

КОНТРАКТНИЙ ПІДХІД ДО ЗБЕРІГАННЯ ПОДІЙ ДЛЯ ВЕБ-ПЛАТФОРМИ ГЕНЕРАЦІЇ НАБОРІВ ДАНИХ

Рубаненко М.С.

e-mail: mariia.rubanenko@nure.ua

Науковий керівник – Широкопетлєва М.С.

Харківський національний університет радіоелектроніки, каф. ПІ,
м. Харків, Україна

Synthetic event data is used to test web analytics and to build training datasets without involving personal information. The diversity of event types and their automated generation increase the risk of schema drift, incomplete records, and inconsistencies at the session level. This paper proposes a contract-based approach to event storage: a data contract specifies the event structure, required attributes, and versioning in accordance with the ODCS. The implementation combines a relational store with a shared core of attributes and a JSON field for event-specific parameters linked to a schema version, along with pre-ingestion validation and enforcement of session invariants. As a result, the system produces a consistent event stream suitable for analytics and controlled schema evolution.

Подієва веб-аналітика ґрунтується на потоках клієнтських подій, що надходять на сервер через API та використовуються для аналізу поведінки користувачів, продуктової й маркетингової аналітики [1]. Для веб-платформ характерне різноманіття типів подій із різними наборами атрибутів і контекстних параметрів, що ускладнює уніфіковане зберігання та керований розвиток формату. Придатність таких даних для аналітики залежить від стабільності структури й семантики, наявності кореляційних ідентифікаторів та узгодженості правил інтерпретації [2]. В умовах агентної симуляції події формуються автоматично клієнтською логікою, що підвищує ризики дрейфу структури, неповних повідомлень і порушення узгодженості сесій. Тому ключовим завданням є контрактний підхід до зберігання подій: контракт даних фіксує формат, правила узгодження та вимоги до контролю якості під час запису у сховище [3]. У практиці синтетичної генерації веб-подій найчастіше розрізняють формування подій на стороні сервера та формування подій на стороні клієнта з передаванням через API, що відрізняються рівнем контролю над схемою та близькістю до реального веб-трекінгу. Генерація подій на стороні сервера забезпечує централізований контроль схеми та знижує ризики дрейфу, однак має нижчу реалістичність порівняно з реальним веб-трекінгом. Формування подій на стороні клієнта найближче відтворює реальний веб-трекінг і добре сумісне з аналітичною обробкою, проте є чутливим до дрейфу схеми, пропусків обов'язкових полів і неузгоджених ідентифікаторів, що спотворює результати аналізу. Як альтернативний

підхід до генерації даних застосовується таблична синтетика (SDV), однак вона обмежено моделює події послідовності та сесійні залежності, критичні для поведінкового аналізу [4].

Метою роботи є проектування контрактної моделі зберігання подій із контролем інваріантів сесій (структура, семантика, цілісність) для формування узгодженого подієвого потоку, придатного для аналітичної обробки. Ключовою проблемою веб-платформи генерації наборів даних є різноманітність подій: різні типи поєднують спільне ядро атрибутів і специфічні параметри, що потребує керованого формату зберігання та еволюції схеми. Збір та агрегація великих масивів подій клієнтського відстеження вимагає гнучких моделей зберігання та підтримки міграції даних [5]. Вимоги до якості й придатності даних задаються моделлю ISO/IEC 25012:2008 [2], а їх реалізація забезпечується специфікацією контракту відповідно до Open Data Contract Standard [3]. Пропонується використовувати контрактну модель зберігання подій, у якій контракт даних фіксує структуру подій, обов'язкові атрибути, правила узгодження та принципи версіонування для керованої еволюції схеми [3]. Такий підхід забезпечує стабільність формату подій і підтримує вимоги до якості даних та їх придатності до використання в аналітиці [2]. Події зберігаються в реляційному сховищі PostgreSQL у вигляді пов'язаних сесій і подій. Використання реляційної бази даних є доцільним через підтримку обмежень цілісності, зв'язків між сутностями та транзакційного запису, що зменшує ризики частково збережених або суперечливих даних. Водночас платформа генерації передбачає велику кількість типів подій, для яких набір атрибутів відрізняється: окрім спільних полів (ідентифікатор сесії, тип події, часова позначка, номер у послідовності), окремі події мають специфічні параметри, релевантні лише для конкретного типу. Щоб не ускладнювати схему великою кількістю маловживаних колонок і водночас зберегти розширюваність, специфічні атрибути подій зберігаються у JSON-полі як структурований набір властивостей із прив'язкою до версії схеми [3]. Наявність версії схеми дозволяє керовано змінювати структуру специфічних атрибутів без порушення сумісності базових полів, на які спираються аналітичні розрахунки. У разі додавання нових параметрів або типів подій зміни локалізуються в JSON-частині, тоді як реляційне ядро залишається стабільним. Це зменшує ризик неконтрольованого розростання таблиці та підтримує прозору еволюцію формату подій у часі. Контракт примусово виконується на етапі приймання подій серверним інтерфейсом, який здійснює їх перевірку і нормалізацію до узгодженого формату збереження. У YAML-файлі контракту фіксуються версія схеми, типи подій, обов'язкові атрибути, типи полів і базові правила валідації для окремих подій. На основі цього опису серверний компонент виконує структурну та семантичну перевірку вхідних повідомлень. Контракт також визначає інваріанти сесії, тобто

умови цілісності, яким має відповідати сукупність подій у межах однієї сесії, зокрема наявність ідентифікатора сесії, коректність її початку та завершення, допустимий порядок подій і обмеження на дублювання. Контроль дотримання цих умов реалізується в серверній логіці обробки подій. Події, що порушують контракт, не передаються до основного аналітичного потоку, а зберігаються в окремій таблиці для аудиту та аналізу відхилень. У роботі спроектовано контрактну модель зберігання подій для веб-платформи генерації наборів даних, що забезпечує кероване зберігання різнорідних типів подій і контроль еволюції формату [3]. Рішення поєднує реляційне сховище зі стабільним ядром атрибутів і збереженням специфічних параметрів подій у JSON-полі з прив'язкою до версії схеми, а також перевірки перед записом і контроль інваріантів сесій. Запропонована модель підвищує узгодженість подієвого потоку та його придатність для аналітичної обробки і формування датасетів. Подальший розвиток передбачає формалізацію правил сумісності версій і автоматизацію перевірок інваріантів та якості даних.

Список використаних джерел:

1. Google. Send events (GA4) // Google Analytics for Developers (Measurement Protocol). URL: https://developers.google.com/analytics/devguides/collection/protocol/ga4/reference?client_type=firebase (дата звернення: 16.02.2026).
2. ISO/IEC 25012:2008. Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Data quality model. URL: <https://iso25000.com/index.php/en/iso-25000-standards/iso-25012> (дата звернення: 10.03.2026).
3. Bitol. Open Data Contract Standard (ODCS), v3.0.2: Specification. URL: <https://bitol-io.github.io/open-data-contract-standard/v3.0.2/> (дата звернення: 03.03.2026).
4. Patki N., Wedge R., Veeramachaneni K. The Synthetic Data Vault. – 2016. URL: <https://dai.lids.mit.edu/wp-content/uploads/2018/03/SDV.pdf> (дата звернення: 01.02.2026).
5. Peretiatko M., Shirokopetleva M., Lesna N. Research of Methods to Support Data Migration Between Relational and Document Data Storage Models. Innovative Technologies and Scientific Solutions for Industries. 2022. No. 2 (20). P. 64–74. URL: <https://doi.org/10.30837/itssi.2022.20.064> (дата звернення: 04.03.2026).