

GlitchTip: чтение событий после выпуска

Назначение: наблюдение известных issue после выпуска без повторного чтения всей истории. Проверено 22.10.2026 на <https://glitchtip.olvery.ru>, **GlitchTip 6.1.6**. Это сервисный рецепт, не изменение collector или сервера.

Поддержка установленного API

У следующих списков **нет серверных фильтров release и времени:**

- GET /api/0/issues/{issue_id}/events/;
- GET /api/0/projects/{organization_slug}/{project_slug}/events/.

Параметры release, query, start, end здесь не применяются. HTTP200, пустая выборка, наличие параметров в следующей Link-ссылке или совместимость SDK с Sentry не доказывают поддержку фильтра.

Рабочий путь — limit и штатный cursor из Link, локальные проверки release/времени. Не конструировать курсор самостоятельно. Для запроса достаточно существующего glitchtip-olv3729 через штатный helper; токен и полный ответ не выводить и не сохранять.

Короткий пример одного GET

Пример читает одну запись известной issue, выводит только HTTP-код, release, времена и наличие следующей страницы. Это не полный observation и не доказательство исправности приложения.

```
/home/olvr/bin/olvr-secret run glitchtip-olv3729 -- python3 - <<'PY'
import json, os, urllib.parse, urllib.request
url = "https://glitchtip.olvery.ru/api/0/issues/29879/events/"
url += "?" + urllib.parse.urlencode({"limit": 1})
req = urllib.request.Request(url, headers={
    "Authorization": "Bearer " + os.environ["OLVR_SECRET"]})
with urllib.request.urlopen(req, timeout=25) as r:
    rows = json.load(r)
    link = r.headers.get("Link", "")
    print("http", r.status, "rows", len(rows))
    for e in rows:
        tags = e.get("tags") or {}
        if isinstance(tags, list):
            tags = {t.get("key"): t.get("value") for t in tags}
        release = tags.get("release") or e.get("release")
        if isinstance(release, dict):
            release = release.get("version")
        print({"release": release,
            "dateReceived": e.get("dateReceived") or
```

```
e.get("date_received"),
    "dateCreated": e.get("dateCreated") or
e.get("date_created"))})
    print("has_next", any('rel="next"' in p and 'results="true"' in p
                          for p in link.split(","))
PY
```

Для обычного обхода используется `?limit=100`. `urllib.parse.urlencode` кодирует query-значения; литеральный плюс в 4.6.5+9716 должен стать %2B, а не пробелом. Это правило encoding, **не обещание наличия release-фильтра**.

Минимальный штатный обход

- Начать с issue events без самодельного cursor. Читать страницы вперёд по Link с `rel=«next»` и `results=«true»`; перед передачей Authorization проверять тот же HTTPS-host `glitchtip.olvery.ru` и API-путь. `limit=100` — размер страницы, не общий предел.
- Нормализовать `dateReceived` / `date_received` как UTC datetime. Это серверное поступление. `dateCreated` / `date_created` — время события от клиента; им нельзя заменять границу поступления.
- Локально сравнивать exact release, без prefix/substrings. Для семантики исходного PSTU observation: release равен `com.tpark_it.pstu@4.6.5+9716` **ИЛИ** app context содержит `app_identifier=com.tpark_it.pstu`, `app_version=4.6.5`, `app_build=9716`. Проверка только release уже этого исходного контракта. Версия/build без app identifier также недостаточны.
- Для результата «после выпуска» учитывать только события с `dateReceived >= 2026-10-22T12:17:07Z`. Проверять все события нужного временного диапазона, независимо от release.
- Списки этой версии упорядочены по -id UUIDv7, что соответствует невозрастанию серверного `dateReceived`. На странице, пересёкшей cutoff, обработать подходящие события; когда встретилось серверное время **строго меньше** cutoff, более старые страницы для post-cutoff результата не нужны. Равное cutoff время включается. При отсутствующем/неразбираемом времени или нарушенном порядке не объявлять раннюю остановку полным доказательством: сохранить gap.
- Если Link закончился раньше cutoff, завершена доступная выдача этого endpoint. Если нужен итог по всей сохранённой истории, а не только после cutoff, раннюю остановку по времени не применять.

Отсутствие событий с ошибкой не доказывает успешные сессии. List endpoint читает свой queryset; полноценность cold-storage-истории этим рецептом не обещается.

Почему фильтрам нельзя доверять

Минимальный read-only control на issue **29879**, 22.10.2026 16:20:17Z:

- `?limit=1` → HTTP200, 1 событие: release `com.tpark_it.pstu@4.6.1+7740`, received 2026-10-22T08:17:55.915Z;
- те же `limit=1` плюс release=`jfvY-no-such-release-2099`,

start=2099-01-01T00:00:00Z, end=2099-01-02T00:00:00Z, query=release:«j fVY-no-such-release-2099» timestamp:>2099-01-01 → HTTP200, **та же запись 2026**.

Идентичность сравнивалась в памяти; идентификаторы и пользовательский payload не сохранялись. В отличие от пустого ответа, непустой контрпример показывает игнорирование переданных ограничений; точный набор поддерживаемых параметров установлен исходником.

Версия, evidence и пересмотр

Фактический target: srv-glitchtip.olvr, glitchtip-web-1, glitchtip/glitchtip:6, running. GLITCHTIP_VERSION=6.1.6.

- Image ID:
sha256:8476bbdf291f946850fc6430d2cab62836918638ced34f4791593081bc067128;
- RepoDigest:
glitchtip/glitchtip@sha256:f45485d296697531464b4484ee9bb40a51a774df820f74128a2b2ae0c5a0b00f;
- /code/apps/issue_events/api/events.py:20–35, 77–85, 209–226: queryset ограничен доступом и issue/project; оба списка не принимают фильтры и сортируют по - id;
- SHA256 этого файла:
1eae9be2df11f27cc5bbc29c24a73dce7d875be1f23c22e030507672ecb1eaac;
- /code/apps/issue_events/schema.py:214–219: created→timestamp, received→received;
- /code/apps/issue_events/models.py:308–317: received извлекается из серверного UUIDv7 с миллисекундной точностью;
- /code/glitchtip/api/pagination.py: AsyncLinkHeaderPagination, limit/cursor и Link. Возвращаемый Link сохраняет исходные query-параметры, но это не означает их применение.

GET /api/0/ возвращает API version 0, не версию приложения.

Evidence: Card **jfVY**, receipts **OvtbEO**, **0r9VZC**, **XEu7hX**, **4XP9Pe**. Безопасные artifacts: /home/BVNq/work/jfVY/installed-api-contract.txt, filter-controls-safe.json и соответствующие диагностические скрипты. Источник исходного observation — **3kfw/0dgUUT**; его collector и принятое observation не менялись и не переаттестовывались.

После смены GlitchTip image/version сверять затронутый API-контракт заново. Доработка сервера, новый collector, runner или общий механизм фильтрации в этот рецепт не входят.

From:

<https://wiki.olvery.ru/> - **Olvery**

Permanent link:

<https://wiki.olvery.ru/doku.php?id=runbooks:glitchtip-event-observation>

Last update: **2026/10/22 16:29**

