Веб-сценарий

Веб-сценарий

Два разных действия:

Что делаете	Где	Для кого
Создать шаблон (цепочку шагов)	Каталог → `/templates/web-scenario` → **Создать шаблон**	админ / опытный
Создать проверку (сервис)	Мониторинг → Добавить → Веб-сценарий	оператор

Готовые шаблоны

В каталоге и в мастере сервиса уже есть популярные HTTP-сценарии:

Шаблон	Что проверяет
Сайт доступен	GET → HTTP 200
Публичная страница	GET → 200 + текст в теле
API health (JSON)	GET health → 200 + поле JSON
Вход и проверка	форма логина → 200 + маркер после входа
API с Bearer-токеном	GET с `Authorization: Bearer`
Страница с Basic Auth	GET с HTTP Basic
Отправка формы	GET формы → POST `x-www-form-urlencoded`
Переход по ссылке	найти `href` по regex → перейти
Редирект и финальная страница	ответ без follow → Location → 200
SSL-сертификат	TLS handshake, дни до истечения, доверие и совпадение имени

Пароли и токены в seed без значений по умолчанию — задаются при создании сервиса (поля «Настроить»).

Шаблон: мастер + редактор шагов

Как в Обнаружении, но **без графа**:

1. **С чего начать** — карточки готовых сценариев или пустой шаблон.

2. Или сразу **Перейти в опытный режим**.
3. Открывается **редактор на холсте** (тот же движок плиток/связей, что у сценариев обнаружения): добавьте шаги слева (вставляются после выбранного), оглавление внизу левой панели («Шагов / Развернуть») для навигации по длинной цепочке, порядок — «Раньше/Позже» или дублирование; исполнение остаётся **линейной** HTTP-цепочкой без ветвлений. Холст занимает оставшуюся высоту окна — счётчик шагов не прижат к рейке.
4. В настройках шаблона (шестерёнка) задаётся **интервал опроса** — не чаще **раз в минуту** (минимум 60 с; по умолчанию 60 с у HTTP-seed, у SSL — сутки).

JSON шагов — только в редакторе шаблона (редко, блок «Показать JSON шагов»). Обычная правка — на холсте блоками.

Сервис: короткий мастер

1. **Что проверяем** — выбрать шаблон из каталога (компактная сетка карточек).
2. **Адрес и доступ** — блок **Откуда проверять** (как у доступности услуги): **Система** / **Агент**; затем URL, логин/пароль (без `{$WS.*}` в подписях). Для **SSL-сертификата** — хост, порт, ожидаемое имя, интервал (часы), переключатели доверия/имени и пороги дней.
3. **Имя** — как проверка называется в списке.

На шагах **Адрес и доступ** и **Имя** баннер **Править шаги?** ведёт в редактор шаблона на холсте (как «Свой сценарий?» на первом шаге). Оператор **не** собирает цепочку при каждом сервисе: свой сценарий = сначала шаблон в каталоге.

Обзор сервиса

Карточка `/services/:id` (тип **Веб-сценарий**) в том же каркасе, что канал и доступность:

- шапка: имя, HTTP/SSL статус, источник проверки (**Система** / **Агент**), группа, теги;
- **Последняя проверка**: кнопка **Опросить сейчас** справа в заголовке; KPI как на узле — точка статуса, крупное значение, **кружок** доли успешных опросов и **мини-графики** времени/размера/кода. Для SSL — дни до окончания и статус сертификата;
- **Динамика** — общий график: время (общее, среднее и по шагам), размер ответа и код HTTP за час/сутки/неделю/месяц;
- **Шаги** — карточки на всю ширину блока, линии от середины карточки к середине следующей. На карточке: имя шага, код/URL/текст проверки (без повторяющейся фразы «проверьте ответ сервера»). Короткая цепочка — один ряд; если не влезает, шаги делятся поровну по рядам (8 ⟶ 4 и 4) и дальше идут змейкой, без горизонтального скролла. Для HTTP-шагов на обзоре видны последний код, размер и время. Для SSL — пороги, Subject/Issuer и проверки. Правка блоков — кнопка **Открыть на холсте** (редактор шаблона), не JSON на обзоре.

Пока нет точек опроса, KPI показывают «—» / «Ожидание». После создания проверка с сервера wiSLA выполняется **сразу в том же запросе** (ответ уже с `last_check`); мастер открывает обзор сервиса. Кнопка **Опросить сейчас** в блоке **Последняя проверка** запускает проверку вручную (без ожидания интервала). При **сборе с агента** метрики пишутся под именем зонда — обзор читает этот бакет (иначе при системе графики есть, а с агента — пусто).

При **Добавить → Веб-сценарии** сначала выбор движка: **HTTP-цепочка** (шаблоны выше, платформа) или **Браузер** (goto/click/fill/wait на зонде с пакетом `wisla-browser-runtime`).

Подробнее: режим Браузер.

Когда какой тип

Тип	Когда использовать
Доступность услуги	Одна простая проверка (в т.ч. TLS availability)
Веб-сценарий	Несколько шагов подряд (открыть → войти → проверить текст) **или** контроль SSL-сертификата в том же разделе
Канал	Качество связи между двумя зондами

Что делает система

Цепочка HTTP-запросов с **системы** мониторинга **или** с выбранного **агента** (движок **http**); сценарий в **браузере** — только на зонде с addon (движок **browser**: goto/click/fill). Шаг **SSL-сертификат** всегда выполняется с системы (TLS handshake + NotAfter). item type BROWSER (export) по-прежнему не импортируется.

Для HTTP-цепочки «с агента» нужен агент ≥ **0.2.27** (роль `channel`). Пока агент старше — шаги остаются в WAITING.

Каждый HTTP-прогон пишет метрики `web_scenario.*` (как items у wiSLA web scenario):

Ключ	Что это
`web_scenario.ok`	Сценарий успешен (1/0)
`web_scenario.failed_step`	Номер упавшего шага (0 = нет)
`web_scenario.latency_ms`	Полное время прогона
`web_scenario.avg_latency_ms`	Среднее время HTTP-шагов
`web_scenario.size_bytes`	Суммарный размер тел ответов
`web_scenario.status`	Последний HTTP-код
`web_scenario.bps`	Скорость загрузки (бит/с)
`web_scenario.step.{n}.latency_ms` / `.status` / `.size_bytes`	Время, код и размер шага

На обзоре сервиса они сводятся в **один график** (время слева, размер справа, код ответа пунктиром).
