ХОДИМА

ИНСТРУКЦИЯ

Виджет бронирования

Как подключить виджет бронирования на сайт оператора и настроить его

Версия 2026.11 · ноябрь 2026



Содержание

Что это и как выглядит 1
Подключение на сайт 2
SEO-страницы 11
PHP-модуль 16
Параметры 25
Оформление 30
Проверка и типовые проблемы 34

Что это и как выглядит

Виджет бронирования на сайте оператора: вид и возможности.

Виджет показывает каталог экскурсий, страницу экскурсии, выбор даты, времени и билетов, оформление, оплату и готовые билеты. Он живёт на сайте оператора и работает на его покупателей, а данные, описания и расписание берёт из «Ходимы». Покупатель не уходит на чужой домен и не заводит второй аккаунт.

Виджет встраивается двумя способами:

- на одну страницу сайта — ручным вызовом `window.mountHodimaWidget`;
- разделом «Экскурсии» — PHP-модулем, который отдаёт серверные страницы каталога и экскурсий для поисковика.

Разделы документа

Раздел	О чём
Подключение на сайт	Требования хостинга, установка ZIP, `config.php`, режимы embedded, standalone и hybrid, обновление модуля.
SEO-страницы	Что индексируется, адреса и canonical, метаданные, Open Graph и JSON-LD, sitemap раздела.
PHP-модуль	Состав архива, функции для CMS, все ключи конфига, кэш, безопасность и примеры для WordPress, «1С-Битрикс», Laravel и Twig.
Параметры	`mountHodimaWidget`: организация и API, скрытые блоки, способ оплаты, основа и цвета.
Оформление	Светлая и тёмная основа, пять цветов, шрифты, читаемость и подпись платформы.
Проверка и типовые проблемы	Как проверить установку и что делать при типовых ошибках.

Что видит покупатель на экране каталога, описано в документе «Обзор возможностей», раздел «Каналы продаж». Настройки оформления менеджер задаёт в админ-панели: документ «Руководство пользователя», раздел «Настройки организации».

Подключение на сайт

Как поставить виджет на сайт оператора: требования хостинга, установка PHP-модуля и три режима его работы.

Виджет подключается двумя путями. Первый — вручную вызвать `window.mountHodimaWidget` на одной странице сайта: так работает сценарий «у оператора есть свои описания экскурсий, нужен только виджет для продажи». Второй — поставить PHP-модуль «Ходимы»: он отдаёт серверные страницы каталога и экскурсий, которые индексирует поисковик. PHP-модуль не обязателен: без него виджет работает как SPA, а серверные страницы можно собрать и другим серверным клиентом рендерера. Параметры ручного вызова — Параметры.

Без PHP: только JavaScript

Если у оператора уже есть страница сайта и нужна только продажа, PHP-модуль не нужен. Виджет подключается обычными тегами `script` и `link`, как любой сторонний виджет.

```
<link rel="stylesheet" href="https://widget.<домен>/<версия>/hodima-widget.css">
<div id="hodima-widget"></div>
<script src="https://widget.<домен>/<версия>/hodima-widget.iife.js"></script>
<script>
  window.mountHodimaWidget({
    target: '#hodima-widget',
    tenantId: '<идентификатор организации>',
    apiBaseUrl: 'https://api.<домен>',
  })
</script>
```

Скрипт сам виджет не монтирует: он только регистрирует `window.mountHodimaWidget`. Монтирование — явный вызов, показанный выше. `tenantId` и `apiBaseUrl` можно не указывать, если они заданы при сборке виджета; для нескольких организаций их передают явно. Файлы `hodima-widget-icons.iife.js` и каталог `brand/` не обязательны: без них виджет работает, но не покажет иконки удобств и иллюстрацию шапки.

Что получает оператор:

- работающий SPA-виджет: каталог, даты, слоты, билеты, оплата и билеты;
- оформление страницы оператора не меняется: стили виджета не протекают на страницу;
- параметры `hiddenBlocks`, `defaultPaymentMethod`, `theme` и `links` работают так же, как в PHP-режиме.

Чего нет без серверного слоя:

- поисковый робот видит страницу оператора и SPA, а не серверный HTML каталога и экскурсий;
- у экскурсий нет отдельных URL на домене оператора, `canonical`, Open Graph, JSON-LD и `sitemap` раздела;
- чистые ссылки `/excursions/{slug}` работают, только если сайт оператора сам отдаёт эту страницу; тогда передайте `publicBaseUrl` и `links` в `mountHodimaWidget`.

Если позже понадобится SEO, виджет перестраивать не нужно: добавьте PHP-модуль (ниже) или вызовите рендерер «Ходимы» из любого серверного окружения — Node, edge-функции или шаблона CMS. PHP для этого не обязателен.

Требования к хостингу

Требования ниже относятся к PHP-модулю; варианту «только JavaScript» они не нужны. Модуль ставится на обычный shared-хостинг и не требует Composer, фреймворков и Node.

- PHP 8.1 или новее.
- Apache с `mod_rewrite` и разрешённым `.htaccess` — для режима standalone и wildcard-маршрутов. На nginx `.htaccess` не работает: маршруты и запрет служебных каталогов настраиваются в конфиге сервера.
- Расширение cURL или включённый `allow_url_fopen`: модуль запрашивает рендерер по HTTPS.
- Записываемый каталог `cache/` — в нём лежит файловый кэш страниц.
- Исходящие HTTPS-запросы к рендереру и публичному API «Ходимы» не заблокированы.
- Собственный домен с HTTPS: canonical, Open Graph и ссылки строятся по нему.

Если cURL и `allow_url_fopen` запрещены хостингом, серверных страниц не будет: останется ручное встраивание виджета на страницу оператора.

Установка

1. Скачайте ZIP модуля со статического хоста платформы: `https://widget.<домен>/php/hodima-widget-php-<версия>.zip`. Домен платформы виден в прямой ссылке из настроек виджета, доступ к GitLab не нужен.

2. Распакуйте архив в подкаталог сайта. Для standalone это отдельный раздел, например `/excursions/`; для embedded и hybrid модуль можно положить вне web-root, например `/local/php/hodima/`.
3. Скопируйте `config.sample.php` в `config.php`. В ZIP файла `config.php` нет: его создаёт оператор, и при обновлении он не перезаписывается.
4. Заполните обязательные ключи `tenantId`, `renderBaseUrl`, `publicBaseUrl`, `apiBaseUrl` и шаблоны ссылок — таблица ниже.
5. Разрешите запись в `cache/`.
6. Откройте каталог в браузере и проверьте страницу с выключенным JS: в исходном HTML должен быть контент, а не пустой контейнер.
7. Проверьте `/excursions/sitemap.xml` — в ответе XML с каталогом и экскурсиями.

В ZIP уже есть `.htaccess`: он закрывает `config.php`, `cache/`, `src/` и `tests/` от доступа из браузера. `config.php` надёжнее держать вне web-root; если он лежит рядом с модулем, доступ к нему закрывает `.htaccess`.

`config.php`

Конфиг — обычный PHP-массив, который возвращает файл. Модуль читает его при каждом запросе и проверяет значения на загрузке.

Ключ	По умолчанию	Что задаёт
`renderBaseUrl`	обязателен	Адрес рендерера «Ходимы», например `https://widget.example.ru/render`.
`tenantId`	обязателен	Идентификатор организации: страницы всегда принадлежат ей и не берутся из запроса.
`apiBaseUrl`	обязателен	Публичный адрес API, к которому обращается браузер покупателя после гидратации.
`publicBaseUrl`	обязателен	Адрес сайта оператора без слэша на конце: из него строятся canonical и абсолютные ссылки.
`routePrefix`	`/excursions/`	Префикс раздела со слэшами на концах.

Ключ	По умолчанию	Что задаёт
`detailUrlPattern`	`publicBaseUrl` + `routePrefix` + слаг	Шаблон ссылки на страницу экскурсии; слаг подставляется в шаблон.
`catalogUrl`	`publicBaseUrl` + `routePrefix`	Адрес каталога и ссылки «← Все прогулки».
`glampingUrlPattern`	не задан	Шаблон ссылки на глэмпинг; в первой версии не используется, глэмпинги — этап 2.
`cacheDir`	`cache` рядом с модулем	Каталог файлового кэша.
`cacheTtl`	`600`	Время жизни записи кэша в секундах; рабочий диапазон — от 5 до 15 минут.
`layout`	`templates/layout.php`	Шаблон standalone-раздела с header и footer сайта оператора.
`presentation`	`full`	`full` — полный HTML виджета, `booking` — только панель брони.
`debug`	`false`	Подробный вывод ошибок рендера; в рабочей установке выключен.

```
<?php
// config.php
return [
    'renderBaseUrl' => 'https://widget.example.ru/render',
    'tenantId' => '00000000-0000-0000-0000-000000000000',
    'apiBaseUrl' => 'https://api.example.ru',
    'publicBaseUrl' => 'https://operator.ru',
    'routePrefix' => '/excursions/',
    'detailUrlPattern' => 'https://operator.ru/excursions/{slug}/',
    'catalogUrl' => 'https://operator.ru/excursions/',
    'cacheDir' => __DIR__ . '/cache',
    'cacheTtl' => 600,
    'presentation' => 'full',
    'debug' => false,
];
```

`renderBaseUrl`, `apiBaseUrl` и `publicBaseUrl` — абсолютные адреса `http(s)`; в рабочей установке только `https`. Пустой обязательный ключ, нечисловой `cacheTtl` или относительный

адрес дают понятное сообщение об ошибке конфига, а не пустую страницу. Секретов в конфиге нет: ключи ЮKassa, SMTP и пароли живут в админ-панели, а не на сайте оператора.

Три режима

Режим выбирается тем, откуда вызывается модуль: шаблон CMS, собственный `index.php` или wildcard-маршрут CMS. В embedded и hybrid URL, `head` и sitemap принадлежат CMS; в standalone всем владеет модуль.

Режим	Кто владеет URL и мета	Точка входа	Sitemap
Embedded	CMS: страницы, адреса, мета	шаблон страницы CMS вызывает `hodima_render`	CMS
Standalone	модуль: раздел `/excursions/*`	`.htaccess` направляет запросы в `index.php`	модуль, `/excursions/sitemap.xml`
Hybrid	CMS: wildcard-маршрут	маршрут CMS вызывает `hodima_render`	CMS по `hodima_sitemap_urls`

Embedded: страницы создаёт CMS

CMS создаёт страницы и владеет адресами, мета и sitemap, а модуль отдаёт готовую разметку виджета. Для каждой экскурсии нужна своя страница CMS или wildcard-шаблон, который подставляет слаг из адреса.

```
<?php
require __DIR__ . '/local/php/hodima/hodima.php';

$slug = 'gorodskaya-progulka'; // слаг из адреса страницы
$page = hodima_render('excursion', $slug, $_GET, 'full');
?>
<!doctype html>
<html lang="ru">
<head>
  <meta charset="utf-8">
  <?= hodima_head_tags($page) ?>
</head>
<body>
  <?= hodima_assets($page) ?>
  <?= $page['html'] ?>
```

```
  <?= hodima_mount($page) ?>
</body>
</html>
```

`hodima_render($route, $slug, $params, $presentation)` возвращает массив с ключами `html`, `head`, `assets` и `state`. Параметры страницы — дата и число билетов — передаются массивом; это обычно параметры запроса, модуль их проверяет. `hodima_assets($page)` печатает ссылки на CSS и JS этой версии, `hodima_mount($page)` — bootstrap с вызовом `mountHodimaWidget`.

В embedded финальный `head` принадлежит CMS, поэтому выберите один источник мета-тегов:

- SEO-плагин CMS включён на этих страницах — перенесите в его поля значения из `$page['head']`: заголовок, описание, canonical, robots, Open Graph и JSON-LD;
- SEO-плагин для этих страниц выключен — теги печатает `hodima_head_tags($page)`.

Оба способа одновременно не применяются: иначе на странице окажутся два `title` и два `canonical`. Ручной вызов `window.mountHodimaWidget` на страницах модуля не нужен: монтаж делает `hodima_mount($page)`, второй вызов смонтирует виджет дважды.

Standalone: раздел в подкаталоге

Модуль — самостоятельный раздел сайта: он владеет адресами, мета, canonical, sitemap и кэшем. CMS даёт только ссылку в меню, например `<a href="https://operator.ru/excursions/">Экскурсии</a>`.

Маршруты раздела:

- `/excursions/` — каталог;
- `/excursions/` и слаг экскурсии — страница экскурсии;
- `/excursions/sitemap.xml` — sitemap раздела.

Из коробки `.htaccess` направляет все запросы раздела в `index.php` и исключает существующие файлы и каталоги. Если модуль лежит не в `/excursions/`, поправьте `RewriteBase` и `routePrefix`.

```
Options -Indexes
DirectoryIndex index.php
```

```
<IfModule mod_rewrite.c>
  RewriteEngine On
  RewriteBase /excursions/
  RewriteCond %{REQUEST_FILENAME} !-f
  RewriteCond %{REQUEST_FILENAME} !-d
  RewriteRule ^ index.php [L]
</IfModule>
```

`layout.php` подключает header и footer сайта оператора, а между ними выводит `head`, ассеты, HTML и bootstrap. Свой layout кладётся в `templates/local/` и переопределяет шаблон модуля; эта папка сохраняется при обновлении.

Ссылку на sitemap раздела добавьте в `robots.txt` или в sitemap-index сайта:

```
Sitemap: https://operator.ru/excursions/sitemap.xml
```

Hybrid: маршрут CMS, рендер модуля

CMS регистрирует wildcard-маршрут и вызывает модуль; адрес и мета остаются у CMS, разметка — у модуля. Это удобно, когда экскурсии уже живут в структуре сайта, но описания ведутся в «Ходиме».

WordPress: правило перезаписи в `functions.php` и шаблон страницы.

```
add_action('init', function () {
    add_rewrite_rule('^excursions/([a-z0-9-]+)/?$', 'index.php?hodima_slug=$matches[1]',
'top');
});

add_filter('query_vars', function (array $vars): array {
    $vars[] = 'hodima_slug';
    return $vars;
});

add_filter('template_include', function (string $template): string {
    return get_query_var('hodima_slug')
        ? get_stylesheet_directory() . '/page-excursions.php'
        : $template;
});
```

```
<?php
// page-excursions.php
require get_template_directory() . '/hodima/hodima.php';
```

```
$page = hodima_render('excursion', get_query_var('hodima_slug'), $_GET, 'full');
$GLOBALS['hodima_page'] = $page;

add_action('wp_head', function () {
    echo hodima_head_tags($GLOBALS['hodima_page']);
}, 1);

get_header();
echo hodima_assets($page);
echo $page['html'];
echo hodima_mount($page);
get_footer();
```

Если мета-теги печатает SEO-плагин WordPress, не вызывайте `hodima_head_tags` и заполните поля плагина из `$page['head']`. После добавления правила зайдите в «Настройки» → «Постоянные ссылки» и сохраните их: WordPress пересоберёт правила перезаписи.

«1С-Битрикс»: компонент в `local/components/hodima/excursion/`, правило в `urlrewrite.php` ведёт адрес вида `/excursions/` и слаг на компонент.

```
<?php
// component.php
require $_SERVER['DOCUMENT_ROOT'] . '/local/php/hodima/hodima.php';

$arResult['PAGE'] = hodima_render('excursion', $arParams['SLUG'], $_GET, 'full');
$this->IncludeComponentTemplate();
```

```
<?php
// templates/.default/template.php
echo hodima_assets($arResult['PAGE']);
echo $arResult['PAGE']['html'];
echo hodima_mount($arResult['PAGE']);
```

Мета-теги в Битриксе задаются через `$APPLICATION->SetTitle()` и свойства страницы: значения берите из `$arResult['PAGE']['head']` или печатайте `hodima_head_tags($arResult['PAGE'])` один раз там, где мета не задаёт сама CMS.

Готовые примеры для Laravel, Blade и Twig — PHP-модуль.

Обновление

Версия модуля входит в имя архива: `hodima-widget-php-<версия>.zip`. Обновление — это замена файлов модуля; `config.php`, `templates/local/` и `cache/` при этом сохраняются.

1. Скачайте новый ZIP.
2. Распакуйте его поверх установки, не трогая `config.php`, `templates/local/` и `cache/`.
3. Откройте любую страницу раздела, затем диагностику модуля: версия должна смениться, предупреждений о несовместимости быть не должно.
4. Проверьте каталог, страницу экскурсии и `sitemap.xml`.

Ассетов виджета в ZIP нет: CSS и JS отдаёт статический хост по версии, поэтому обновление виджета не требует замены модуля. Старый HTML и его ассеты живут до истечения TTL кэша; ключ кэша включает версию, поэтому новая версия не получит старую разметку. Откат — предыдущий ZIP с теми же сохранёнными `config.php` и `templates/local/`.

Куда смотреть дальше

- SEO-страницы — что индексируется, какие адреса и мета получают страницы, как устроен sitemap.
- PHP-модуль — состав архива, все ключи конфига, кэш, безопасность и примеры для CMS.
- Проверка и типовые проблемы — что делать, если страница пустая, старая или пропала из поиска.
- Параметры — ручной вызов `mountHodimaWidget`, скрытые блоки и способ оплаты.
- Оформление — основа, цвета и шрифты виджета.

SEO-страницы

Какие страницы виджета попадают в поиск, какие адреса и мета они получают и как их находит робот.

Страницы раздела «Экскурсии» отдаёт PHP-модуль: он получает готовый HTML виджета из «Ходимы» и вставляет его в страницу оператора. Поисковик видит серверный HTML с описанием, мета-тегами и ссылками, а SPA подключается после загрузки и добавляет даты, слоты и бронь. PHP-модуль — готовый клиент рендерера; страницы можно собрать и из Node, edge-функции или шаблона CMS, если серверный слой вызывает `GET /render/…` и вставляет ответ в HTML. Как поставить модуль — Подключение на сайт.

Что индексируется

Индексируются каталог и страницы экскурсий — при условиях:

- экскурсия активна и не удалена;
- в горизонте расписания есть будущее отправление;
- контент не скрыт настройками организации в режиме `presentation=full`.

Активная экскурсия без будущих отправлений отдаёт контент, но получает `noindex` и не попадает в sitemap: покупателю страница полезна, роботу — нет. Экскурсия со скрытым SEO-критичным блоком в `full` — тоже `noindex` и вне sitemap. Черновик и архив отдают `404`: их нет ни в поиске, ни в sitemap.

Оформление, билеты, предпросмотр и личные страницы брони закрыты от индексации всегда. Это не ошибка: `noindex` на `checkout` и `tickets` — ожидаемое поведение модуля.

Глэмпинги — этап 2, после фичи `glampings-feature-switch`: в первой версии публичных страниц и эндпоинтов глэмпингов нет. Каталог на выбранную дату показывает всё доступное, а серверный каталог перечисляет только индексируемые экскурсии: списки намеренно различаются.

Адреса и canonical

Адреса строятся от префикса раздела; по умолчанию это `/excursions/`.

- `/excursions/` — каталог;

- `/excursions/` и слаг экскурсии — страница экскурсии.

Слаг генерируется из названия транслитерацией: до 120 символов, только строчные латинские буквы, цифры и дефис, уникален внутри организации. Слаг не меняется при переименовании экскурсии, поэтому старые ссылки продолжают работать. Служебные слова `sitemap.xml` и `glampings` в слаги не попадают.

`canonical` — абсолютный чистый адрес страницы на домене оператора. Фильтры, выбранная дата и число билетов не создают отдельных индексируемых страниц: такие адреса канонизируются на базовый. Пагинация каталога ссылается сама на себя.

Старые ссылки на параметры `hodima-excursion-id`, `hodima-trip-id` и `hodima-glamping-id` продолжают работать. В standalone-режиме модуль по возможности отвечает `301` на чистый адрес, если может сопоставить идентификатор со слагом; иначе открывает SPA. Существующие встраивания на одной странице не ломаются: параметры ссылки по-прежнему понимает сам виджет.

Метаданные

Заголовок страницы — название экскурсии плюс город и организация. Описание берётся из короткого описания и обрезается до 160 символов; если короткого нет — из первых 160 символов полного. Пустые поля не дают пустых тегов: тег просто не выводится.

Open Graph: `og:type`, `og:title`, `og:description`, `og:image`, `og:url` и `twitter:card`. Фотографии — абсолютные адреса, `alt` берётся из названия экскурсии.

JSON-LD описывает экскурсию как `TouristTrip` и `Product`: название, описание, изображение, предложение с ценой «от», валютой и наличием, организацию-провайдера и адрес страницы. Там же — `BreadcrumbList` «каталог → экскурсия» и `FAQPage`, если блок «Частые вопросы» виден. Язык разметки — `ru`.

Предложение в JSON-LD строится по цене «от», а не по расписанию: лента дат, слоты и тарифы остаются клиентскими и после гидратации подгружаются заново. Поэтому разметка не зависит от даты и не устаревает к полуночи.

Заголовки и описания для поиска не редактируются в админ-панели: они выводятся из полей экскурсии. Ручные переопределения появятся позже.

Режимы `full` и `booking`

Ключ `presentation` в `config.php` задаёт, что отдаёт модуль.

Режим	Что на странице	Кто отвечает за мета и индексацию
`full`	Полный HTML виджета: каталог или страница экскурсии с описанием, фото, маршрутом, возвратом и ценой «от».	«Ходима»: модуль выводит canonical, robots, Open Graph и JSON-LD.
`booking`	Контентные блоки скрыты, остаётся панель «дата → время → билеты».	CMS: описание уже на странице, мета и индексацию определяет сайт оператора.

`full` подходит для standalone-раздела и для страниц, у которых нет своего описания. `booking` — для страниц, где описание экскурсии уже написано в CMS: виджет добавляет только продажу и не дублирует текст.

Настройки организации модуль уважает и в `full`. Если оператор скрыл SEO-критичные блоки — шапку каталога, «Подробности» экскурсии или «Частые вопросы», — страница остаётся доступной покупателю, но получает `noindex` и выпадает из sitemap. Остальные скрытые блоки на индексацию не влияют. Порядок источников настроек тот же, что у виджета: адрес страницы → параметры монтирования → черновик предпросмотра → сохранённые настройки → умолчания.

Режим виджета ограничивает набор страниц: только прогулки, только глэмпинги, одна экскурсия или один глэмпинг. Страницы вне выбранного состава отдают `404` и не попадают в sitemap.

Ссылки

Карточки каталога — настоящие ссылки `<a href>` на страницы экскурсий по шаблону `detailUrlPattern`. Внутри карточки нет вложенных интерактивных элементов: ссылку можно открыть в новой вкладке, скопировать и обойти роботом.

Если SPA настроена перехватывать клик, переход идёт без перезагрузки; если нет — обычная полная перезагрузка страницы. На индексируемых страницах не остаётся переходов, работающих только через JS.

Ссылка «← Все прогулки» ведёт на `catalogUrl`, блоки «Ещё в этот день» и соседние глэмпинги — тоже реальные ссылки.

Если JS или ассеты не загрузились, серверный HTML с контентом и ссылками остаётся: страницу можно читать и обходить. Бронь в этом случае недоступна, а `<noscript>` объясняет, что для покупки нужен JavaScript.

Sitemap раздела

В sitemap попадают только индексируемые страницы: каталог и экскурсии с датой изменения. Оформления, билетов и страниц с `noindex` в нём нет.

- В standalone-режиме sitemap отдаёт модуль по адресу `/excursions/sitemap.xml`.
- В embedded и hybrid его собирает CMS по списку `hodima_sitemap_urls($config)`: функция возвращает абсолютные адреса страниц раздела по шаблонам оператора.

Ссылку на sitemap добавьте в `robots.txt` сайта или в его sitemap-index. Sitemap всего сайта оператора модуль не заменяет: он описывает только раздел «Экскурсии».

Ограничения

- Индексация — необходимое условие, но не обещание позиций: результат смотрите в Search Console и Яндекс Вебмастере.
- Мультиязычности и `hreflang` нет: страницы отдаются на русском.
- Одна и та же экскурсия может продаваться на сайтах нескольких операторов и партнёров, поэтому дубли контента между доменами возможны. Канонический адрес — страница своего оператора; кросс-доменный canonical на «Ходиму» не ставится, отдельная стратегия для партнёрских сайтов появится позже.
- Глэмпинги, отдельные адреса для комбинаций фильтров и дат и редактирование SEO-полей в админ-панели — вне первой версии.

Как проверить

1. Откройте страницу экскурсии с выключенным JavaScript: в исходном HTML должны быть описание, `title`, `canonical` и ссылки.
2. Проверьте, что на странице один `title`, один `canonical` и один `h1`.
3. Откройте `/excursions/sitemap.xml` и найдите в нём эту экскурсию.
4. Убедитесь, что на `checkout` и `tickets` стоит `noindex` — так и должно быть.

Если что-то не сходится — Проверка и типовые проблемы.

PHP-модуль

Справочник по модулю: состав архива, функции для шаблонов CMS, ключи `config.php`, кэш, sitemap и безопасность.

Модуль универсальный: он не зависит от CMS, не требует Composer и фреймворков и работает на PHP 8.1 и новее. Режимы embedded, standalone и hybrid и пошаговая установка — Подключение на сайт. Эта страница — справочник для разработчика сайта оператора и администратора CMS.

Как модуль общается с «Ходимой»



PHP-модуль стоит на хостинге оператора: это единственная часть, которая ходит в рендерер. Рендерер и публичный API работают в «Ходиме», ассеты лежат на статическом хосте по версии. Готовый ответ кэшируется на хостинге оператора; в «Ходиме» страницы не хранятся — рендерер только отдаёт `ETag` и `Cache-Control`. Если рендерер недоступен, модуль отдаёт устаревший кэш; если кэша нет — SPA-шелл с `noindex`.

Состав архива

Файл или каталог	Что внутри
`hodima.php`	Функции для шаблонов CMS: рендер, мета, ассеты, монтаж, список адресов для sitemap.
`index.php`	Вход standalone-раздела: маршруты, layout, отдача страниц и sitemap.

Файл или каталог	Что внутри
`config.sample.php`	Образец конфига; копируется в `config.php`.
`src/`	Router, RenderClient, Cache, Seo, Renderer.
`templates/`	`layout.php` для standalone, `shell.php` для SPA-шелла, `sitemap.php` для XML.
`assets/`	Собственные стили диагностики и страниц ошибок; ассетов виджета здесь нет.
`.htaccess`	Маршруты раздела и запрет доступа к служебным файлам и каталогам.
`README.md`	Короткая памятка по установке и режимам.

`cache/` создаётся при первом рендере, если каталога ещё нет. `config.php` и `tests/` в архив не входят: конфиг создаёт оператор, тесты остаются в репозитории модуля. Ассеты виджета отдаёт статический хост по версии, модуль получает их адреса из ответа рендерера.

Функции

Функция	Что делает
`hodima_render($route, $slug, $params, $presentation)`	Рендерит страницу и возвращает массив с ключами `html`, `head`, `assets` и `state`. `$route` — `catalog` или `excursion`; `$presentation` — `full` или `booking`.
`hodima_head_tags($page)`	Печатает мета-теги страницы: заголовок, описание, canonical, robots, Open Graph, JSON-LD, шрифты и preload.
`hodima_assets($page)`	Печатает ссылки на CSS и JS версии страницы.
`hodima_mount($page)`	Печатает bootstrap с вызовом `mountHodimaWidget` и параметрами страницы.
`hodima_sitemap_urls($config)`	Возвращает абсолютные адреса страниц раздела для sitemap CMS: каталог и индексируемые экскурсии.

```
<?php
require __DIR__ . '/local/php/hodima/hodima.php';
```

```
$page = hodima_render('excursion', 'gorodskaya-progulka', $_GET, 'full');

$page['html'];   // разметка виджета без <html> и <body>
$page['head'];   // title, description, canonical, robots, Open Graph, JSON-LD
$page['assets']; // адреса CSS и JS этой версии
$page['state'];  // сериализованное состояние для гидратации
```

`$params` — параметры страницы: дата, число билетов, выбранный тариф. Обычно это параметры запроса; модуль их проверяет и передаёт рендереру. `$route` вне `catalog` и `excursion` даёт ошибку: глэмпинги — этап 2.

`hodima_mount($page)` — единственный монтаж на страницах модуля. Ручной вызов `window.mountHodimaWidget` рядом с ним не нужен: виджет смонтируется дважды и покажет ошибку. Существующие ручные встраивания на других страницах сайта работают как раньше.

config.php

Ключи и значения по умолчанию — в таблице. Модуль проверяет конфиг при загрузке: обязательные ключи не пусты, адреса абсолютные `http(s)`, `cacheTtl` — число в допустимом диапазоне. Ошибка конфига даёт понятное сообщение и не превращается в пустую страницу.

Ключ	По умолчанию	Что задаёт
`renderBaseUrl`	обязателен	Адрес рендерера, например `https://widget.example.ru/render`; только `https` в рабочей установке.
`tenantId`	обязателен	Идентификатор организации: он фиксирован в конфиге и не берётся из запроса.
`apiBaseUrl`	обязателен	Публичный адрес API для браузера покупателя после гидратации.
`publicBaseUrl`	обязателен	Адрес сайта оператора без слэша на конце: база для canonical и абсолютных ссылок.
`assetBaseUrl`	пусто	База статического хоста ассетов: из неё SPA-шелл строит ссылки на CSS и JS, когда рендерер недоступен.

Ключ	По умолчанию	Что задаёт
`widgetVersion`	пусто: версия модуля	Версия в адресе ассетов: `<assetBaseUrl>/<widgetVersion>/hodima-widget.iife.js`.
`routePrefix`	`/excursions/`	Префикс раздела со слэшами на концах.
`detailUrlPattern`	`publicBaseUrl` + `routePrefix` + слаг	Шаблон ссылки на экскурсию; слаг подставляется в шаблон.
`catalogUrl`	`publicBaseUrl` + `routePrefix`	Адрес каталога и ссылки «← Все прогулки».
`glampingUrlPattern`	не задан	Шаблон ссылки на глэмпинг; в первой версии не используется.
`cacheDir`	`cache` рядом с модулем	Каталог файлового кэша; должен быть записываемым.
`cacheTtl`	`600`	Время жизни записи кэша в секундах; рабочий диапазон — от 5 до 15 минут.
`layout`	`templates/layout.php`	Шаблон standalone-раздела с header и footer оператора.
`presentation`	`full`	`full` — полный HTML виджета, `booking` — только панель брони.
`debug`	`false`	Подробный вывод ошибок рендера; в рабочей установке выключен.

`config.php` лежит вне web-root или закрыт `.htaccess`. Секретов в нём нет: ключи ЮKassa, SMTP и пароли учётных записей хранятся в админ-панели, а не на сайте оператора.

Режимы

Режим	Кто владеет URL и мета	Точка входа	Sitemap
Embedded	CMS	шаблон страницы вызывает `hodima_render`	CMS

Режим	Кто владеет URL и мета	Точка входа	Sitemap
Standalone	модуль	`.htaccess` → `index.php`	модуль, `/excursions/sitemap.xml`
Hybrid	CMS: wildcard-маршрут	маршрут CMS вызывает `hodima_render`	CMS по `hodima_sitemap_urls`

Standalone: модуль занимает подкаталог и владеет всем — адресами, мета, canonical, sitemap и кэшем. Embedded: CMS создаёт страницы и владеет мета; модуль отдаёт готовую разметку. Hybrid: адрес и мета у CMS, рендер у модуля.

`presentation` работает во всех режимах. `full` нужен там, где описания экскурсий берутся из «Ходимы»; `booking` — где описание уже написано в CMS. Что именно индексируется и как ведут себя скрытые блоки — SEO-страницы.

Layout и шаблоны

`templates/layout.php` — обвязка standalone-раздела: header и footer сайта оператора, а между ними мета, ассеты, HTML и bootstrap. Свой layout, шапка, подвал и страница ошибки кладутся в `templates/local/` и переопределяют шаблоны модуля. Папка `templates/local/` сохраняется при обновлении.

`templates/shell.php` — SPA-шелл с `noindex`. Он показывается, когда рендерер недоступен и в кэше нет даже устаревшей страницы: контейнер виджета пуст, но SPA монтируется и бронь работает. `templates/sitemap.php` — XML раздела для standalone-режима.

Кэш и версия

Кэш модуля — файловый, в каталоге `cacheDir`. Ключ записи — хеш от версии, организации, маршрута, слага, параметров, публичного адреса, набора шаблонов ссылок и `presentation`. Поэтому разные домены и разные наборы ссылок не смешивают кэш: для зеркала или второго домена нужна отдельная установка с отдельным конфигом.

Запись атомарная, с блокировкой от одновременной генерации одной страницы. Пока свежая страница не готова или рендерер недоступен, отдаётся устаревшая запись — это и есть stale-while-revalidate. Обновление происходит при следующем запросе после истечения `cacheTtl`. Ответы отдаются с `ETag` и `Last-Modified`: браузер и робот получают `304`, если страница не

менялась.

Версия входит в ключ кэша и в разметку. Разметка ссылается на неизменяемые ассеты своей версии, поэтому старый HTML не получит новые CSS и JS, а новый — старые. Старые версии живут не меньше TTL кэша модуля.

Sitemap

В standalone-режиме sitemap раздела отдаёт сам модуль по адресу `/excursions/sitemap.xml`. В embedded и hybrid его собирает CMS из списка функции:

```
<?php
require __DIR__ . '/local/php/hodima/hodima.php';

$config = require __DIR__ . '/local/php/hodima/config.php';
$urls = hodima_sitemap_urls($config);
```

Функция возвращает абсолютные адреса каталога и индексируемых экскурсий с датой изменения; список передаётся в генератор sitemap CMS. Оформления, билетов и страниц с `noindex` в нём нет. Sitemap всего сайта модуль не переписывает: раздел «Экскурсии» добавляется в общий файл или sitemap-index оператора, а ссылка на него — в `robots.txt`.

Безопасность

- Организация и адрес рендерера фиксированы в `config.php`: модуль не принимает организацию, произвольный URL или чужой домен из запроса, поэтому открытым прокси не становится.
- Слаги проверяются по шаблону из строчных латинских букв, цифр и дефиса; переходы вроде `../` и служебные слова дают `404`.
- HTML принимается только от доверенного рендерера; текстовые мета-теги экранируются.
- Состояние для гидратации сериализуется с экранированием `<`: содержимое экскурсии не может закрыть тег `script`.
- `.htaccess` из архива закрывает `config.php`, `cache/`, `src/` и `tests/`; на nginx те же запреты настраиваются в конфиге сервера.
- Запросы к рендереру идут с таймаутом и ограничением частоты; ошибки не отдают стек и пути сервера.

Обновление и диагностика

`HODIMA_PHP_VERSION` — версия модуля; она видна в диагностическом блоке вместе с версией рендерера и состоянием кэша. Если major-версия API рендерера не совпадает с ожидаемой, модуль предупреждает об этом: страницы продолжают работать, но обновление стоит поставить в план.

Порядок обновления:

1. Сохраните `config.php`, `templates/local/` и `cache/`.
2. Распакуйте новый ZIP поверх установки.
3. Откройте каталог и страницу экскурсии, затем диагностику: версия сменилась, предупреждений нет.
4. Проверьте `sitemap.xml` и исходный HTML страницы.

`debug` включают на время разбора: он показывает подробности ошибок рендера. На рабочей установке его выключают. Типовые симптомы и действия — Проверка и типовые проблемы.

Примеры

WordPress

Правило перезаписи и шаблон страницы — в `functions.php` темы и файле шаблона. Полный разбор режима hybrid — Подключение на сайт.

```
add_action('init', function () {
    add_rewrite_rule('^excursions/([a-z0-9-]+)/?$', 'index.php?hodima_slug=$matches[1]',
'top');
});

add_filter('query_vars', function (array $vars): array {
    $vars[] = 'hodima_slug';
    return $vars;
});

add_filter('template_include', function (string $template): string {
    return get_query_var('hodima_slug')
        ? get_stylesheet_directory() . '/page-excursions.php'
        : $template;
});
```

```
<?php
// page-excursions.php
require get_template_directory() . '/hodima/hodima.php';

$page = hodima_render('excursion', get_query_var('hodima_slug'), $_GET, 'full');
$GLOBALS['hodima_page'] = $page;

add_action('wp_head', function () {
    echo hodima_head_tags($GLOBALS['hodima_page']);
}, 1);

get_header();
echo hodima_assets($page);
echo $page['html'];
echo hodima_mount($page);
get_footer();
```

Если мета-теги печатает SEO-плагин, `hodima_head_tags` не вызывайте: заполните поля плагина из `$page['head']`. После правки правил перезаписи сохраните «Настройки» → «Постоянные ссылки», иначе WordPress отдаст `404`.

«1С-Битрикс»

Компонент в `local/components/hodima/excursion/`, правило в `urlrewrite.php` ведёт адрес вида `/excursions/` и слаг на компонент.

```
<?php
// component.php
require $_SERVER['DOCUMENT_ROOT'] . '/local/php/hodima/hodima.php';

$arResult['PAGE'] = hodima_render('excursion', $arParams['SLUG'], $_GET, 'full');
$this->IncludeComponentTemplate();
```

```
<?php
// templates/.default/template.php
echo hodima_assets($arResult['PAGE']);
echo $arResult['PAGE']['html'];
echo hodima_mount($arResult['PAGE']);
```

Мета-теги задаются через `$APPLICATION->SetTitle()` и свойства страницы: значения берите из `$arResult['PAGE']['head']`. Если мета не задаёт сама CMS, напечатайте `hodima_head_tags($arResult['PAGE'])` один раз.

Laravel

```
<?php
// routes/web.php
use Illuminate\Support\Facades\Route;

Route::get('/excursions/{slug}', function (string $slug) {
    $page = hodima_render('excursion', $slug, request()->query(), 'full');

    return view('excursions.show', ['page' => $page]);
});
```

```
{{-- resources/views/excursions/show.blade.php --}}
@extends('layouts.site')

@section('head')
  {!! hodima_head_tags($page) !!}
@endsection

@section('content')
  {!! hodima_assets($page) !!}
  {!! $page['html'] !!}
  {!! hodima_mount($page) !!}
@endsection
```

Twig

```
{# templates/excursion.twig #}
{% extends 'layout.twig' %}

{% block head %}
  {{ hodima_head_tags(page)|raw }}
{% endblock %}

{% block content %}
  {{ hodima_assets(page)|raw }}
  {{ page.html|raw }}
  {{ hodima_mount(page)|raw }}
{% endblock %}
```

Blade и Twig печатают разметку без экранирования: HTML приходит от доверенного рендерера, экранировать его повторно не нужно. Исключение — текстовые значения из `head`, если шаблон подставляет их в атрибуты вручную: их экранирует сам шаблонизатор.

Параметры

Параметры вызова `window.mountHodimaWidget`: организация и адрес API, основа и цвета, скрытые блоки и способ оплаты.

Организация и адрес API

Виджет монтируется явным вызовом со страницы оператора: при загрузке скрипт только регистрирует `window.mountHodimaWidget`. Параметр `target` — селектор контейнера, по умолчанию `#hodima-widget`.

```
window.mountHodimaWidget({
  target: '#hodima-widget',
  tenantId: 'YOUR_TENANT_ID',
  apiBaseUrl: 'https://api.example.com',
})
```

Организация берётся из первого непустого источника:

1. аргумент `tenantId` в `mountHodimaWidget`;
2. `TENANT_ID` или `VITE_TENANT_ID` сборки;
3. параметр `TenantId` в адресе страницы — только если это GUID.

Параметр `TenantId` в ссылке не переопределяет организацию, заданную при встраивании или сборкой. Если организация не задана ни одним источником, виджет не делает запросов и показывает плашку «Витрина не настроена: не указана организация. Проверьте ссылку или код встраивания».

Вызов возвращает `true`, если контейнер найден и виджет смонтирован, и `false`, если нет. Размонтировать виджет можно вызовом `window.unmountHodimaWidget()`.

Скрытые блоки (`hiddenBlocks`)

Часть блоков виджета оператору не нужна: промокод без промокодов, выбор оплаты при одном способе, шапка каталога со своей первой секцией на сайте. Менеджер скрывает их в админ-панели, а разработчик сайта может заменить набор при монтировании.

```
window.mountHodimaWidget({
  target: '#hodima-widget',
```

```
  hiddenBlocks: ['catalogHeader', 'excursionFaq'],
})
```

- Параметр заменяет список, сохранённый в админ-панели, целиком: `[]` показывает все блоки, отсутствие параметра — действует сохранённый список.
- Принимаются только имена из таблицы ниже. Неизвестное имя (опечатка, блок из другой версии виджета) не ломает монтаж: оно пропускается с предупреждением в консоль `[Hodima Widget] …`.
- Порядок источников: адрес страницы → `hiddenBlocks` → черновик предпросмотра → сохранённые настройки → умолчания. Через адрес блоки не задаются: набор для одной организации достигается только параметрами монтирования.
- Обязательные части покупки не скрываются: панель выбора даты, времени и билетов, форма покупателя, согласия, таймер брони, сообщения об ошибках, QR-коды и кнопки билетов.
- Подпись «Работает на платформе Ходима» внизу виджета тоже не отключается.

Экран	Блок	Имя
Каталог	Шапка каталога: иллюстрация, градиент, заголовок, город, плашки	`catalogHeader`
	Фильтры: формат, время суток, город	`catalogFilters`
	Лента дат и календарь цен	`catalogDateRail`
	Сортировка	`catalogSort`
Экскурсия	Фотогалерея	`excursionGallery`
	Ссылка «← Все прогулки»	`excursionBackLink`
	Подробности: факты и описание	`excursionDetails`
	Что включено	`excursionIncluded`
	Маршрут	`excursionRoute`
	Место встречи	`excursionMeetingPoint`
	Если планы изменятся (возврат)	`excursionRefund`
	Частые вопросы	`excursionFaq`
	Проживание к экскурсии	`excursionAddStay`

Экран	Блок	Имя
Глэмпинги	Фильтры списка	`glampingFilters`
	Сортировка списка	`glampingSort`
	Фотогалерея	`glampingGallery`
	Значки «Глэмпинг», «N прогулки рядом»	`glampingBadges`
	Питание	`glampingMealPlans`
	Дополнительно	`glampingAddons`
	Правила возврата	`glampingRefund`
	Экскурсии рядом	`glampingNearbyExcursions`
Оформление	Промокод	`checkoutPromoCode`
	Выбор способа оплаты	`checkoutPaymentMethod`
Билеты	Место встречи	`ticketsMeetingPoint`
	Что взять с собой	`ticketsWhatToBring`
	Плашка о возврате	`ticketsRefundNote`
	Ещё в этот день	`ticketsMore`

Период и число гостей проживания, а также число билетов в каталоге — входы поиска, а не фильтры: они остаются видимыми и при скрытых `glampingFilters` и `catalogFilters`. Скрытые «Питание» и «Дополнительно» означают, что бронь создаётся без питания и без дополнительных услуг. Блок на экране, недоступном в выбранном составе виджета, просто не встречается.

Способ оплаты при скрытом выборе (`defaultPaymentMethod`)

```
window.mountHodimaWidget({
  target: '#hodima-widget',
  hiddenBlocks: ['checkoutPaymentMethod'],
  defaultPaymentMethod: 'sbp', // 'card' (по умолчанию) | 'sbp'
})
```

Когда блок `checkoutPaymentMethod` скрыт, виджет платит этим способом — тем же путём, что

при ручном выборе: `'sbp'` уходит в СБП, карта открывает встроенную форму ЮKassa. Когда выбор виден, параметр не используется: на телефоне по умолчанию СБП, на компьютере — карта.

Неподдерживаемое значение (например, `'bitcoin'`) игнорируется с предупреждением `[Hodima Widget] …`: действует способ из админ-панели. Способ из параметров не проверяет подключение в ЮKassa: если СБП у организации не подключён, платёж не пройдёт.

Основа и цвета (`theme`)

Объект `theme` задаёт основу и пять цветов виджета. Остальные цвета — вторичный текст, рамки, статусы, декор — выводятся из них пакетом `@hodima/ui`, виджет их не дублирует.

Поле	CSS-переменная	Умолчание	Что окрашивает
`scheme`	—	`'light'`	Основа: готовая заготовка фона, карточек, текста и статусов. Допустимо `'light'` и `'dark'`.
`backgroundColor`	`--hd-bg`	из основы	Фон виджета.
`surfaceColor`	`--hd-surface`	из основы	Карточки, панели, поля ввода, диалоги.
`textColor`	`--hd-text`, `--hd-icon`	из основы	Основной текст и иконки.
`primaryColor`	`--hd-primary`	из настроек (`#3F65E9`)	Кнопки, выбор, ссылки, выделение.
`brandColor`	`--hd-brand`	из настроек (`#6986EB`)	Логотип, декор и градиент шапки каталога.

```
window.mountHodimaWidget({
  target: '#hodima-widget',
  theme: {
    scheme: 'dark',
    backgroundColor: '#121212',
    surfaceColor: '#1E1E1E',
    textColor: '#F2F2F2',
    primaryColor: '#F0A030',
    brandColor: '#F0A030',
  },
```

```
})
```

Пустые `backgroundColor`, `surfaceColor` и `textColor` берутся из заготовки основы, поэтому на тёмном сайте достаточно `scheme: 'dark'`. Цвета принимаются в любом CSS-формате (`#abc`, `#AABBCC`, `rgb(18, 18, 18)`, имя цвета) и приводятся к `#RRGGBB`. Некорректное значение игнорируется с предупреждением в консоль `[Hodima Widget] …`: действует сохранённый цвет или заготовка основы, виджет не падает.

Шрифты, отступы и скругление оболочки задаются в том же объекте `theme` — поля `fontBody`, `fontHeading`, `shellPaddingBlock`, `shellPaddingInline`, `shellBorderRadius` и `shellPaddingMobile`. `fontBody` и `fontHeading` — только семейство, без ссылки на файл: файл шрифта подключает страница оператора до монтирования, иначе останется запасная цепочка `system-ui`. Ссылку на файл шрифта задаёт менеджер в настройках организации, и виджет подключает её сам — см. Оформление. Как основа и цвета выглядят на экране и какие предупреждения о читаемости выводит виджет — там же.

Параметры монтирования главнее сохранённых настроек организации. Те же значения менеджер задаёт в админ-панели, поэтому один и тот же виджет на разных страницах сайта можно оформить по-разному.

Оформление

Как настроить внешний вид виджета под сайт оператора.

Светлая и тёмная основа

Основа — готовая заготовка оформления: фон, карточки, текст, статусы и декор. «Светлая» — оформление по умолчанию; «Тёмная» нужна сайтам с тёмным оформлением, чтобы виджет не светился белым прямоугольником. На тёмном сайте достаточно выбрать основу: фон, карточки и текст берутся из заготовки, отдельно их задавать не нужно.

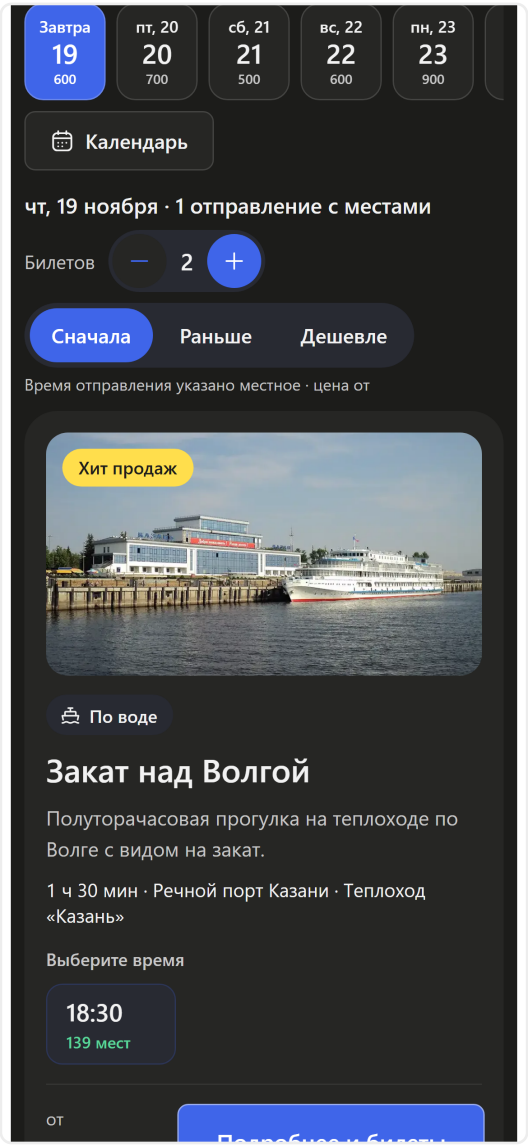

Figure 1. Тёмная основа на телефоне: фон, карточки и текст из заготовки

Пока настройки организации грузятся, виджет не показывает экраны — поэтому светлая основа не мигает перед тёмной. Если настройки недоступны, виджет работает на значениях по

умолчанию.

Пять цветов

Оператор настраивает пять цветов; остальные оттенки виджет выводит из них.

Цвет	Поле	Что окрашивает	Переменная
Фон	`backgroundColor`	подложку виджета	`--hd-bg`
Карточки	`surfaceColor`	карточки, панели, поля ввода, диалоги	`--hd-surface`
Текст	`textColor`	основной текст и иконки	`--hd-text`
Основной	`primaryColor`	кнопки, выбор, ссылки, выделение	`--hd-primary`
Бренд	`brandColor`	логотип, декор и градиент шапки каталога	`--hd-brand`

Фон, карточки и текст необязательны: пустое значение берётся из основы и в админ-панели показано серым с пометкой «как в основе». Основной и брендовый цвет хранятся всегда и от основы не зависят.

Остальные цвета — вторичный текст, рамки, разделители, фоны статусов, оттенки основного и брендового цвета — выводит пакет дизайн-системы. Виджет их не дублирует и отдельно не настраивает, поэтому оформление остаётся согласованным на обеих основах.

Цвета принимаются в любом CSS-формате: `#abc`, `#AABBCC`, `rgb(18, 18, 18)`, имя цвета. Виджет приводит их к `#RRGGBB`, а некорректное значение не применяет и не роняет монтаж: действует сохранённый цвет или заготовка основы, в консоль уходит предупреждение `[Hodima Widget]` ….

Читаемость и предупреждения

Виджет проверяет контраст и пишет предупреждения в консоль браузера `[Hodima Widget]` …, но ничего не блокирует:

- текст к фону и к карточкам ниже 4,5:1 — покупателю будет тяжело читать;
- на основном цвете пришлось поставить тёмный текст: белый не проходит проверку AA.

Тёмный текст на кнопках и выбранных элементах виджет ставит сам, поэтому кнопка остаётся читаемой даже на жёлтом или светло-зелёном основном цвете. В админ-панели те же предупреждения показаны под карточкой «Цвета», и сохранить настройки можно всё равно: решение за менеджером, а результат виден в предпросмотре.



Светлый текст на светлых карточках и тёмный текст на тёмной кнопке — самая частая ошибка при ручном подборе цветов. Смотрите предпросмотр, а не только сообщения.

Шрифты и форма оболочки

Шрифт основного текста и шрифт заголовков задаются семейством и ссылкой на файл. Ссылка — только `https`: файл `.woff2`, `.woff`, `.ttf`, `.otf` или Google Fonts. Виджет подключает шрифт сам: для Google Fonts добавляет на страницу ссылку на таблицу стилей, для файла — правило `@font-face`. Ссылку с другим протоколом или на другой файл виджет пропускает: остаётся введённое семейство и запасная цепочка `system-ui`. Файлы шрифтов в стили виджета не встраиваются — бюджет CSS этого не позволяет, поэтому шрифт скачивает браузер покупателя.

Отступы и скругление оболочки задают поля `shellPaddingBlock`, `shellPaddingInline`, `shellBorderRadius` и `shellPaddingMobile`. Пустое значение — отступы по умолчанию, они подстраиваются под ширину экрана.

Подпись платформы

В самом низу виджета на всех экранах, в предпросмотре и на самостоятельной странице стоит строка «Работает на платформе Ходима» со ссылкой на https://hodima.ru/. Строка мелкая и приглушённая, но на тёмной основе тоже видна: её цвета берутся из темы. Отключить подпись нельзя — ни в настройках организации, ни списком скрытых блоков, ни параметрами монтирования.

Где менять оформление

Ту же основу, пять цветов, шрифты и радиус оболочки менеджер задаёт в админ-панели: «Настройки» → «Виджет», карточка «Цвета». Параметры монтирования `theme` главнее сохранённых настроек: ими разработчик сайта переопределяет оформление для отдельной страницы. Порядок источников: значения по умолчанию → сохранённые настройки → параметры `mountHodimaWidget`. Подробнее о параметрах — Параметры, о карточках в админ-панели —

документ «Руководство пользователя», раздел «Настройки организации».

Проверка и типовые проблемы

Как проверить установку модуля и что делать, если страница пустая, старая или пропала из поиска.

Сначала проверьте страницу без JavaScript: поисковик и покупатель с медленной связью видят именно её. Потом включите JS и убедитесь, что виджет гидрирует и продаёт.

Быстрая проверка

1. Откройте страницу экскурсии с выключенным JavaScript или через `curl`. В исходном HTML должны быть описание, `title`, `canonical`, Open Graph и ссылки `<a href>`, а не пустой контейнер.
2. Проверьте, что на странице один `title`, один `canonical` и один `h1`.
3. Откройте `/excursions/sitemap.xml`: в ответе XML с каталогом и индексируемыми экскурсиями и датой изменения.
4. Проверьте код ответа: активная экскурсия — `200`, черновик и архив — `404`.
5. Включите JavaScript: виджет гидрирует, даты и слоты загружаются, бронь и оплата работают.
6. Откройте оформление и билеты: `noindex` на них — норма, а не ошибка.

Если страница не открывается вовсе, начните с требований хостинга — Подключение на сайт.

Симптом, причина, действие

Симптом	Вероятная причина	Что делать
Страница пустая: контейнер виджета без разметки	Рендерер недоступен, а в кэше нет даже устаревшей страницы.	Проверьте `renderBaseUrl`, доступность адреса рендерера и его `/render/health`, исходящие HTTPS-запросы и расширение cURL. Пока рендерер недоступен, модуль отдаёт SPA-шелл с `noindex`: бронь работает, но страница не индексируется.

Симптом	Вероятная причина	Что делать
Контент есть, но старый	Запись кэша ещё не истекла или рендерер не отвечает и отдаётся устаревшая копия.	Дождитесь `cacheTtl` (5–15 минут) или очистите `cache/`; проверьте доступность рендерера и включите `debug` на время разбора.
Страница в `noindex`, хотя должна индексироваться	В `presentation=full` скрыт SEO-критичный блок — шапка каталога, «Подробности» экскурсии или «Частые вопросы»; либо у экскурсии нет будущих отправлений; либо включён `booking`.	Верните блоки в настройках организации, добавьте отправление в расписание, проверьте `presentation` и состав виджета.
Разметка без стилей	Шаблон не вывел `hodima_assets($page)` или ассеты версии недоступны.	Добавьте вызов в шаблон; проверьте, что статический хост отдаёт CSS и JS этой версии и они не закрыты на сайте оператора.
Виджет не гидрирует, даты не загружаются	Версия HTML и ассетов не совпала, ошибка JS или недоступен публичный API.	Откройте консоль браузера и сравните версию в разметке с версией ассетов; проверьте `apiBaseUrl` и доступность API. Клиент в этом случае очищает контейнер и монтируется заново: контент страницы остаётся.
Ошибка `500` или пустая страница PHP	PHP старше 8.1, запрещены cURL и `allow_url_fopen`, нет прав на `cache/` или ошибка в `config.php`.	Посмотрите лог PHP; сверьте версию PHP, расширения, права на каталог кэша и обязательные ключи конфига.
`404` на `/excursions/`	Не работает `mod_rewrite`, запрещён `.htaccess` или не совпадают `RewriteBase` и `routePrefix`.	Включите `mod_rewrite` и `AllowOverride`, поправьте `RewriteBase` в `.htaccess` и префикс в конфиге.
Два `title` или два `canonical`	Мета-теги печатают и SEO-плагин CMS, и `hodima_head_tags($page)`.	Оставьте один источник: либо значения из `$page['head']` в плагине, либо теги модуля.

Симптом	Вероятная причина	Что делать
Sitemap пуст или без экскурсий	Нет индексируемых страниц: черновики, нет будущих отправлений, скрытые блоки; либо неверные шаблоны ссылок; либо CMS не вызывает `hodima_sitemap_urls`.	Проверьте экскурсии и расписание, `detailUrlPattern` и `catalogUrl`, вызов функции в sitemap CMS.
WordPress отдаёт `404` на странице экскурсии	Правила перезаписи не пересобраны или их перекрывает другой плагин.	Сохраните «Настройки» → «Постоянные ссылки», отключите конфликтующий плагин, проверьте правило в `functions.php`.
Предупреждение о несовпадении версии	Major-версия API рендерера отличается от ожидаемой модулем.	Обновите ZIP модуля: страницы продолжают работать, но совместимость стоит восстановить.

Что смотреть в конфиге и диагностике

В `config.php` проверяйте ключи, от которых зависят адреса и запросы:

- `renderBaseUrl`, `tenantId`, `apiBaseUrl` — куда модуль обращается и от чьего имени строит страницы;
- `publicBaseUrl`, `routePrefix`, `detailUrlPattern`, `catalogUrl` — какие адреса попадают в canonical, ссылки и sitemap;
- `cacheDir`, `cacheTtl` — где лежит кэш и как быстро он обновляется;
- `presentation` — полный HTML виджета или только панель брони.

Диагностический блок модуля показывает версию модуля `HODIMA_PHP_VERSION`, версию рендерера и состояние кэша. Ключ `debug` включает подробный вывод ошибок рендера: включайте его на время разбора и выключайте после. Секретов в конфиге быть не должно: ключи ЮKassa, SMTP и пароли живут в админ-панели.

Обновление

1. Скачайте свежий ZIP со статического хоста платформы.

2. Распакуйте его поверх установки, сохранив `config.php`, `templates/local/` и `cache/`.
3. Откройте каталог, страницу экскурсии и диагностику: версия сменилась, предупреждений нет.
4. Проверьте `sitemap.xml` и исходный HTML страницы без JavaScript.

Если после обновления страницы пустые, верните предыдущий ZIP: конфиг и локальные шаблоны при откате не меняются.

Куда смотреть дальше

- SEO-страницы — что индексируется, canonical, метаданные и sitemap.
- PHP-модуль — ключи конфига, кэш, безопасность и примеры для CMS.
- Подключение на сайт — требования хостинга, установка и режимы.