

Check-List SEO: для тестирования

1. Тестировщик первым делом приступает к проверки наличия требования из списка ниже
2. После того, как все требования удовлетворены - переходит к задаче в РМ
3. Тем самым у нас 3 уровня ограничений и дополнительная защита от выката багов в задаче

Авто-тестирование происходит через файл [тут ссылка на внутреннюю таблицу, которую парсит собственный бот и тестирует страницу автоматически].

Вводная

* - пункты, которые не нужно тестировать руками. (они покрыты автотестами)

Чек-лист с алгоритмами ручного тестирования SEO требований для выпускаемых сервисов/проектов.



Check-List что тестируем

Check-List SEO

1) Наличие и содержимое мета-тегов:

- title
- description
- keywords
- og:title
- og:description
- og:type*
- og:url
- og:image
- og:image:width*
- og:image:height*
- og:site_name*
- og:locale*
- canonical
- shortcut icon*
- amphtml

2) Наличие JSON*

4) Наличие и содержание заголовка h1

5) Требования по рендеру продукта и ответам сервера:

- честные ответы сервера
- отображение контента

6) Наличие правил в URL:

- www*
- закрывающий /*
- Верхний регистр в URL*

7) Подсказка по User-Agent поисковиков

1. Требования в <head>...</head>

1.1 Размещение мета-тегов. Данные мета-теги должны располагаться на странице сервиса/продукта всегда. Содержимое этих тегов брать с прикрепленной задачи SEO в РМ.

Мета-теги	Что должно быть на странице.
Title	title <title>Текст сервиса</title>
Description	description <meta name="description" content="Текст сервиса">

Keywords	keywords <meta name="keywords" content="ключевое слово сервиса, ключевое слово сервиса 2">
OG:Title	OG:Title <meta property="og:title" content="Текст продукта">
OG:Description	OG:Description <meta property="og:description" content="Текст продукта">
OG:type*	OG:Type <meta property="og:type" content="website">
OG:url	OG:url <meta property="og:url" content="Абсолютный URL сервиса">
OG:image	OG:image <meta property="og:image" content="https://www.mos.ru/open-graph/v1/frontend/ваш URL сервиса без https://www.mos.ru/">
OG:image:width*	OG:image:width <meta property="og:image:width" content="1200">
OG:image:height*	OG:image:height <meta property="og:image:height" content="630">
OG:site_name*	OG:site_name <meta property="og:site_name" content="mos.ru">
OG:locale*	OG:locale <meta property="og:locale" content="ru_RU">
Canonical	canonical <link rel="canonical" href="Абсолютный URL сервиса">
Shortcut icon*	Shortcut icon <link rel="shortcut icon" type="image/x-icon" href="/otvet-static/images/favicon/favicon.ico">
amphtml	amphtml <link rel="amphtml" href="https://www.mos.ru/amp/ваш URL сервиса">

1.2 Наличие точного соответствия JSON в head или в body*

Микроразметка Organization и WebSite

```
<script type="application/ld+json">{
  "@context": "http://schema.org",
```

```
"@type": "WebSite",
"name": "mos.ru",
"alternateName": "Сайт Москвы",
"url": "https://www.mos.ru"
}</script>

<script type="application/ld+json">{
  "@context": "http://schema.org",
  "@type": "Organization",
  "url": "https://www.mos.ru/",
  "logo": "https://www.mos.ru/front/markup/header-footer/img/logo.svg"
}</script>
```

2. Требования к <body> </body>

2.1 Наличие заголовка h1 на странице. Содержимое h1 брать с прикрепленной задачи РМ. Если заголовка два – требуется уточнение у SEO

3. Требования по рендеру продукта и ответам сервера

3.2 Честные ответы сервера

3.2.1 Если ввести некорректный URL сервиса/проекта отдается серверная ошибка 404 error? Если да - ок. (Например **url-services/12345** , **url-servicesdndkcsd** , **url-services12345** , **url-services/qwe** , **url-services/qwe34**)

3.2.2 Проверить 301 редиректы путем коррекции URL (убрать закрывающий / на конце). Если редирект **СЕРВЕРНЫЙ** - то ок. (Серверный ответ можно проверить с помощью плагина для Google Chrome HTTP Headers или Link Redirect Trace или Redirect Path)

3.2.3 Ответы сервера 200, 404, 301 для пользователей и поисковых систем совпадают?*

Алгоритм проверки:

Делаете 2 запроса с помощью curl или Postman.

Один под юзер агентом браузера (как обычный пользователь, например я) (curl -I -A "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/76.0.3809.132 Safari/537.36)" <https://www.mos.ru/>) , только вместо главной страницы вставляете URL сервиса

Другой под юзер агентом Yandex (как поисковый бот) (curl -I -A "Mozilla/5.0 (compatible; YandexBot/3.0; +<http://yandex.com/bots>)" <https://www.mos.ru/>) , только вместо главной страницы вставляете URL сервиса)

Смотрим ответы сервера, они одинаковые для пользователя и поисковика? Если разное содержимое - то Alert.

3.3 Отображение контента

3.3.1 iframe*

Данного тега не должно быть на странице сервиса/проекта.

3.3.2 Контент страницы и код страницы содержит все основные элементы? (Если тест успешный, переходим сразу к 3.3.4. Если alert, переходим к 3.3.3)

Алгоритм проверки:

Открыть две одинаковые страницы в браузере.

В одной открыть панель разработчика Inspect, в другой код страницы View Page Source

Сопоставить элементы страниц, имеется ли в коде первой то, что есть во второй?

Inspect vs View Page Source:

- Имеется ли h1?
- Имеются ли мета-теги title, description, keywords, og:title, og:description, og:image?
- Имеется ли описание(любой текст в сервисе) сервиса в обеих вкладках?
- Имеются ли фильтры, ссылки в обеих вкладках?

3.3.3 Сервис Puppeteer подключен? (Если 3.3.3 валидный, то пункт 3.3.4 пропускаем)

Puppeteer - это сервис, который рендерит страницы на JS, отрисовывает весь HTML и отдает поисковому боту.

- Проверяем наличие Puppeteer через команду CURL или Postman (curl -I -A "Mozilla/5.0 (compatible; YandexBot/3.0; +<http://yandex.com/bots>)" <https://www.mos.ru/>) , только вместо главной страницы вставляете URL сервиса. Если будет в заголовке header атрибут PUPPETEER: true - то тест пройден.

```

Last login: Fri Nov  8 12:29:22 on ttys000
[MacBook-Pro-Urij:~ aqva_spirt$ curl -I -A "Mozilla/5.0 (compatible; YandexBot/
ts)" https://www.mos.ru/clinics/
HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8
Content-Length: 409088
Connection: keep-alive
Server: nginx
Date: Fri, 08 Nov 2019 09:34:57 GMT
Vary: Accept-Encoding
Host: www.mos.ru
status: 200
cache-control: private, must-revalidate
pragma: no-cache
expires: -1
x-session-fingerprint: d74fc47c4093e73dbcccf154362f90
x-xss-protection: 1; mode=block
x-content-type-options: nosniff
ETag: W/"63e00-cjF5x51V0Mg5Ig7PZeBMD35qjFY"
PUPPETEER: true
X-Session-Fingerprint: f46879ad9215440c441490f15b3f1da8

MacBook-Pro-Urij:~ aqva_spirt$ █

```

3.3.4 Контент страницы одинаковый для ботов и пользователей?

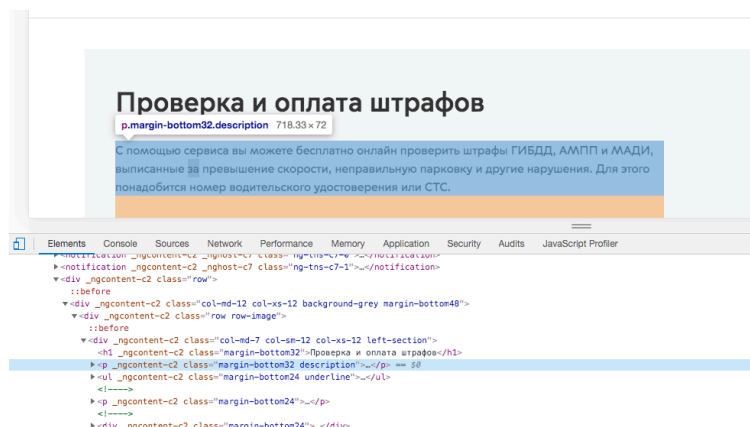
Алгоритм проверки:

Делаем запрос(curl or postman) под юзер агентом Yandex (как поисковый бот) (curl -i -A "Mozilla/5.0 (compatible; YandexBot/3.0; +<http://yandex.com/bots>)" <https://www.mos.ru/>) , только вместо главной страницы вставляете URL сервиса)

Сравниваем код в Inspect в браузере и код полученный в curl or postman. Он одинаковый для пользователя и поисковика?

Какие параметры сравнивать:

- В head (title, description, keywords, og:title, og:description, og:image)
- В body (h1, другие заголовки h если имеются, описание сервиса в блоках <p>, , <div>). Нет точного указания в каком теге будет текст, так как разработчики делают по разному.



Если разное содержимое - то Alert.

4. Формирование URL

4.1 WWW*

Любой URL сервиса должен содержать www.

Сервер должен отдавать 301 redirect если пользователь/поисковый бот попадет на URL **без** www.

4.2 URL со / на конце*

Любой URL сервиса должен заканчиваться на закрывающий /.

Сервер должен отдавать 301 redirect если пользователь/поисковый бот попадет на URL **без** закрывающего /.

4.3 URL с регистром*

URL не должен содержать верхний регистр.

Если подставить верхний регистр в URL - должен производиться 301 серверный редирект.

Пример:

<https://www.mos.ru/clinicS/> → 301 redirect → <https://www.mos.ru/clinics/>

4.4 URL с параметрами*

URL содержащие параметры на конце должны отдавать ответ сервера 200ок

Пример:

https://www.mos.ru/authority/person/33391093/?utm_source=search&utm_term=widget

Юзер-агенты поисковых ботов (вспомогательная информация)

Используйте для проверки Юзер-агенты ботов, чтобы смотреть на страницу так же как они. Обновленный user-agent для Google

<https://webmasters.googleblog.com/2019/10/updating-user-agent-of-googlebot.html>

User agent

YandexBot

Mozilla/5.0 (compatible; YandexBot/3.0; +http://yandex.com/bots) //
Основной бот

GoogleBot

Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html) //
Основной бот

Googlebot

Mozilla/5.0 (Linux; Android 6.0.1; Nexus 5X Build/MMB29P)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/41.0.2272.96 Mobile
Safari/537.36 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)
// Бот для смартфонов

Mail.Ru

Mozilla/5.0 (compatible; Linux x86_64; Mail.RU_Bot/2.0;
+//go.mail.ru/help/robots) // Основной бот
Mozilla/5.0 (compatible; Linux x86_64; Mail.RU_Bot/Fast/2.0;
+//go.mail.ru/help/robots) // Бот быстрого индекса