

Документация

МСП-шлюз native.msp

Версия 1.0.0 от 27.10.2026

Содержание

Страниц: 38 · Слов: 4 264 · Время чтения: ≈ 22 мин

Описание модуля	3
Начало работы	
Установка	6
Удаление	9
Доступ и группы	10
ИИ-агенты	
Ручное подключение	12
Инструкция для агента	19
Разработчикам	
Инструменты	22
Протокол JSON-RPC	26
Разработчикам — Примеры инструментов	
Собственный инструмент	29
Название сайта	31
Версия модуля	33
Список заказов	35
Используемые ресурсы	38

Описание модуля

Название	Описание
Наименование	МСР-шлюз
Код модуля	native.mcp
Дата релиза	27.10.2026
Дата обновления	27.10.2026
Версия	1.0.0
Зависимость	<u>API-платформа</u> 2.0.0 и новее

МСР-шлюз — расширение API-платформы для управляемого взаимодействия сайта с ИИ-агентами по протоколу MCP (Model Context Protocol). Он предоставляет агенту явно разрешённые роуты API в виде инструментов: агент получает их список, выбирает нужный и вызывает его.

Кому что читать

- **Администратору сайта.** «Установка» и «Доступ и группы»: включить шлюз и добавить пользователей в нужные группы. Дальше — «Ручное подключение» или «Инструкция для агента», чтобы агент подключился сам.
- **Разработчику сайта.** «Инструменты»: как объявить роут инструментом, какие ключи есть у секции `mcp` и как ограничить инструмент группами. Готовые разборы — в примерах инструментов, детали обмена с агентом — в «Протоколе JSON-RPC».

Как это работает

1. Разработчик сайта описывает роуты в карте роутов API-платформы и добавляет нужным роутам секцию `mcp`.
2. Агент обращается к роуту `POST /mcp` относительно пути API, например `/api/mcp`, с access-токеном API-платформы. Настройки популярных агентов приведены в разделе «ИИ-агенты» → «Ручное подключение».
3. По запросу `tools/list` шлюз возвращает роуты с секцией `mcp`, доступные группам пользователя токена.

- По запросу `tools/call` шлюз проверяет аргументы и выполняет выбранный роут общим конвейером API-платформы: с проверками доступа, лимитами, событиями и кешем.
- Результат контроллера возвращается агенту в ответе JSON-RPC.

Шлюз не содержит бизнес-логики. Что агент может прочитать или изменить на сайте, определяют роуты и их контроллеры.

Ответственность

API-платформа	MCP-шлюз
Путь, описание и проверка доступа роута <code>/mcp</code>	Протокол JSON-RPC, версии и заголовки MCP
Токены, клиенты, группы, лимиты, CORS и журналирование	Каталог инструментов и проверка аргументов
Флаг активации шлюза на вкладке «Роуты»	Встроенные инструменты сведений о сайте и платформе
Карты роутов и константы секции публикации <code>ToolConfig</code>	Группы «MCP: Клиенты» и «MCP: Заказы», которые создаёт и удаляет установщик

Собственных таблиц, токенов, журналов и административных страниц у MCP-шлюза нет.

Требования

- API-платформа 2.0.0 и новее: действующий модуль или активный демонстрационный период.
- Версия PHP — не ниже минимальной для установленной API-платформы, см. системные требования.
- «1С-Битрикс: Управление сайтом» или Битрикс24 (коробочная версия).

Ограничения версии

- Вход через OAuth не поддерживается: агент передаёт access-токен API-платформы в заголовке.
- Поддерживаются инструменты. Resources, prompts, tasks и отдельный SSE-поток не реализованы.

- Подключение, получение каталога и вызов инструментов проверены в Codex и Claude.

Установка

Текст

- 1 1. API-платформа 2.0.0+ установлена и запущена?
 - 2 └ Нет → установить и запустить
 - 3 └ Да ↓
- 4
- 5 2. Установить native.msp
 - 6 └ Проверки не пройдены → установка остановлена
 - 7 └ Проверки пройдены ↓
- 8
- 9 3. Добавить пользователей в группы
 - 10 └ «МСП: Клиенты» → от чьего имени работают агенты (обязательно)
 - 11 └ «МСП: Заказы» → только для демо-инструмента заказов
- 12
- 13 4. Отметить флажок МСП-шлюза: API-платформа → Настройки модуля → Роуты
- 14
- 15 5. Проверить GET /up/msp (нужны сервисные роуты)
 - 16 └ 200 → шлюз активен ↓
 - 17 └ 503 MODULE_INACTIVE → шлюз выключен, вернуться к шагу 4
- 18
- 19 6. Логировать входящие запросы? (необязательно)
 - 20 └ Да → API-платформа → Настройки модуля:
 - 21 | «Аналитика» → «Собирать входящие запросы»
 - 22 | «Логирование» → группа «Входящие запросы»
 - 23 └ Нет → пропустить
- 24
- 25 Далее → опубликовать инструменты, подключить агента

API-платформа

Установите и запустите API-платформу версии 2.0.0 или новее. МСП-шлюз работает только вместе с ней: без установленной и доступной API-платформы установка будет остановлена.

Установите модуль

1. Загрузите модуль native.msp на сайт.
2. Откройте «Marketplace → Установленные решения».
3. Найдите «МСП-шлюз» и выберите «Установить».

Перед регистрацией модуля установщик проверяет окружение:

Проверка	Сообщение при ошибке
API-платформа зарегистрирована, действующая или в демо-периоде	Модуль «API-платформа» не установлен

Версия API-платформы не ниже 2.0.0 и есть нужные возможности	Модуль «API-платформа» не поддерживает необходимые возможности MCP-шлюза
Версия PHP не ниже <u>минимальной</u> для API-платформы	Версия PHP должна быть не ниже ...

Истёкший демонстрационный период API-платформы тоже останавливает установку.

Группы пользователей

При установке создаются две группы:

Группа	Код	Назначение
«MCP: Клиенты»	NATIVE_MCP_CLIENT	Вход в шлюз: без неё запросы к /mcp не проходят
«MCP: Заказы»	NATIVE_MCP_ORDER	Доступ к демонстрационному инструменту заказов

Если группа с таким кодом уже есть, она и её участники сохраняются.

Добавьте в «MCP: Клиенты» пользователей, от имени которых будут работать агенты, а в «MCP: Заказы» — тех, кому нужен демонстрационный инструмент заказов из статьи [«Список заказов»](#). Подробнее — в разделе [«Доступ и группы»](#).

Активация шлюза

Установка не включает шлюз автоматически.

1. Откройте «Сервисы → API-платформа → Настройки модуля», вкладка «Роуты».
2. Отметьте флажок активации MCP-шлюза и сохраните настройки.

Проверка шлюза

Для проверки состояния включите сервисные роуты и вызовите [GET /up/mcp](#). Активный шлюз отвечает 200, выключенный — MODULE_INACTIVE / 503.

Логирование запросов

Необязательный шаг. Собственного журнала у MCP-шлюза нет: обращения к **/mcp** записывает API-платформа, как и остальные входящие запросы.

1. Откройте «Сервисы → API-платформа → Настройки модуля».
2. На вкладке «Аналитика» включите «Собирать входящие запросы».
3. На вкладке «Логирование» в группе «Входящие запросы» отметьте «Записывать входящие данные» и «Записывать исходящие данные» и выберите место хранения: база, файл или оба.
4. Сохраните настройки.

Записи появятся в разделе «Входящие запросы».

Дальше опубликуйте инструменты, получите токен и подключите ИИ-агента к шлюзу.

Удаление

Удаление модуля

1. Откройте «Marketplace → Установленные решения».
2. Найдите «MSP-шлюз» и выберите «Удалить».

Что происходит	Подробности
Снимается регистрация модуля	Роут /msp и встроенные инструменты перестают работать
Удаляются группы «MSP: Клиенты» и «MSP: Заказы»	Пользователи остаются на сайте, снимается только их членство в группах
Сбрасывается флажок активации шлюза	Флажок на вкладке «Роуты» API-платформы выключается
Очищается кеш API-платформы	Карты роутов пересобираются без маршрутов MSP

Токены, клиенты, настройки и карты роутов API-платформы сохраняются. Секции msp в пользовательских картах остаются и снова работают после повторной установки. Истёкший демонстрационный период MSP-шлюза или API-платформы удалению не мешает.

Если ядро Битрикса откажется удалять группу, например из-за обработчика другого модуля, появится сообщение «Не удалось удалить группу «MSP: Клиенты»». Остальная очистка всё равно будет выполнена.

Удаление API-платформы

Если установлен MSP-шлюз, в мастере удаления API-платформы есть флажок «Удалить «MSP-шлюз»». При его выборе мастер выполняет то же штатное удаление MSP-шлюза. Без него MSP-шлюз останется установленным, но без API-платформы работать не будет.

Повторная установка

При повторной установке группы «MSP: Клиенты» и «MSP: Заказы» создаются заново: добавьте в них пользователей ещё раз. Шлюз нужно снова включить вручную.

Доступ и группы

Агент работает от имени пользователя сайта, чей токен указан в его настройках. Доступ проверяется на двух уровнях:

1. **Вход в шлюз.** Роут `/mcp` принимает access-токен API-платформы только от пользователя группы «MCP: Клиенты» (NATIVE_MCP_CLIENT).
2. **Инструменты.** Каждый инструмент — роут API-платформы со своими списками групп. Агент видит и вызывает только роуты, доступные группам пользователя.

Пользователь

Агента можно подключить от имени любого пользователя сайта, в том числе уже существующего клиента. Добавьте его в группу «MCP: Клиенты».

Не подключайте агента от имени администратора: администраторы не ограничены списками групп роутов и увидят все инструменты.

Токен

Агент передаёт access-токен пользователя в заголовке `Authorization: Bearer`.

Токены выпускает API-платформа: сам пользователь через API или администратор вручную. Способы выпуска, сроки, обновление и привязка к клиенту описаны в разделе «Токены авторизации».

Чтобы отключить агента, отзовите токен или исключите пользователя из группы «MCP: Клиенты».

Настройка Claude, Codex и других агентов — в разделе «Ручное подключение». Чтобы агент подключился сам, передайте ему «Инструкцию для агента».

Разграничение по группам

Группа «MCP: Клиенты» даёт вход в шлюз, но не определяет набор инструментов. Чтобы разные пользователи видели разные инструменты, создайте дополнительные группы и укажите их в whitelist нужных роутов, см. «Инструменты».

Так устроен пример «Список заказов»: инструмент доступен только группе «MCP: Заказы» (NATIVE_MCP_ORDER). Чтобы агент его увидел, пользователь должен входить и в «MCP: Клиенты», и в «MCP: Заказы». Роуты без whitelist доступны

всем пользователям «MCP: Клиенты».

Правила whitelist и blacklist групп — в разделе «Безопасность роутов». Кроме того:

- Вызов недоступного инструмента по имени возвращает «Unknown or unavailable tool»: при выполнении группы проверяются повторно.
- Вне шлюза тот же роут можно вызвать обычным REST-запросом, если пользователь проходит его ограничения.

Ручное подключение

ИИ-агент подключается к сайту как к удалённому MCP-серверу. Для подключения нужны два значения:

Значение	Пример	Где указать
Адрес сервера	https://example.com/api/mcp	Поле URL сервера
Токен доступа	ACCESS_TOKEN	Заголовок Authorization со значением Bearer ACCESS_TOKEN

Значение заголовка начинается со слова Bearer и пробела, затем следует токен. Не передавайте токен другим людям: агент с ним получает доступ к инструментам сайта от имени вашего пользователя.

Клиент должен поддерживать протокол MCP версии 2025-11-25 или 2026-07-28 и транспорт Streamable HTTP. Вход через OAuth шлюз не поддерживает, поэтому токен всегда указывается в заголовке.

Чтобы агент сам провёл вас по шагам подключения, передайте ему ссылку на страницу «Инструкция для агента».

Подготовка на сайте

Шлюз должен быть установлен и включён, а пользователь, от имени которого работает агент, — входить в группу «MCP: Клиенты» и иметь access-токен. См. «Доступ и группы».

Адрес сервера состоит из адреса сайта, пути API и /mcp .

Приложения

Claude

Веб-версия claude.ai и приложения Claude подключают сервер как пользовательский коннектор. Запросы к серверу идут из инфраструктуры Anthropic, поэтому сайт должен быть доступен из интернета.

1. Откройте страницу Customize → Connectors и нажмите **Add custom connector**.
2. В поле **MCP server URL** укажите адрес сервера.

3. В разделе **Authentication** выберите **No sign-in**.
4. В разделе **Request headers** выберите заголовок `authorization` и введите значение `Bearer ACCESS_TOKEN`.
5. Нажмите **Add**.

Раздел **Request headers** находится в бета-версии и доступен не всем организациям. Если его нет, подключить шлюз в Claude пока нельзя: используйте Codex или Claude Code. В тарифах Team и Enterprise участник не может добавить коннектор сам. Его добавляет владелец организации на странице Organization settings → Connectors через **Add** → **Custom**, а участники подключают его в **Customize** → **Connectors**. Токен из **Request headers** становится общим для всех участников.

Документация клиента: Add a connector that isn't in the directory.

Claude Desktop

Способ подходит, если сайт недоступен из интернета, например работает на `localhost`. Claude Desktop запускает на компьютере пользователя мост mcp-remote: он обращается к сайту и передаёт токен в заголовке. Нужен установленный Node.js.

1. Узнайте полный путь к `npx` командой `which npx` в терминале macOS или `where npx` в Windows. Приложение, запущенное не из терминала, может не найти `npx` по короткому имени.
2. Откройте файл `~/Library/Application Support/Claude/claude_desktop_config.json` в macOS или `%APPDATA%\Claude\claude_desktop_config.json` в Windows.
3. Добавьте сервер в объект `mcpServers`, указав полный путь к `npx`, адрес сервера и токен:

JSON

```

1  {
2    "mcpServers": {
3      "site": {
4        "command": "/opt/homebrew/bin/npx",
5        "args": [
6          "-y",
7          "mcp-remote",
8          "http://localhost/api/mcp",
9          "--header",
10         "Authorization:${AUTH_HEADER}"
11       ],
12       "env": {
13         "AUTH_HEADER": "Bearer ACCESS_TOKEN"
14       }
15     }
16   }
17 }

```

4. Полностью закройте Claude Desktop и запустите его снова.

Значение заголовка передаётся через переменную `AUTH_HEADER`, чтобы пробел после `Bearer` не разделил аргумент. Если `mcp-remote` отказывается подключаться по `http://`, добавьте в `args` параметр `--allow-http`. Журнал подключения в macOS находится в файле `~/Library/Logs/Claude/mcp-server-site.log`.

Документация клиента: [mcp-remote](#).

ChatGPT

Сервер подключается в Codex — агенте приложения ChatGPT. Обычный чат ChatGPT подключить нельзя: он принимает только серверы с OAuth и не передаёт токен в заголовке.

1. Откройте **Settings → MCP servers** и нажмите **Add server**.
2. Введите название сервера, например `site`, и выберите тип **Streamable HTTP**.
3. В поле URL укажите адрес сервера.
4. Добавьте заголовок `Authorization` со значением `Bearer ACCESS_TOKEN`. Поле «Переменная окружения токена Bearer» оставьте пустым: в нём указывается имя переменной, а не токен.
5. Сохраните сервер и нажмите **Restart**.

Список подключённых серверов показывает команда `/mcp` в поле ввода. Если в форме нет полей для заголовков, укажите токен в файле настроек Codex. Приложение ChatGPT, Codex CLI и расширение Codex для редакторов используют общие настройки MCP-серверов.

Документация клиента: Model Context Protocol в Codex.

VS Code

Для агента GitHub Copilot:

1. Выполните команду **MCP: Add Server** из палитры команд.
2. Выберите тип **HTTP** и укажите адрес сервера.
3. Введите идентификатор сервера, например `site`, и выберите, где сохранить настройки: для всех проектов или только для текущего.
4. В открывшемся файле `mcp.json` добавьте серверу заголовок с токеном, как в примере ниже, и сохраните файл.

JSON

```

1  {
2    "inputs": [
3      {
4        "type": "promptString",
5        "id": "site-token",
6        "description": "Access-токен MCP-шлюза",
7        "password": true
8      }
9    ],
10   "servers": {
11     "site": {
12       "type": "http",
13       "url": "https://example.com/api/mcp",
14       "headers": {
15         "Authorization": "Bearer ${input:site-token}"
16       }
17     }
18   }
19 }
```

При первом запуске сервера VS Code запросит токен и сохранит его в защищённом хранилище, поэтому в файле токен не хранится.

Документация клиента: MCP servers in VS Code.

Cursor

Добавьте сервер в файл `~/.cursor/mcp.json` для всех проектов или в `.cursor/mcp.json` в корне проекта:

```
JSON
1  {
2    "mcpServers": {
3      "site": {
4        "url": "https://example.com/api/mcp",
5        "headers": {
6          "Authorization": "Bearer ${env:MCP_SITE_TOKEN}"
7        }
8      }
9    }
10 }
```

Токен берётся из переменной окружения `MCP_SITE_TOKEN`. Задайте её в профиле оболочки или в переменных окружения системы и перезапустите Cursor. Подключённые серверы отображаются на странице **Customize**.

Документация клиента: [Model Context Protocol в Cursor](#).

Командная строка

Этот способ подходит разработчикам. В примерах токен берётся из переменной окружения `MCP_SITE_TOKEN`: не записывайте его в файлы, которые попадают в репозиторий.

Claude Code

```
Bash
1  claude mcp add --transport http site https://example.com/api/mcp \
2  --header "Authorization: Bearer $MCP_SITE_TOKEN"
```

Оболочка подставит значение переменной при выполнении команды, и токен сохранится в личных настройках Claude Code `~/.claude.json` только для текущего проекта. Чтобы сервер был доступен во всех проектах, добавьте параметр `--scope user`.

Для общего доступа команды опишите сервер в файле `.mcp.json` в корне проекта. Токен в нём задаётся ссылкой на переменную окружения:

JSON

```

1  {
2    "mcpServers": {
3      "site": {
4        "type": "http",
5        "url": "https://example.com/api/mcp",
6        "headers": {
7          "Authorization": "Bearer ${MCP_SITE_TOKEN}"
8        }
9      }
10   }
11 }

```

Проверьте подключение командой `claude mcp list` или командой `/mcp` внутри сессии.

Документация клиента: [Connect Claude Code to tools via MCP](#).

Codex

Добавьте сервер в файл настроек `~/.codex/config.toml` :

TOML

```

1  [mcp_servers.site]
2  url = "https://example.com/api/mcp"
3  bearer_token_env_var = "MCP_SITE_TOKEN"

```

Codex читает токен из указанной переменной окружения и передаёт его в заголовке `Authorization: Bearer` . Настройки действуют и в приложении ChatGPT.

Документация клиента: [Model Context Protocol в Codex](#).

Gemini CLI

Bash

```

1  gemini mcp add --scope user --transport http \
2  --header "Authorization: Bearer $MCP_SITE_TOKEN" site https://example.com/api/mcp

```

С параметром `--scope user` настройки сохраняются в `~/.gemini/settings.json` . Без него сервер добавляется в файл проекта `.gemini/settings.json` вместе с токеном. Проверьте подключение командой `/mcp` внутри сессии.

Документация клиента: [MCP servers with Gemini CLI](#).

Ошибки подключения

Признак	Что проверить
Ответ 404	Адрес роута и путь API, включён ли MCP-шлюз
Ответ 401 или 403, клиент предлагает пройти авторизацию	Передаётся ли заголовок с токеном, не истёк ли токен, входит ли пользователь в группу «MCP: Клиенты»
Ошибка с кодом -32022	Поддерживает ли клиент MCP 2025-11-25 или 2026-07-28; при необходимости обновите клиент
Подключение есть, но список инструментов пуст	Секции mcp у роутов, группы пользователя в их whitelist; очистите кеш API-платформы

Если клиент не обновляет токен сам, после окончания срока укажите в его настройках новый access-токен и перезапустите клиент. Как получить токен — в разделе [«Доступ и группы»](#) → [«Токен»](#).

Инструкция для агента

Передайте агенту:

Текст

Подключись к MCP-шлюзу моего сайта по инструкции:
<https://webco.one/docs/bitrix/modules/native.mcp/agents/instruction/>

Агент спросит адрес сервера и токен и проверит настройки. Что нужно подготовить на сайте — в разделе «Ручное подключение» → «Подготовка на сайте».

Правила для агента

- Идите по разделам по порядку; к следующему — после подтверждения пользователя.
- Токен пользователь вводит в настройки клиента сам. Если без чата не обойтись, предупредите, что токен останется в истории.
- Не записывайте токен в URL и в файлы репозитория.
- Не подключайтесь от имени администратора сайта.
- Меняйте настройки клиента только с согласия пользователя.

Выбор клиента

Определите свой клиент. Если вы не можете изменить его настройки сами, их вносит пользователь по вашим указаниям.

Клиент	Настройка
Claude: веб-версия и приложения	Claude
Claude Desktop	Claude Desktop
ChatGPT: чат	Не подключается, используйте Codex: ChatGPT
Codex в приложении ChatGPT	ChatGPT
VS Code	VS Code
Cursor	Cursor
Claude Code	Claude Code

Codex CLI	<u>Codex</u>
Gemini CLI	<u>Gemini CLI</u>

Адрес сервера

Адрес сервера — адрес сайта, путь интерфейса и `/mcp`, например `https://example.com/api/mcp`. Спросите у пользователя:

1. Адрес сайта.
2. Путь интерфейса: «Сервисы → API-платформа → Настройки модуля», вкладка «Основные». См. «Основные настройки» → «Путь интерфейса».
3. Включён ли флажок MCP-шлюза на вкладке «Роуты». См. «Настройки роутов» → «MCP-шлюз».

Покажите собранный адрес и получите подтверждение.

Проверьте шлюз запросом `GET` без токена: `https://example.com/api/up/mcp`. Ответ 200 означает, что шлюз включён. Остальные ответы и их причины — в разделе «Сервисные роуты» → «GET up/mcp».

Если вместо JSON открылась страница сайта с ошибкой 404, путь интерфейса указан неверно.

Пользователь и токен

Попросите пользователя подтвердить:

1. Пользователь агента входит в группу «MCP: Клиенты» (NATIVE_MCP_CLIENT) и в группы нужных инструментов. См. «Доступ и группы».
2. У пользователя есть действующий access-токен. См. «Доступ и группы» → «Токен».

Подключение

Настройте клиент по его разделу, указав адрес сервера и access-токен. Если настройки меняете вы, берите токен из переменной окружения, например MCP_SITE_TOKEN.

Если клиенту нужен перезапуск, попросите пользователя после него прислать ссылку на эту страницу с текстом «Проверь подключение к МСР-шлюзу» и продолжайте с раздела «Проверка».

Проверка

Проверку выполняет агент сам, пользователю ничего открывать не нужно.

1. Найдите среди своих инструментов встроенные `project_info` и `module_info` .
Если их нет, сервер не подключился.
2. Вызовите `project_info` и сообщите пользователю название сайта из ответа.
Пользователь подтверждает, что это его сайт.
3. Если сервер не подключился или вызов вернул ошибку, разберите её по «Ручное подключение» → «Ошибки подключения».

Итог

Сообщите пользователю адрес сервера, доступные инструменты и дату окончания токена. Если доступны только встроенные инструменты, роуты сайта для агента не опубликованы: см. «Инструменты».

Инструменты

Инструмент — роут пользовательской карты API-платформы с типом `mcp` и секцией `mcp`. Отдельный код в MCP-шлюзе не нужен: при каждом запросе шлюз читает итоговые карты роутов.

Как добавить инструмент

1. Опишите роут в пользовательской карте роутов: адрес, HTTP-метод, параметры и контроллер.
2. Укажите тип `RouteConfig::ROUTE_TYPE_MCP`. У роута другого типа или без типа секция `mcp` игнорируется.
3. Задайте авторизацию `SecurityConfig::AUTH => SecurityConfig::TOKEN`. Тип токена не указывайте: по умолчанию используется access-токен.
4. Укажите группы в whitelist роута: «MCP: Клиенты» (`NATIVE_MCP_CLIENT`), если инструмент нужен всем агентам, или отдельную группу для части пользователей, см. «Доступ и группы».
5. Добавьте секцию `mcp` с кодом и описанием инструмента.
6. Очистите кеш API-платформы.

Примеры роутов и контроллеров — в разделе «Примеры инструментов»: «Собственный инструмент», «Название сайта», «Версия модуля» и «Список заказов».

Чтобы встроенный инструмент отдавал данные проекта, объявите в пользовательской карте роут с тем же методом и адресом, например `GET mcp/orders`, и укажите свой контроллер. Такой роут заменяет встроенный целиком, поэтому перенесите в него все настройки, включая параметры, группы и секцию `mcp`.

Роут без whitelist агент видит при входе в группу «MCP: Клиенты», но обычным REST-запросом его может вызвать любой пользователь с access-токеном. Поэтому группы указывают всегда.

Контроллер

Аргументы агента метод получает через `Context::getCurrent()->getParams()`: шлюз уже проверил их по схеме.

Шлюз не фильтрует результат и не требует для него схемы: состав ответа задаёт контроллер, а права на данные проверяет он же.

Секция mcp

Ключи секции — константы `Mcp\ToolConfig` API-платформы. Используйте их вместо строк: тот же контракт читает MCP-шлюз.

Константа	Ключ	Назначение
<code>ToolConfig::ROUTE_PUBLICATION</code>	<code>mcp</code>	Секция публикации роута как инструмента
<code>ToolConfig::CODE</code>	<code>code</code>	Код инструмента; агент получает его как имя инструмента
<code>ToolConfig::DESCRIPTION</code>	<code>description</code>	Описание для агента; если не задано, берётся описание роута
<code>ToolConfig::ANNOTATIONS</code>	<code>annotations</code>	Подсказки клиенту о характере операции

Код инструмента: от 1 до 128 символов, латинские буквы, цифры, `_`, `.` и `-`.

Коды должны быть уникальными среди всех опубликованных роутов: дубликат или недопустимый код делает недоступным весь каталог.

Описание роута `RouteConfig::DESCRIPTION` предназначено для людей и выводится в документации OpenAPI. Описание инструмента `ToolConfig::DESCRIPTION` читает модель: по нему агент решает, когда вызывать инструмент. Укажите в нём, что возвращает инструмент и в каких случаях его использовать.

Подсказки задаются булевыми значениями:

Константа	Подсказка	Смысл для клиента
<code>ToolConfig::ANNOTATION_READ_ONLY</code>	<code>readOnlyHint</code>	Инструмент только читает данные
<code>ToolConfig::ANNOTATION_DESTRUCTIVE</code>	<code>destructiveHint</code>	Инструмент может удалять или перезаписывать данные
<code>ToolConfig::ANNOTATION_IDEMPOTENT</code>	<code>idempotentHint</code>	Повторный вызов не меняет результат
<code>ToolConfig::ANNOTATION_OPEN_WORLD</code>	<code>openWorldHint</code>	Инструмент обращается к внешним системам

Клиент может использовать подсказки, например, чтобы запросить подтверждение. Они не заменяют проверок в контроллере.

Какие роуты публикуются

Инструментом становится роут, у которого выполнены все условия:

- тип `RouteConfig::ROUTE_TYPE_MCP` ;
- непустая секция `mcp` с кодом `ToolConfig::CODE` ;
- роут активен;
- авторизация `SecurityConfig::AUTH => SecurityConfig::TOKEN` по access-токену.

Сверх этих условий шлюз никогда не публикует сам роут `/mcp` и роуты выдачи токенов.

Если у роута с секцией `mcp` другая авторизация, инструмент в каталог не попадёт, а на вызов по коду шлюз ответит ошибкой с указанием требуемой авторизации, а не «инструмент не найден».

Отдельного флага активности у секции `mcp` нет. При `RouteConfig::ACTIVE => false` роут отвечает 403 и пропадает из каталога. Чтобы снять инструмент с публикации, оставив REST-роут рабочим, уберите секцию.

Каталог учитывает пользовательское переопределение карт и текущие группы пользователя.

Аргументы инструмента

Схему аргументов шлюз строит из параметров роута:

- параметры адреса, например `{status}` , обязательны;
- типы `string`, `integer` и `array` переносятся в схему, вложенные параметры — во вложенные объекты и массивы;
- обязательная строка не принимает пустое значение, обязательная коллекция — пустой массив или объект;
- допустимые значения и диапазоны переносятся в ограничения схемы.

Перед выполнением аргументы проверяются по схеме. Затем роут выполняется общим конвейером API-платформы с исходным HTTP-методом: проверки доступа, лимиты, события и кеш результата сохраняются.

Встроенные инструменты

Инструмент	Роут	Группа	Результат
project_info	GET /mcp/project	«МСП: Клиенты»	Название сайта
module_info	GET /mcp/module	«МСП: Клиенты»	Код, версия и дата версии API-платформы
orders_list	GET /mcp/orders	«МСП: Заказы»	Количество заказов: демонстрационное число, учитывает период date_from–date_to

Все три инструмента только читают данные. Обе группы создаёт установщик шлюза, каждый инструмент разобран в разделе «Примеры инструментов».

Протокол JSON-RPC

Агент обращается к одному роуту — `POST /mcp` относительно пути API, например `/api/mcp`. `tools/list` и `tools/call` — методы JSON-RPC этого адреса, отдельные HTTP-маршруты для них не нужны. `GET /mcp` отвечает 405: отдельного SSE-потока нет.

Каждый запрос передаёт access-токен в заголовке `Authorization: Bearer`. Пользователь токена должен входить в группу «MCP: Клиенты».

Версии протокола

Версия	Начало работы и методы
2026-07-28	server/discover, ping, tools/list и tools/call; метаданные в каждом запросе
2025-11-25	initialize, затем ping, tools/list и tools/call; notifications/initialized/cancelled принимаются без фоновой отмены

Для версии 2026-07-28 каждый запрос передаёт заголовки MCP-Protocol-Version и Mcp-Method, а `params._meta` — версию протокола и объект clientCapabilities. clientInfo необязателен; если он передан, это объект со строковыми name и version. Для tools/call нужен также заголовок Mcp-Name, совпадающий с `params.name`.

Для версии 2025-11-25 первоначальный initialize может не содержать заголовка версии; он передаёт protocolVersion, capabilities и clientInfo. Последующие запросы указывают согласованную версию. Принятое уведомление возвращает 202 без тела.

Получение каталога

ACCESS_TOKEN замените реальным access-токеном. Не размещайте токен в публичных файлах и URL.

Bash

```

1  curl 'https://example.com/api/mcp' \
2    -H 'Authorization: Bearer ACCESS_TOKEN' \
3    -H 'Content-Type: application/json' \
4    -H 'Accept: application/json' \
5    -H 'MCP-Protocol-Version: 2026-07-28' \
6    -H 'Mcp-Method: tools/list' \
7    --data '{"jsonrpc":"2.0","id":1,"method":"tools/list","params":{"_meta":
{"io.modelcontextprotocol/protocolVersion":"2026-07-28","io.modelcontextprotocol/clientCapabilities":
{}}}}'

```

Инструменты сортируются по имени и выдаются страницами по 50. Следующая страница запрашивается с `params.cursor` из `nextCursor` предыдущего ответа. Курсор не даёт дополнительных прав: на каждом запросе снова учитываются текущие группы пользователя.

Вызов инструмента

Для вызова задайте заголовки `Mcp-Method: tools/call` и `Mcp-Name` с именем инструмента. Тело запроса:

JSON

```

1  {
2    "jsonrpc": "2.0",
3    "id": 2,
4    "method": "tools/call",
5    "params": {
6      "name": "project_info",
7      "arguments": {},
8      "_meta": {
9        "io.modelcontextprotocol/protocolVersion": "2026-07-28",
10       "io.modelcontextprotocol/clientCapabilities": {}
11     }
12   }
13 }

```

Результат контроллера возвращается в `result.structuredContent` и дублируется текстом JSON в `result.content`. При ошибке выполнения роута ответ содержит `result.isError = true` и безопасную детализацию ошибки API-платформы с исходным id запроса.

Ошибки

Ситуация	Ответ

Ошибка выполнения роута	<code>result.isError = true</code> с данными ошибки
Аргументы не соответствуют схеме инструмента	<code>result.isError = true</code> с кодом <code>PARAMETERS_INVALID</code>
Неизвестный или недоступный инструмент	<code>error</code> с кодом -32602
Неверные метаданные или параметры запроса	<code>error</code> с кодом -32602
Заголовки противоречат телу запроса	<code>error</code> с кодом -32020
Неподдерживаемая версия протокола	HTTP 400, код -32022, <code>data.supported</code> и <code>data.requested</code>
Нет токена, нет группы или шлюз выключен	Обычный ответ API-платформы без формата JSON-RPC

JSON-RPC batch-запросы не поддерживаются. Идентификатор запроса — строка или целое число; `null` не принимается. Запрос без `id` допускается только для уведомлений версии 2025-11-25.

Ограничения размера

Запрос ограничен 1 МиБ и глубиной JSON 32. Для результата действует бюджет 1 МиБ: сначала убирается дублирование текста и `structuredContent`, затем при необходимости данные заменяются сообщением с признаком `_meta["native.mcp/resultOmitted"] = true`. Признак успеха или ошибки инструмента при этом сохраняется.

Не повторяйте изменяющую операцию после `resultOmitted`: она уже выполнена, пропущен только её результат.

Собственный инструмент

Собственный инструмент — роут пользовательской карты API-платформы с типом `mcp` и секцией `mcp`.

Роут добавляется в файл GET-роутов карты, например в `local/php_interface/api/routes/GET/`, а класс контроллера — в каталог контроллеров проекта, например в `local/api/controllers/`.

Подробнее — в статье API-платформы [«Подключение карты»](#), требования к роуту инструмента — в разделе [«Инструменты»](#).

Пример ниже — инструмент `today`, который возвращает текущую дату сайта.

Роут в карте

PHP

```
1  'mcp/today' => [  
2      RouteConfig::DESCRIPTION => 'Текущая дата сайта',  
3      RouteConfig::ROUTE_TYPE => RouteConfig::ROUTE_TYPE_MCP,  
4      ControllerConfig::CONTROLLER => [DateController::class, 'today'],  
5      RouteConfig::FORMAT => TypeConfig::FORMAT_JSON,  
6      RouteConfig::SECURITY => [  
7          SecurityConfig::AUTH => SecurityConfig::TOKEN,  
8          SecurityConfig::GROUP => [SecurityConfig::WHITELIST => ['NATIVE_MCP_CLIENT']],  
9      ],  
10     RouteConfig::CACHE => [CacheConfig::TTL => 0],  
11     ToolConfig::ROUTE_PUBLICATION => [  
12         ToolConfig::CODE => 'today',  
13         ToolConfig::DESCRIPTION => 'Получить текущую дату сайта в формате YYYY-MM-DD',  
14         ToolConfig::ANNOTATIONS => [ToolConfig::ANNOTATION_READ_ONLY => true],  
15     ],  
16 ],
```

Контроллер

PHP

```
1  <?php
2
3  namespace Local\Api;
4
5  use Bitrix\Main\Type\Date;
6
7  final class DateController
8  {
9      public function today(): array
10     {
11         return ['date' => (new Date())->format('Y-m-d')];
12     }
13 }
```

Дата берётся по часовому поясу сервера сайта. Кеш роута отключён, иначе агент получал бы прошлую дату. Результат вызова:

JSON

```
1  {"date": "2026-10-25"}
```

Задача агенту

Текст

```
1  Какое сегодня число на сайте?
```

Агент вызовет `today` без аргументов и ответит значением `date` .

Название сайта

Встроенный инструмент `project_info` возвращает название сайта, к которому подключён агент. Аргументов у инструмента нет. Общие правила публикации — в разделе «Инструменты».

Роут в карте

PHP

```

1  'mcp/project' => [
2      RouteConfig::DESCRIPTION => '«MCP-шлюз»: сведения о сайте',
3      RouteConfig::ROUTE_TYPE => RouteConfig::ROUTE_TYPE_MCP,
4      ControllerConfig::CONTROLLER => [Info::class, 'project'],
5      RouteConfig::FORMAT => TypeConfig::FORMAT_JSON,
6      RouteConfig::SECURITY => [
7          SecurityConfig::AUTH => SecurityConfig::TOKEN,
8          SecurityConfig::GROUP => [SecurityConfig::WHITELIST => ['NATIVE_MCP_CLIENT']],
9      ],
10     RouteConfig::CACHE => [CacheConfig::TTL => 0],
11     ToolConfig::ROUTE_PUBLICATION => [
12         ToolConfig::CODE => 'project_info',
13         ToolConfig::DESCRIPTION => 'Получить название сайта, к которому подключён агент',
14         ToolConfig::ANNOTATIONS => [ToolConfig::ANNOTATION_READ_ONLY => true],
15     ],
16 ],

```

Контроллер

PHP

```

1  public function project(): array
2  {
3      $name = (string)Option::get('main', 'site_name');
4      if ($name === '') {
5          $site = SiteTable::getRow([
6              'select' => ['SITE_NAME', 'NAME'],
7              'filter' => ['=DEF' => 'Y'],
8          ]);
9          $name = (string)($site['SITE_NAME'] ?? '') ?: (string)($site['NAME'] ?? '');
10     }
11
12     return ['name' => $name];
13 }

```

Название берётся из настройки «Название сайта» главного модуля, а если она пустая — из названия сайта по умолчанию. Результат вызова:

JSON

```
1 {"name": "Интернет-магазин «Пример»"}
```

Задача агенту

Текст

```
1 Как называется сайт, к которому ты подключён?
```

Агент вызовет `project_info` без аргументов и ответит значением `name` .

Версия модуля

Встроенный инструмент `module_info` возвращает код, версию и дату версии API-платформы. Аргументов у инструмента нет. Общие правила публикации — в разделе [«Инструменты»](#).

Роут в карте

PHP

```

1  'mcp/module' => [
2      RouteConfig::DESCRIPTION => '«MCP-шлюз»: версия и дата версии модуля «API-платформа」',
3      RouteConfig::ROUTE_TYPE => RouteConfig::ROUTE_TYPE_MCP,
4      ControllerConfig::CONTROLLER => [Info::class, 'module'],
5      RouteConfig::FORMAT => TypeConfig::FORMAT_JSON,
6      RouteConfig::SECURITY => [
7          SecurityConfig::AUTH => SecurityConfig::TOKEN,
8          SecurityConfig::GROUP => [SecurityConfig::WHITELIST => ['NATIVE_MCP_CLIENT']],
9      ],
10     RouteConfig::CACHE => [CacheConfig::TTL => 0],
11     ToolConfig::ROUTE_PUBLICATION => [
12         ToolConfig::CODE => 'module_info',
13         ToolConfig::DESCRIPTION => 'Получить версию и дату модуля «API-платформа» (native.api)',
14         ToolConfig::ANNOTATIONS => [ToolConfig::ANNOTATION_READ_ONLY => true],
15     ],
16 ],

```

Контроллер

PHP

```

1  public function module(): array
2  {
3      $date = ModuleApi::getVersionDate();
4
5      return [
6          'id' => ModuleApi::getModuleId(),
7          'version' => ModuleApi::getVersion(),
8          'version_date' => $date?->toString(),
9          'timestamp' => ['version_date' => $date?->getTimestamp()],
10     ];
11 }

```

Дата версии возвращается строкой для модели и меткой времени во вложенном `timestamp` для вычислений. Если даты нет, оба значения равны `null`. Результат вызова:

JSON

```
1  {  
2    "id": "native.api",  
3    "version": "2.0.0",  
4    "version_date": "15.09.2026 20:30:00",  
5    "timestamp": {"version_date": 1789504200}  
6  }
```

Задача агенту

Текст

1 Какая версия API-платформы установлена на сайте?

Агент вызовет `module_info` без аргументов и ответит версией и датой версии.

Список заказов

Встроенный инструмент `orders_list` демонстрационный: он не выбирает реальные заказы, а вычисляет число по текущей дате и необязательному периоду. Общие правила публикации — в разделе [«Инструменты»](#).

Роут в карте

PHP

```

1  'mcp/orders' => [
2      RouteConfig::DESCRIPTION => '«MCP-шлюз»: список заказов сайта',
3      RouteConfig::ROUTE_TYPE => RouteConfig::ROUTE_TYPE_MCP,
4      ControllerConfig::CONTROLLER => [Order::class, 'list'],
5      RouteConfig::FORMAT => TypeConfig::FORMAT_JSON,
6      RouteConfig::PARAMETERS => [
7          'date_from' => [
8              ParameterConfig::REQUIRED => false,
9              ParameterConfig::TYPE => TypeConfig::STRING,
10             RouteConfig::DESCRIPTION => 'Начало периода в формате YYYY-MM-DD, включительно.
Передаётся вместе с date_to',
11             ParameterConfig::EXAMPLE => '2026-10-01',
12         ],
13         'date_to' => [
14             ParameterConfig::REQUIRED => false,
15             ParameterConfig::TYPE => TypeConfig::STRING,
16             RouteConfig::DESCRIPTION => 'Конец периода в формате YYYY-MM-DD, включительно, не
раньше date_from. Передаётся вместе с date_from',
17             ParameterConfig::EXAMPLE => '2026-10-10',
18         ],
19     ],
20     RouteConfig::SECURITY => [
21         SecurityConfig::AUTH => SecurityConfig::TOKEN,
22         SecurityConfig::GROUP => [SecurityConfig::WHITELIST => ['NATIVE_MCP_ORDER']],
23     ],
24     RouteConfig::CACHE => [CacheConfig::TTL => 0],
25     ToolConfig::ROUTE_PUBLICATION => [
26         ToolConfig::CODE => 'orders_list',
27         ToolConfig::DESCRIPTION => 'Демонстрационный инструмент: возвращает количество заказов
без выборки реальных данных. Необязательный период date_from-date_to (YYYY-MM-DD, обе даты включительно,
передаются вместе) увеличивает результат на число дней периода',
28         ToolConfig::ANNOTATIONS => [ToolConfig::ANNOTATION_READ_ONLY => true],
29     ],
30 ],

```

Параметры `date_from` и `date_to` становятся аргументами инструмента, их описания читает модель.

Контроллер

PHP

```

1 public function list(): array
2 {
3     return ['count' => (int)date('j') + $this->getPeriodDays(Context::getCurrent()-
>getParams())];
4 }

```

Без периода контроллер возвращает текущий день месяца, с периодом добавляет к нему число дней периода, считая обе даты:

Аргументы (вызов 25-го числа)	Результат
<code>{}</code>	<code>{"count": 25}</code>
<code>{"date_from": "2026-10-01", "date_to": "2026-10-10"}</code>	<code>{"count": 35}</code>
<code>{"date_from": "2026-10-05", "date_to": "2026-10-05"}</code>	<code>{"count": 26}</code>

Формат и порядок дат проверяет контроллер. Ошибка возвращается как `PARAMETERS_INVALID` с указанием поля:

Ошибка	Данные ошибки
Передана только одна дата	<code>{"date_to": {"required": true}}</code>
Дата не в формате YYYY-MM-DD или не существует	<code>{"date_from": {"format": "YYYY-MM-DD"}}</code>
<code>date_to</code> раньше <code>date_from</code>	<code>{"date_to": {"min": "2026-10-10"}}</code>

Задача агенту

Агент сам переводит период из запроса в аргументы. Если год не назван, модель подставит его сама, обычно текущий. Относительные формулировки вроде «за прошлую неделю» модель пересчитывает в даты по своему представлению о текущей дате. Если результат важен, называйте даты.

Без периода

Текст

1 Сколько заказов на сайте?

Период

Текст

1 Сколько заказов с 1 по 10 октября 2026 года?

JSON

1 `{"date_from": "2026-10-01", "date_to": "2026-10-10"}`

Один день

Текст

1 Покажи количество заказов за 5 октября 2026 года.

JSON

1 `{"date_from": "2026-10-05", "date_to": "2026-10-05"}`

Используемые ресурсы

<https://claude.ai/customize/connectors>

<https://claude.ai/admin-settings/connectors>

<https://claude.com/docs/connectors/custom/add-unlisted>

<https://github.com/geelen/mcp-remote>

<https://learn.chatgpt.com/docs/extend/mcp>

<https://code.visualstudio.com/docs/copilot/customization/mcp-servers>

<https://cursor.com/docs/context/mcp>

<https://code.claude.com/docs/en/mcp>

<https://geminicli.com/docs/tools/mcp-server/>