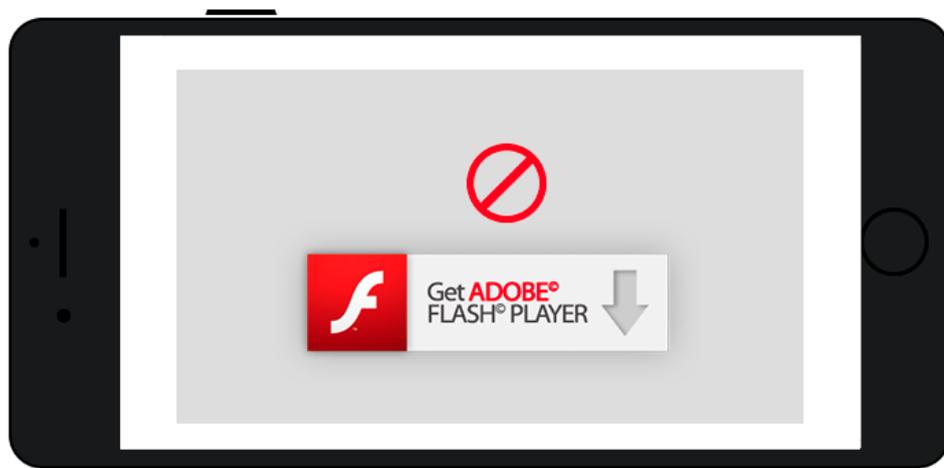
TINKOFF.RU

Анкета разбита на шаги, что облегчает заполнение. Всего 4 поля и крупная кнопка СТА для конвертации посетителя в лида.

Perezvoni.com

2. Номер телефона компании указан без +998 или в виде картинки.

3. Использование Flash на сайте или контент, который нельзя воспроизвести на мобильном. Желательно для мобильной версии полностью исключить такой контент. Он не будет нормально работать и вызовет много трудностей у посетителей. Видео не пойдет, игра не заиграет, а особая анимация в меню разьет мозг. Яблочные устройства ios вообще не поддерживают Flash анимацию. Удаляйте лишнее, зачем усложнять просмотр.



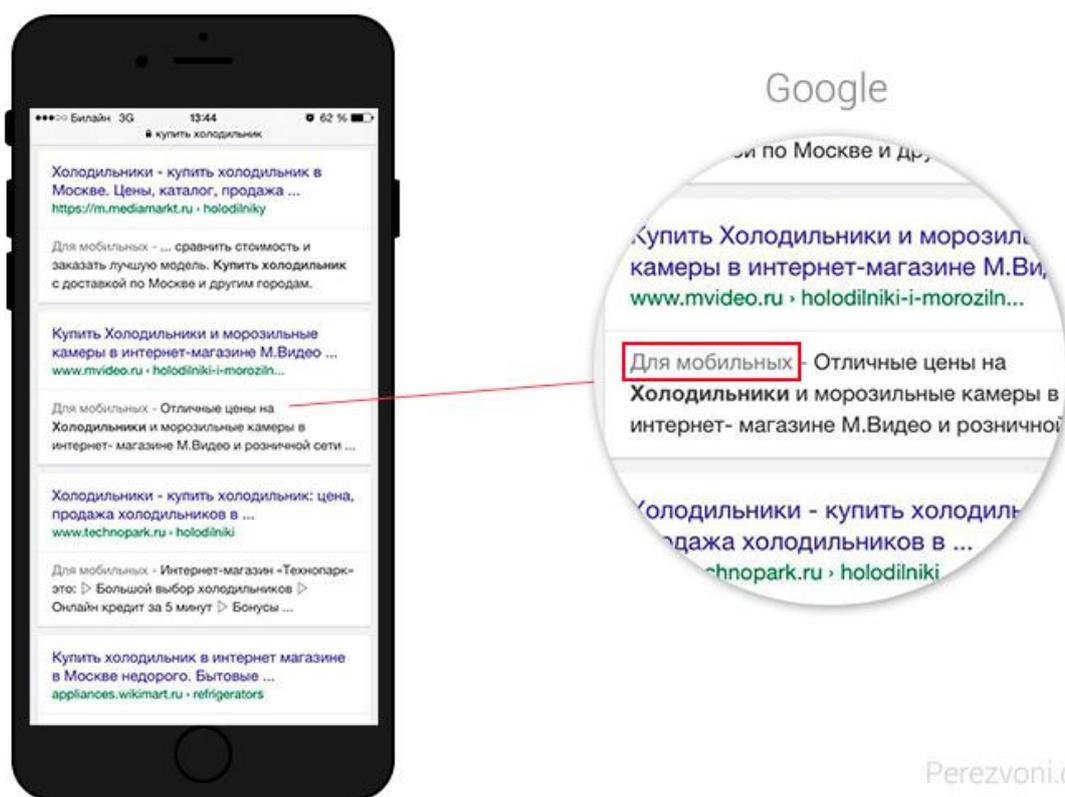
#3. Контент, который нельзя воспроизвести

4. Виджеты и поп-апы на сайте, которые не адаптированы под мобильные. Онлайн-консультанты, виджеты обратного звонка, соц. шаринги, e-mail сборщики — эти инструменты помогут увеличить конверсию сайта, но если они не адаптированы под мобильный трафик, грузят сайт, перекрывают контент, то скорее всего, принесут больше вреда, чем пользы. Поэтому, обязательно, проверяйте ваш сайт на мобильных устройствах после того, как установите сторонний сервис.

5. Отсутствие адаптированной мобильной версии. Проблема кроссбраузерности и кроссплатформенности.

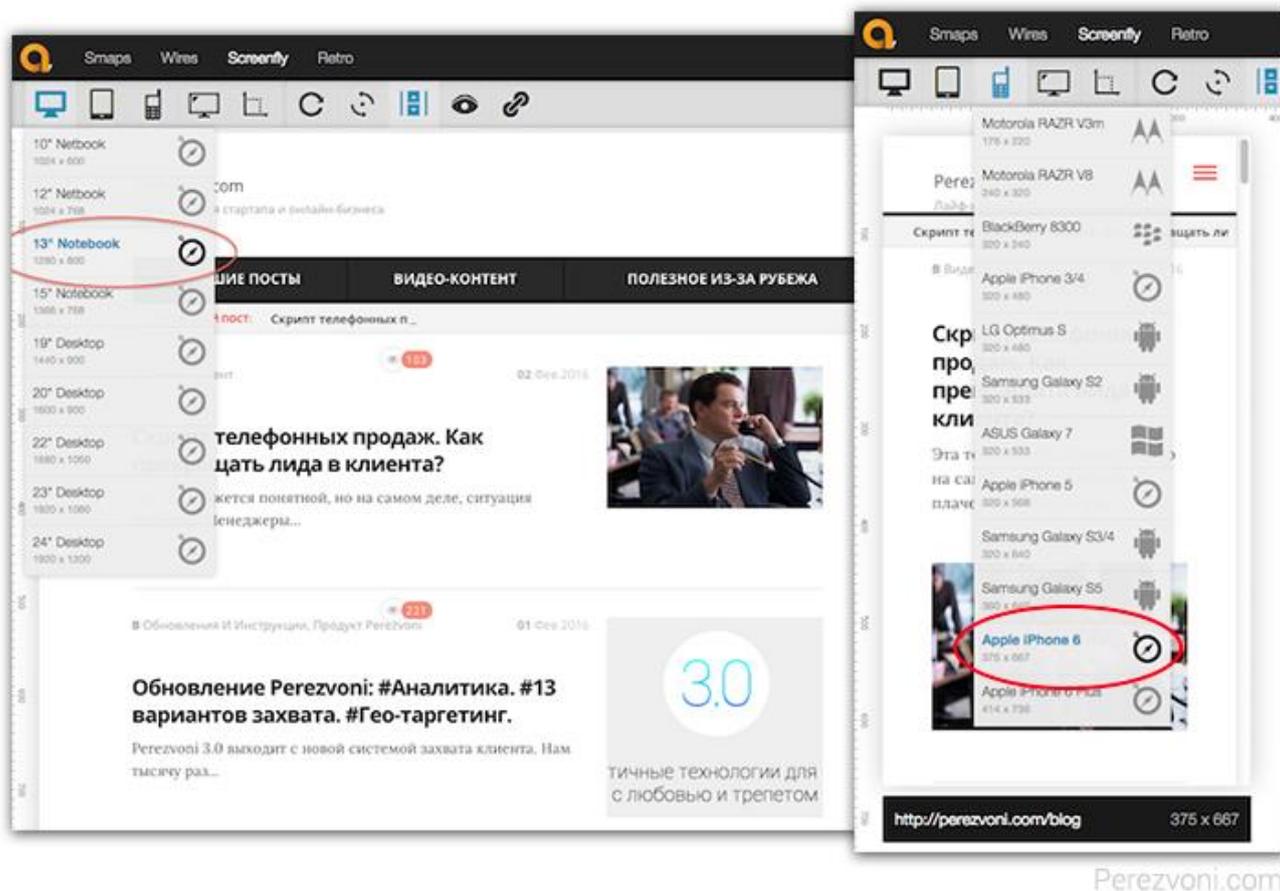
Какие плюсы от наличия мобильной версии?

Во-первых, поисковые системы отдают предпочтения в выдаче. Так google помечает сайты отметкой «Mobile friendly»:

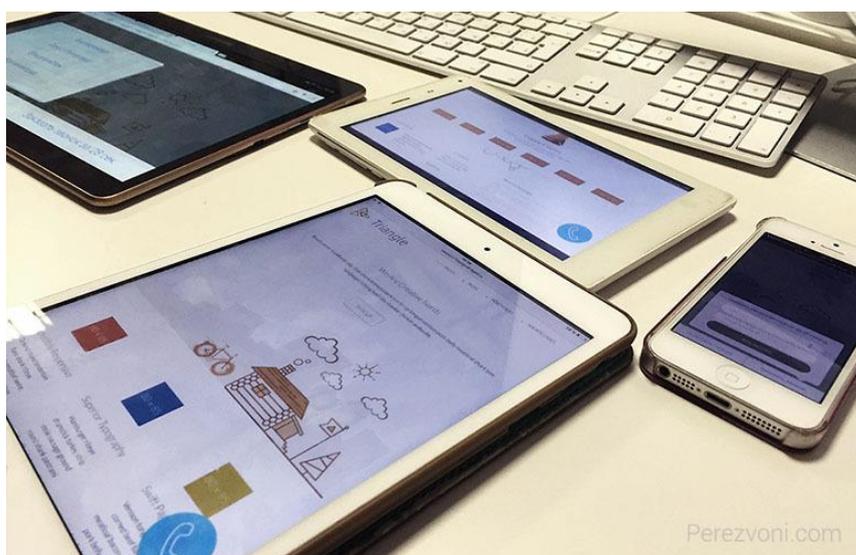


Во-вторых, пользователям удобно просматривать ваш сайт с разных устройств. Для этого, мобильная версия обязана работать исправно на всех типах мобильных устройств, включая мобильные телефоны и планшеты с разными диагоналями экранов. Независимо от платформы, производителя или браузера. Да, это очевидно, но всё же это важнейший критерий.

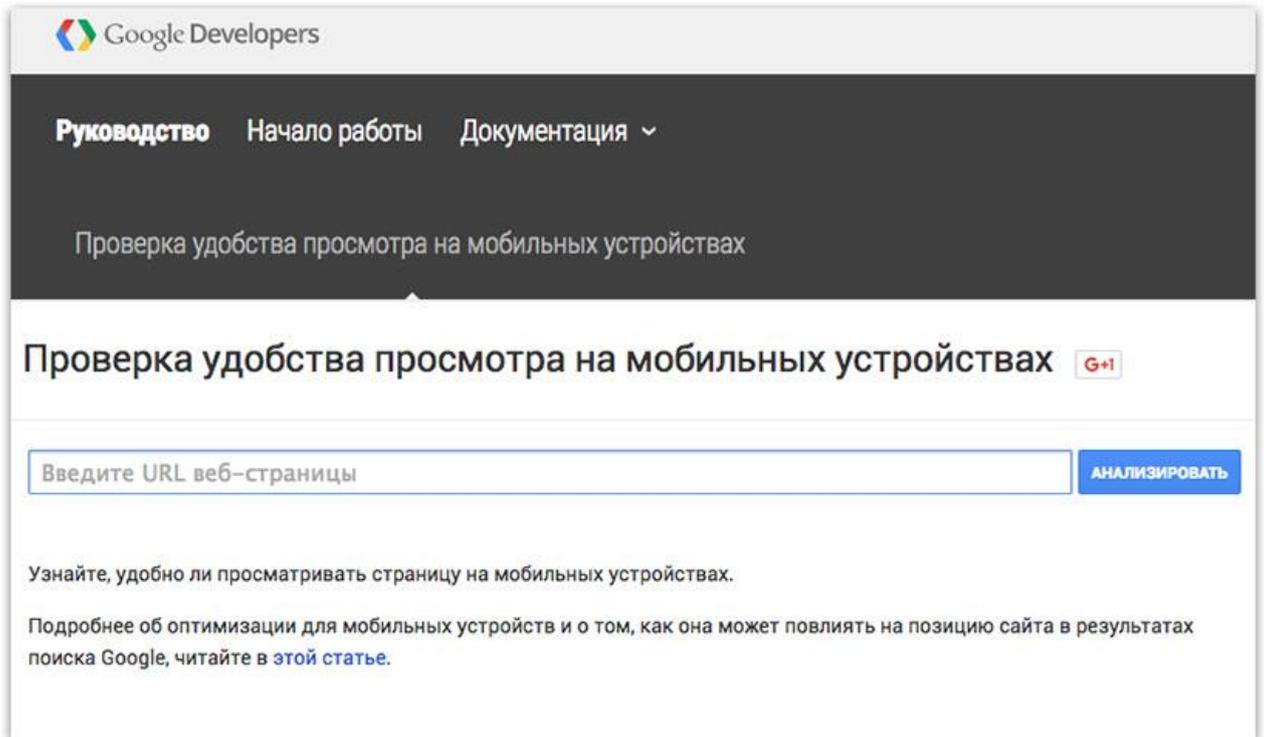
Есть хороший сервис quirktools.com для проверки сайта на разных устройствах. Например, для просмотра на маленьких и больших экранах. Заметьте, что это не модуляция мобильного телефона, а просто подгонка под размер его экрана.



Конечно, чтобы точно убедиться в правильности, стоит проверить это в живую на настоящих устройствах:



Проверить мобильную версию сайта на удобство и скорость загрузки можно через [Google сервис](#). Рекомендации по скорости можно соблюдать до оценки 90, дальше они теряют актуальность.



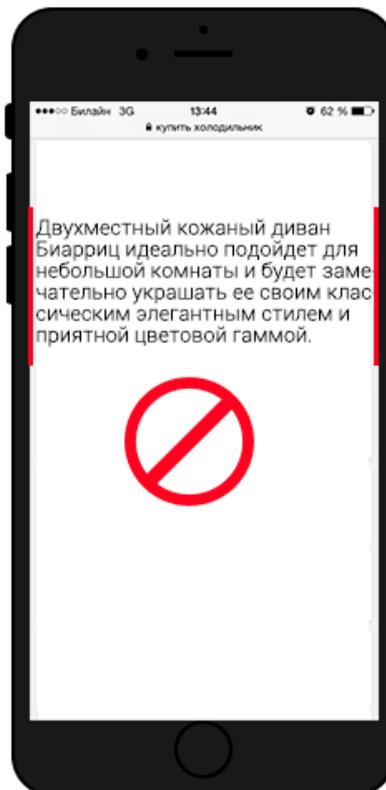
6. Слишком мелкий шрифт и отсутствие отступов.

Важный текст не должен быть менее 16px и убедитесь, что межстрочный интервал не превращает текст в единую кашу.



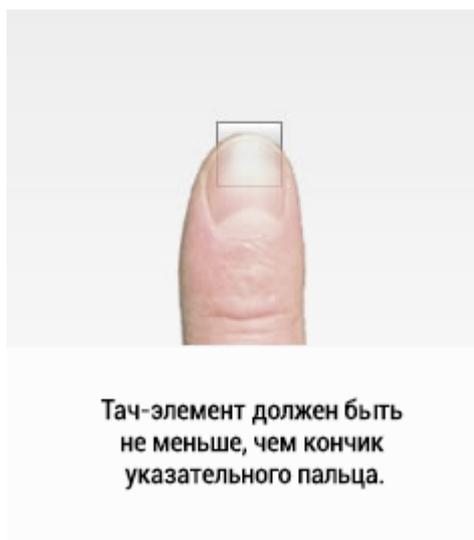
Не забудьте про отступы. Минимум 15px от края экрана. Часто контент прилипает к самому краю экрана, что портит внешний вид:

#6. Отсутствие отступа от края



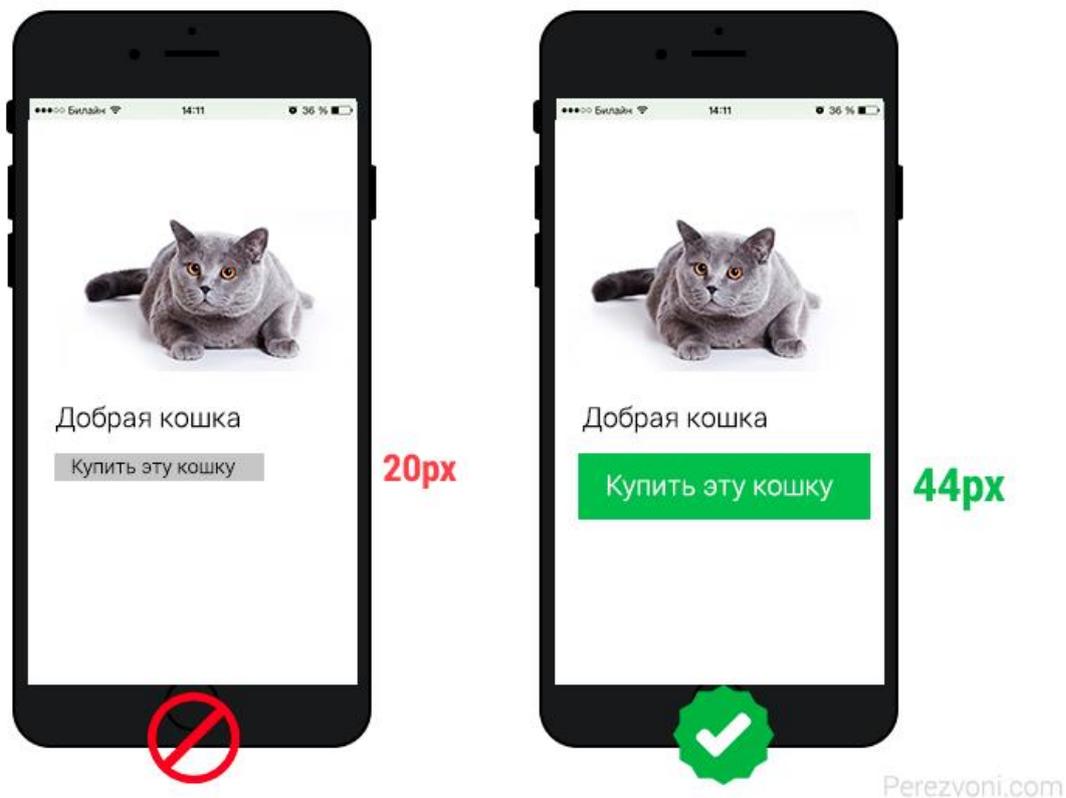
7. Кнопки и ссылки, которые не дружат с тач-скрином.

Кнопка «купить» или «оставить заявку» должна быть не только заметной, но и удобной для тач-касания пальцем.



Хотя пальцы у всех разные, есть рекомендации от производителей. Например, компания Apple призывает по [iPhone Human Interface Guidelines](#) использовать размер тач-элемента минимум 44px на 44px. Microsoft гайдлайн рекомендует размер

34рх, а минимально допустимый 26рх. Полезно почитать статью [Идеальный дизайн для тач-скринов](#). Правда, статья на английском языке, но смысл понять можно.



8. Долгая загрузка сайта.

Еще один пункт, который может понизить конверсию сайта. Проблема скорости характерна для всех устройств, но наиболее актуальной она становится для мобильных пользователей. По важности скорость для них идет сразу после понятного интерфейса: если пользователь смог разобраться, какую кнопку ему нажать для совершения покупки, то следующим подводным камнем будет время ожидания этой самой кнопки на экране телефона.

Для справки: 1 секунда задержки «стоит» порядка 3-7% конверсии сайта (10 секунд «стоит» 20-40%). Если сайт загружается 15 секунд и более, то ускорение до 3-5 секунд повысит конверсии на те самые 20-40 пунктов. Проверить скорость сайта можно при помощи сервисов: [Айри.рф](#) или [WebPageTest](#) или [Pingdom](#)

Pingdom Website Speed Test
Enter a URL to test the load time of that page, analyze it and find bottlenecks

perezvoni.com **Test Now**

Settings ▾

perezvoni.com
Tested from Stockholm, Sweden on March 23 at 19:48:30

Perf. grade	Requests	Load time	Page size
79/100	61	2.56s	1.9MB

Your website is faster than 62% of all tested websites

[DOWNLOAD HAR](#) [Tweet](#) [Post to Timeline](#) [Email](#)

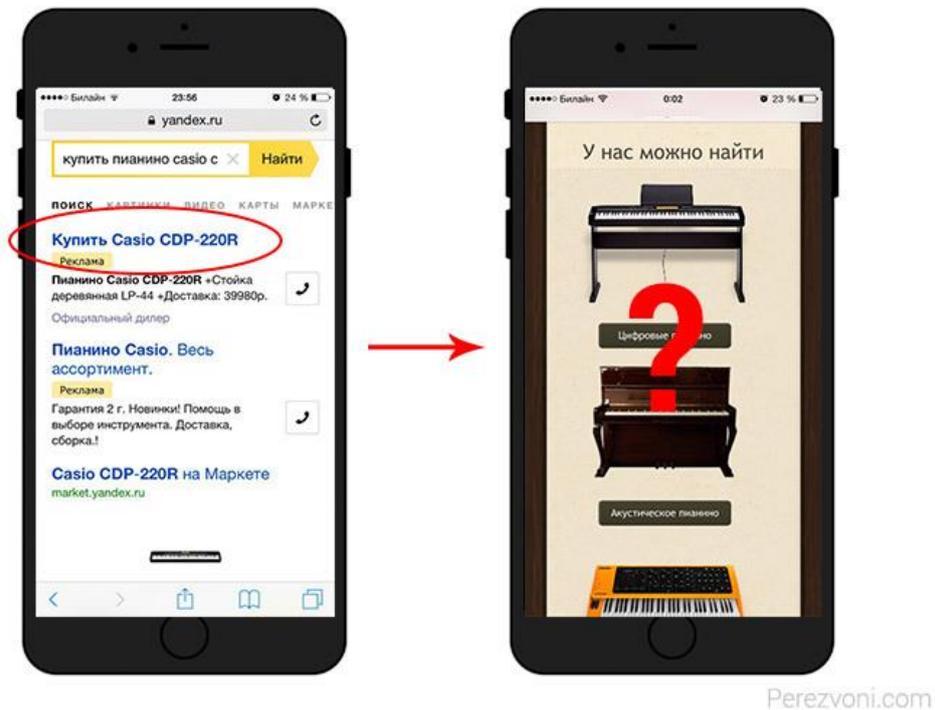
Существуют 3 ключевые проблемы скорости сайта — это размер, расстояние и виджеты. Обычно, для мобильного пользователя не требуются какие-то специальные эффекты на сайте, подойдет обычная верстка сайта. Дополнительно экран мобильного телефона, чаще всего, обладает низким разрешением, поэтому для мобильных пользователей можно «отгружать» меньшие по размеру изображения (и использовать более прогрессивный формат, например, WebP).

Для решения проблемы расстояния нужно убедиться, что хостинг сайта нормально доступен из сетей основных мобильных операторов. Оптимальным будет размещение в хорошем московском (или региональным — для региональных проектов) Датацентре, который заявляет хорошую связность с сетями мобильных операторов. Зарубежный хостинг мало подходит (и имеет ряд проблем со скоростью передачи данных из-за границы).

9. *Ошибочный редирект.* Отсутствие перехода на полную версию сайта.

Естественно, что все мобильные пользователи должны попадать на мобильную версию сайта, но случаются типичные ошибки в редиректе. Например, как бывает?

Пользователь ищет конкретный товар в Яндексе и переходит к вам. Система определяет заход с мобильного телефона и перенаправляет на главную страницу мобильной версии, а не нужную клиенту. Приходится заново искать этот товар на сайте или попробовать уйти к конкурентам.



Дополнение: Не всегда мобильной версии достаточно, возможно, вы скрыли нужные клиенту функции, ссылки или вообще, клиент привык к desktop-варианту. Всегда оставляйте возможность вернуться к полной версии сайта.

10. Автозаполнение. Включение набора цифр.

Конверсия растет, если мы уменьшаем сопротивление для клиента. Каждое поле можно сделать более удобным для заполнения. Для этого прописываем соответственно:

Поле E-mail – `input type=«email»`

Поле Телефон – `input type=«tel»`

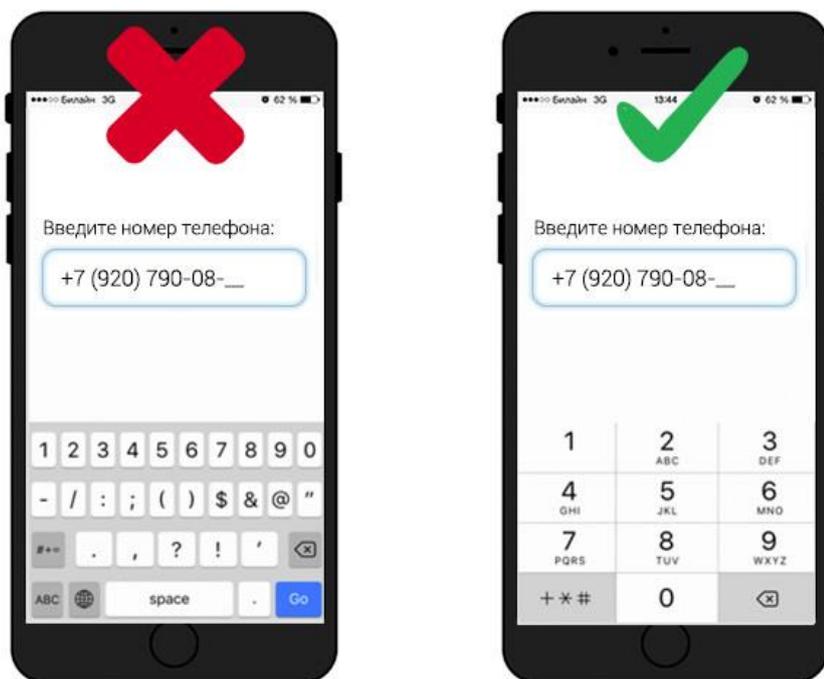
Поле Адрес – `input type=«text»`

Поле Индекс, номер карты – `input type=«text» pattern=«\d*»`

Поле Дата – `input type=«date»`

Поле Кнопка «Отправить» – `input type=«submit»`

Более подробное руководство с наглядным примером можно посмотреть на этом сайте: <http://ionicframework.com/>



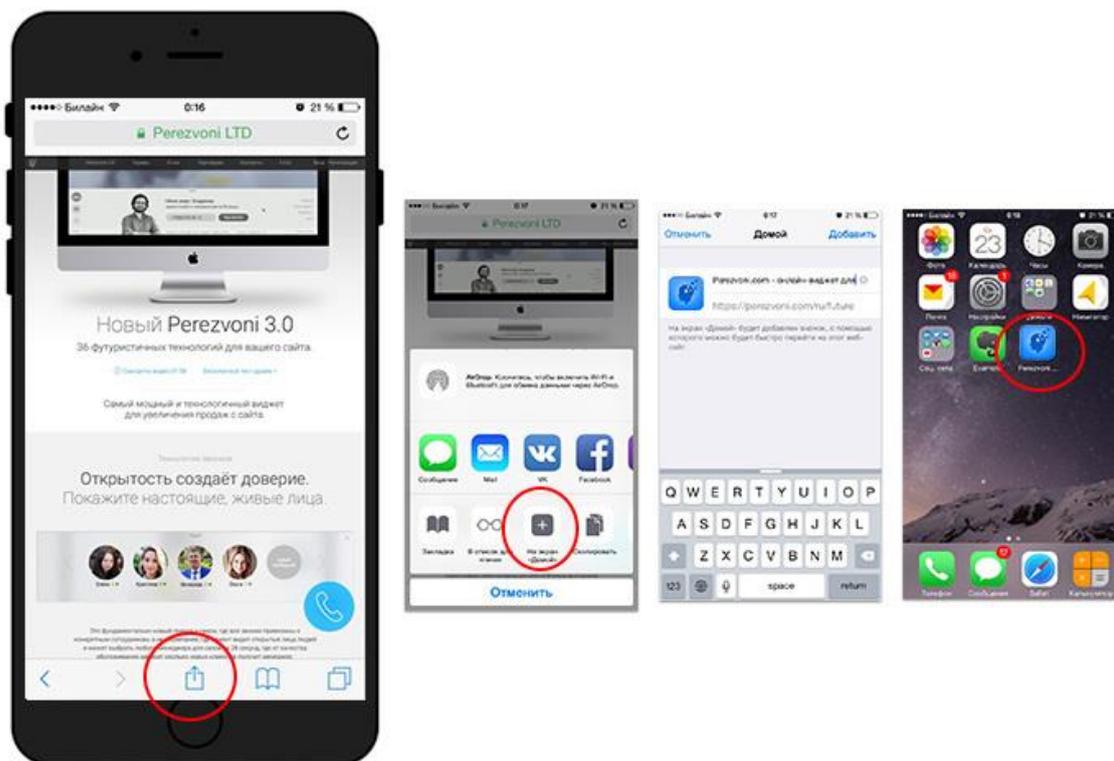
Perezvoni.com

Дополнение для удобства

Мало, кто знает, что можно добавить закладку сайта на рабочий стол для быстрого доступа. Иконка выглядит как приложение и будет закреплена на главном экране пользователя.

Как это сделать:

Сделайте иконку размером 180px на 180px (без закругления), а на сайте добавьте тег `<link rel=«apple-touch-icon» href=«иконка.png»`.



Perezvoni.com

На данный момент в стране динамично развивается национальная социальная сеть Davra.uz, авторизация происходит через ID.UZ — Единую систему идентификации и аутентификации для сайтов и информационных систем Национального сегмента сети интернет.

ID.UZ выполняет функцию единого центра идентификации, открывающего доступ к таким проектам как:

- my.gov.uz
- Lex.uz
- Национальная библиотека им. Алишера Навои
- WWW.UZ
- Ziyonet.uz
- Utube.uz
- Fikr.uz
- Desk.uz
- cctld.uz
- Gazeta.uz
- Любым сайтам, поддерживающим протокол OpenID

Если вы зарегистрировались однажды на ID.UZ, то вам больше не надо регистрироваться на других сайтах, поддерживающих технологию OpenID, достаточно указать свой логин на ID.UZ. Такая возможность имеет много преимуществ: не нужно больше запоминать много паролей и каждый раз заново вводить данные о себе на многих сайтах.

Зарегистрированный пользователь ID.UZ создаёт персональный профиль со своей фотографией, в который вносит свои личные данные. Желательно заполнять

все имеющиеся поля, правильно указывать все запрашиваемые данные, по которым вас смогут найти другие пользователи сайта.

Участники сети ID.UZ могут иметь 2 статуса:

- Анонимный пользователь;
- пользователь PassportID.

Внешние сервисы, использующие авторизацию через ID.UZ, могут давать разные права в зависимости от этого статуса.

Статус PassportID присваивается администрацией после предъявления пользователем своих паспортных данных. Этот статус даст возможность доступа к государственным, коммунальным, банковским и прочим услугам, требующим идентификации пользователя.

ИНСТРУКЦИЯ ПО ИНТЕГРАЦИИ OPENID (ID.UZ) С САЙТОМ

OpenID — это открытая децентрализованная система, которая позволяет пользователю использовать единую учётную запись для авторизацию на множестве не связанных друг с другом сайтов, порталов, блогов и форумов.

Существует несколько OpenID-провайдеров, посредством которых совершается аутентификация пользователей. Одним из них является проект ID.UZ.

Для пользователя это означает, что ему теперь нет необходимости регистрироваться на каждом сайте, достаточно зарегистрироваться у OpenID-провайдера (например, на ID.UZ), и с этой учетной записью он сможет проходить авторизацию на любом сайте, котором установлена поддержка OpenID.

Для владельца сайта это возможность, во-первых, быстро и без существенных трудностей сделать на своем сайте авторизацию пользователей; во-вторых, упростить пользователям вашего сайта процесс регистрации, а вернее вообще исключить его тем, кто имеет учетную запись на ID.UZ.

Если ваш сайт работает на популярной CMS, то процесс интеграции существенно упрощен. Почти на все популярные движки имеются плагины или модули для поддержки OpenID. Например, в Drupal поддержка OpenID встроена по умолчанию (<http://drupal.org/documentation/modules/openid>), на WordPress имеется плагин (<http://wordpress.org/extend/plugins/openid/>), а для Joomla – расширение (<http://extensions.joomla.org/extensions/access-a-security/site-access/authenticationbridges/19189>).

Если у вас эксклюзивный движок, то для того, чтобы установить у себя поддержку OpenID и дать возможность пользователям входить на ваш сайт с логином и паролем, зарегистрированным на ID.UZ, необходимо встроить в движок вашего сайта библиотеку OpenID.

Инструкция по внедрению OpenID аутентификации на PHP

Скачайте библиотеку OpenID с нашего сайта. В архиве также имеются скрипты с примером аутентификации. Содержимое архива:

Auth/_____	Библиотека для работы с протоколом openid (не требует модификации).
detect.php__	Утилита проверки необходимых компонентов для работы библиотеки
common.php	Необходимый функционал.
index.php__	Форма авторизации.
try_auth.php	Обработчик формы авторизации, пересылает данные на OpenID сервер.
finish_auth.php	Обработчик результата.

Распакуйте архив и вставьте содержимое в любую директорию на веб-сервере, например, openid. Введите в браузере URL к файлу detect.php, например <http://site.uz/openid/detect.php>. После непродолжительной проверки, утилита выдаст информацию о поддержке вашим сервером библиотеки OpenID. Если в полученной информации имеются критические замечания, исправьте их или установите дополнительные компоненты.

Откройте в браузере скрипт index.php (например, <http://site.uz/openid/index.php>). На странице появится пример формы входа на сайт. Введите в поле ваш логин на ID.UZ и нажмите —Войти—. Скрипт try_auth.php обработает введенные данные, проверит существование логина и переместит вас на сайт ID.UZ, где вы должны ввести свой пароль. Далее необходимо подтвердить передачу учетной информации с ID.UZ на ваш сайт. После согласия, вы вернетесь на свой сайт, где откроется скрипт finish_auth.php. В нем, производится обработка результата, полученного от ID.UZ (отмена аутентификации, ошибка пароля, успешный вход). В случае успешной аутентификации, на странице отобразятся данные пользователя, полученные от ID.UZ.

Если все прошло успешно, на основе файлов с примером (common.php, index.php, try_auth.php, finish_auth.php) встройте интеграции к себе в движок.

Скопируйте html форму с файла index.php в страницы на вашем сайте, где вы хотели бы видеть форму авторизации:

```
<form method="get"
action="/openid/try_auth.php">
  Ваш логин на ID.UZ: >
  <input type="hidden" name="action"
value="verify" /
  <input type="text" name="iduz_login" value=""
/>
.id.uz
```

```
<input type="submit" value="Войти" />
</form>
```

Если вы загрузили файлы не в директорию /openid/, тогда изменить атрибут action в тэге form.

В файле try_auth.php в строке #35 методу Auth_OpenID_SRegRequest::build передаются два массива:

```
$sreg_request =
Auth_OpenID_SRegRequest::build(
    // Required
    array('nickname'),
    // Optional
    array('fullname', 'email'));
```

Первый содержит обязательные поля, которые должны будут переданы с ID.UZ на ваш сайт, второй – с необязательными. Всего доступно 9 полей:

- fullname – Полное имя
- nickname – Псевдоним
- dob – Дата рождения в формате уууу-mm-dd
- email – E-mail адрес
- gender – Пол (M, F)
- postcode – Почтовый индекс
- country - Страна
- language – Предпочитаемый язык
- timezone – Часовой пояс

Когда пользователь с вашего сайта через форму попытается авторизоваться, его перекинет на ID.UZ, где он введет свой пароль и подтвердит передачу информацию. Затем его перекинет на скрипт finish_auth.php. Добавьте в этом обработчике необходимый вам код для сохранения сессии и/или занесения пользовательских данных в вашу БД (после строки #35).

Если вы хотите использовать свой обработчик, созданный на основе finish_auth.php, в файле common.php, в функции getReturnTo измените адрес страницы возврата (обработчик результата) в строке #105 на адрес своего обработчика.

Интеграция функционала «Поделиться» на Davra.uz

Скопируйте ссылку и разместите её у себя на сайте. И тогда ваши посетители смогут делиться интересными ссылками с вашего сайта в социальной сети Davra.uz



```
<a href="#"  
onclick="window.open('http://davra.uz/api/apishare?link='+window.location.href, ",  
'width=761,height=500');"></a>
```



```
<a href="#"  
onclick="window.open('http://davra.uz/api/apishare?link='+window.location.href, ",  
'width=761,height=500');"></a>
```



```
<a href="#"  
onclick="window.open('http://davra.uz/api/apishare?link='+window.location.href, ",  
'width=761,height=500');"></a>
```