hiclick

DEVELOPER GUIDE

DEMO-SITE.RU · 12.05.2026

# Инструкция по созданию сайта demo-site.ru

Гайд для команды разработки. Как превратить семантическое ядро и структуру в работающий сайт.

| ВЕРСИЯ | ОБЪЁМ | АУДИТОРИЯ | ПОДГОТОВЛЕНО |
| --- | --- | --- | --- |
| 1.0 | 20 разделов | Tech-команда | hiclick · SEO |

# Что это и как пользоваться

Документ описывает, **как превратить собранное семантическое ядро в работающий сайт**. Это не ТЗ на разработку в полном смысле, а карта того, что архитектура сайта должна поддерживать «из коробки» — потому что иначе SEO-ядро не отстреливает.

## Как читать структуру сайта

Открой `site_structure.xlsx`, лист «3. Только посадочные». Каждая строка — будущая страница на сайте. Ключевые колонки:

- `target_url` — какой URL должен генерироваться (это **обязательный** контракт между разработкой и SEO).
- `parent_url` — родитель в иерархии (для хлебных крошек и навигации).
- `page_type` — какой шаблон применяется (catalog/subcatalog/solution/brand/product_card/commercial_landing/info_article/glossary).
- `h1`, `title`, `description` — мета-данные страницы (берутся из `content_briefs.xlsx`).
- `priority` — какой волной запускать (1 — критично к запуску, 2 — второй волной, 3 — длинный хвост).

# 01 Рекомендуемый технический стек

## CMS

**Не рекомендуем** делать на Tilda / готовом конструкторе магазинов: 231 кастомных URL с разными шаблонами и фасетной навигацией такие решения тянут плохо, плюс ограничения на schema.org и canonical-теги.

**Рекомендуемые варианты:**

| Стек | Когда подходит | Плюсы | Минусы |
|---|---|---|---|
| **1С-Битрикс «Малый бизнес»** или выше | Если уже есть учёт в 1С, нужна интеграция | Готовые модули каталога/корзины/YML, документация на русском | Тяжёлый, сложно делать кастомный фронт |
| **OpenCart / WooCommerce + кастомизация** | Бюджетный B2C-каталог без сложной B2B-логики | Дёшево, много модулей | Производительность на 5K+ SKU слабая |
| **Headless + Next.js / Nuxt** (Strapi/Directus + React/Vue) | Если хочется быстрый сайт с SSR + независимый бэк-офис | Идеальный Core Web Vitals, гибкая структура URL | Нужна команда фронтенда |
| **Самописная (Laravel/Django/Node)** | Если есть техкоманда и нужна максимальная гибкость | Полный контроль | Дольше и дороже |

**По собственному ядру (231 URL + потенциально 5-10 тыс. SKU)** наиболее сбалансированный вариант — **1С-Битрикс** (если уже есть учёт) или **Next.js + headless CMS** (если строите с нуля и важна скорость).

## Обязательные технические требования (вне зависимости от стека)

- **SSR или SSG** для всех публичных страниц (не SPA с client-side рендером — это убивает SEO).
- **PHP 8.2+** / **Node 20+** / **Python 3.11+** (в зависимости от стека).
- **PostgreSQL 14+** или **MySQL 8+** для каталога.

- **Redis** для кеша.
- **CDN** для статики (фото товаров, CSS/JS) — Yandex Cloud CDN / Cloudflare / Selectel CDN.
- **Поиск** — Elasticsearch / Meilisearch (для большого каталога). Для MVP можно ILIKE по PostgreSQL.

---

# 02 Структура URL — обязательные правила

## Общие принципы

✅ Правильно:
/catalog/seyfy/
/catalog/stellazhi/metallicheskie/
/brands/valberg/seyfy/
/resheniya/magazin-odezhdy/
/goroda/barnaul/
/blog/kak-vybrat-seyf/
/spravochnik/seyfy-vidy/

❌ Неправильно:
/catalog/sejfy
/catalog/stellazhi/Metallicheskie/
/catalog/seyfy/?brand=valberg
/resheniya/magazin_odezhdy
/catalog/seyfy/barnaul/
/blog?article=12
/article/567

## Жёсткие правила

- **Только lowercase**: `/catalog/seyfy/`, не `/Catalog/Seyfy/`. Сервер должен 301-редиректить uppercase → lowercase.
- **Дефис как разделитель слов**: `magazin-odezhdy`, не `magazin_odezhdy` и не `magazinOdezhdy`.
- **Транслит ГОСТ**: «й» → `y`, «х» → `kh`, «ш» → `sh`, «щ» → `shch`, «ы» → `y`, «ь/ъ» опускаются. Файл `site_structure.xlsx` уже содержит готовые slug-и в `target_url`.
- **Trailing slash обязательно**: все URL заканчиваются на `/`. Если пользователь зашёл без слэша — 301-редирект на версию со слэшем.
- **Никаких параметров в основных URL**: `?page=2`, `?sort=cheap`, `?utm=*` допустимы только как технические параметры с canonical на базовый URL.
- **Глубина — максимум 3 уровня** для каталога (`/catalog/{cat}/{subcat}/`), 3 для solution (`/resheniya/{industry}/{category}/`), 3 для бренда (`/brands/{brand}/{category}/`). Глубже не идём — это требование к фильтрам в фасетной навигации (см. ниже).

## Правила формирования slug

В админке для каждой сущности (категория, подкатегория, бренд, отрасль, статья) должно быть **редактируемое поле slug**. При создании новой сущности — slug автогенерируется из названия (транслит + дефис), но редактор может изменить вручную.

**Уникальность slug-а** — в пределах родителя: - `metallicheskie` в `/catalog/stellazhi/metallicheskie/` и `/catalog/vitriny/metallicheskie/` — это нормально (разные родители). - Двух категорий с slug `seyfy` на уровне `/catalog/` быть не может.

## Редиректы

При изменении slug-а в админке — **автоматически создаётся 301-редирект** со старого URL на новый. Это критично, иначе теряем ссылочный вес и трафик.

Таблица редиректов хранится в БД, обрабатывается на уровне middleware/роутера до отдачи 404.

# 03 Модель данных — основные сущности

Минимальный набор таблиц/моделей, который должен поддерживаться:

## Сущности

```
┌──────────────────────────────────────────────────────┐
│ Category (Категория каталога)                        │
│ ├── id, slug, name, parent_id (для иерархии)         │
│ ├── h1, title, description, intro_text, body_text    │
│ ├── canonical_url, robots, schema_type               │
│ ├── priority (1/2/3), publish_status (draft/published/archived)  │
│ ├── sort_order, level                                │
│ ├── image_main, image_alt                            │
│ └── seo_lsi_keywords, faq_block_id (FK)              │
└──────────────────────────────────────────────────────┘
        ↓ has many
┌──────────────────────────────────────────────────────┐
│ Product (Товар/SKU)                                  │
│ ├── id, sku, slug, name, short_description, full_description     │
│ ├── price, price_b2b, currency, in_stock_qty         │
│ ├── brand_id (FK), main_category_id (FK)             │
│ ├── images[], video_url, certificates[]              │
│ ├── attributes (JSON: {material, width, height, weight, ...})    │
│ ├── h1, title, description (для product_card-страниц)│
│ ├── publish_status, created_at, updated_at           │
│ └── related_products[], cross_sell[]                 │
└──────────────────────────────────────────────────────┘

┌──────────────────────────────────────────────────────┐
│ Brand                                                │
│ ├── id, slug, name, logo, description, country       │
│ ├── h1, title, description, body_text                │
│ └── certificates[], website_url                      │
└──────────────────────────────────────────────────────┘

┌──────────────────────────────────────────────────────┐
│ Industry (Отрасль / solution)                        │
│ ├── id, slug, name                                   │
│ ├── h1, title, description, body_text, calculator_form_id        │
│ └── case_studies[] (кейсы реализованных проектов)    │
└──────────────────────────────────────────────────────┘

┌──────────────────────────────────────────────────────┐
│ City (Город / commercial_landing)                    │
│ ├── id, slug, name, region, latitude, longitude      │
│ ├── h1, title, description, body_text                │
│ ├── delivery_terms, manager_phone, manager_email     │
│ ├── warehouse_address, opening_hours, pickup_available           │
│ └── is_priority_geo (boolean — для Барнаула/НСК/Кемерово true)   │
└──────────────────────────────────────────────────────┘

┌──────────────────────────────────────────────────────┐
│ Article (info_article + glossary)                    │
│ ├── id, slug, type (info/glossary), title, h1        │
```

```
│ ├── description, content_html, faq_block_id                    │
│ ├── author_id (FK на Person — критично для E-E-A-T)            │
│ ├── published_at, updated_at, reading_time                     │
│ └── related_products[], related_categories[]                   │
└────────────────────────────────────────────────────────────────┘


┌────────────────────────────────────────────────────────────────┐
│ FAQBlock                                                       │
│ ├── id, page_id, page_type                                     │
│ └── items[]: [{question, answer, order}, ...]                  │
└────────────────────────────────────────────────────────────────┘


┌────────────────────────────────────────────────────────────────┐
│ Redirect                                                       │
│ ├── id, from_url, to_url, code (301/302), created_at           │
│ └── (для автоматических редиректов при смене slug)             │
└────────────────────────────────────────────────────────────────┘
```

## Ключевые связи

**Многие-ко-многим:**

- `Product ↔ Category` — товар может быть в нескольких категориях (главная + дополнительные). Поле `main_category_id` определяет каноническую категорию (для хлебных крошек, canonical URL карточки).
- `Product ↔ Industry` — товар может быть рекомендован для нескольких отраслей. Используется для генерации блока «состав комплекта» на solution-страницах.
- `Category ↔ Brand` — какие бренды представлены в категории. Используется для блока «производители и бренды» в каталоге.
- `Article ↔ Category/Product/Industry` — статьи могут ссылаться на разные сущности. Используется для блока «к этой статье подходят».

## Атрибуты товаров (критично для фильтров!)

В семантическом ядре много запросов вида «стеллажи металлические перфорированные с подсветкой 2000 мм». Чтобы такие запросы попадали на правильные подкатегории/фильтры, у товара должны быть **структурированные атрибуты**, а не плоский текст в описании.

Минимальный набор атрибутов (для категории «Стеллажи»):