Event-Sourcing

¿Qué es event-sourcing y por qué debería preocuparme?

Event sourcing es un patrón de arquitectura que define un enfoque para las operaciones de manipulación de datos. Garantiza que cada cambio de estado de una aplicación se almacene en un evento. Con esto lo que conseguimos es poder ir hacia atrás y ver los pasos que hemos seguido, así como saber cómo hemos llegado a ese punto.

Event sourcing nos permite llevar un registro de los eventos y poder reconstruir estados pasados y ver que es lo causante de ese resultado, así como revertirlos si fuera necesario.

¿Por qué deberíamos darle una oportunidad?, porque nos ahorraríamos tiempo en caso de pérdida de datos, es decir; si pierdes la información de la base de datos que tienes almacenada serviría con ir la instantánea anterior y recuperaríamos la información perdida

¿Cómo funciona event-sourcing?

La idea principal de cómo funciona este patrón es la siguiente:

El objeto evento es aquel objeto que almacena el estado de la aplicación en un momento dado. Los objetos se almacenan en el orden en el que han sido capturados (como si hubiéramos hecho una foto en ese instante a la aplicación.) Esto nos permite realizar consultas temporales sí como revertir cambios que hemos hecho.

Los eventos se almacenan en el denominado **events long** (almacen de eventos), a este se puede acceder en cualquier momento para acceder al estado de la aplicación en el momento que queramos.

¿Ejemplo de event-sourcing?

Pongamos una empresa que se dedica al seguimiento de barcos mercantiles, y que desea notificar que uno está en movimiento en un cierto puerto. Cuando un barco entra o sale de un puerto, se llamaría a un service, el cual graba la posición actual. Con event sourcing, el servicio crearía un evento que almacena el cambio del barco y lo procesa para actualizar su posición. Sin el event sourcing, si u barco hubiese estado en más de 1 puerto no podríamos averiguarlo, puesto que solo guardaría el ultimo que visito. Sin embargo, con event sourcing, tendríamos capturado cada evento que sucedió, permitiéndonos ver los cambios de localización del navio.

Event-sourcing y el IoT.

En un sistema de IoT los sensores actuan como los generadores de evento y los actuadores como los consumidores de eventos. Los libros de eventos (Event ledgers) son una gran

herramienta que nos permite comprobar que los sistemas estan conectados y funcionan correctamente, y son usados por muchos ecosistemas de IoT. Otra forma que se comprueba el event sourcing es con CRFTs (tipos de datos convergentes), que permiten la convergencia de datos entre dispositicos conectados ocasionalmente.

Event-sourcing y las arquitecturas sin servidor:

La computación sin servidor es un modelo de ejecución de computación en la nube, en el cual la máquina pide recursos al proveedor de la nube y este le concede los que necesite, cuidando los servidores en nombre de sus clientes. Este tipo de computación es fundamentalmente dirigida por eventos.

Hay dos formas de lanzar una función sin servidor, directamente con una invocación o indirectamente con algo que ocurra en el entorno, a esto último se le llaman triggers (disparadores). Una vez que estás cómodo trabajando con una metodología de eventos disparados, lo lógico es buscar una forma de rastrear, mantener y procesar esos eventos de una manera confiable y sólida, y los libros de eventos (Event ledgers) pueden proporcionárnoslo.

Ventajas y Desventajas:

Entre los beneficios de utilizar una arquitectura basada en eventos se encuentran:

Incremento de la calidad de los datos de Machine Learning y análisis.

La arquitectura se asemeja más a la del mundo real, lo que hace que sea más comprensible y esté alineada con el resto.

El sistema tendrá una reacción más rápida a lo que ocurra en cualquier parte del mismo y será una reacción más sencilla.

Escalabilidad, el sistema puede escalar rápidamente a cientos de miles de eventos por segundo.

Suministro de servicios, el fuerte desacoplamiento reduce el tiempo de lanzamiento de nuevos servicios o cambios en los antiguos.

Los eventos no dependen de detalles técnicos de sus sistemas fuente y son entendibles por la gente de negocios.

Permite recuperarse de errores y volver a estados anteriores.

Desventajas:

La principal desventaja de este patrón es su dificultad. Aunque normalmente en internet event-sourcing se plantee como la solución a todos los problemas, la realidad es que no es una solución sencilla de implementar.

El coste y esfuerzo al comienzo son elevados, lleva bastante tiempo integrarlo en el sistema.

Además, en este patrón el front-end se vuelve totalmente independiente del back-end.

Referencias:

https://eamodeorubio.wordpress.com/2012/09/17/cqrs-3-event-sourcing/

https://medium.com/tech-at-nordstrom/adventures-in-event-sourced-architecture-part-1-cc21d06187c7

https://martinfowler.com/eaaDev/EventSourcing.html

https://chriskiehl.com/article/event-sourcing-is-hard

https://our-academy.org/posts/eventos-como-mecanismo-de-almacenamiento