

Software Design Patterns

Event Sourcing Pattern & Retry Pattern

Event Sourcing Pattern

Εισαγωγή

- Ένας διαφορετικός τρόπος αποθήκευσης δεδομένων από αυτόν που έχουμε συνηθίσει.
- Απλούστευση πολύπλοκων διασυνδέσεων.
- Βελτίωση απόδοσης, δυνατότητα κλιμάκωσης και ταχύτητα ανταπόκρισης.
- Μέχρι τώρα σε ένα ηλεκτρονικό μαγαζί οι αλλαγές στο εμπόρευμα σε μια βάση δεδομένων σε σχέση με την αποθήκη:

Sku	Quantity	Last Received	Last Shipped
DER565	15	2021-1-25	Null
YYZ232	27	2021-3-18	2020-3-30
SDS106	9	2020-6-01	2021-1-02

Event Sourcing Pattern

Παράδειγμα e-shop: Παραδοσιακός τρόπος

- Καταγραφή της τωρινής κατάστασης του stock με επαναλαμβανόμενους ελέγχους και αλλαγές στο σύστημα.
- Αλλαγή της ποσότητας στο DER565 καθώς λάβαμε 15 τεμάχια. Λίγες ώρες αργότερα συνειδητοποιεί ένας υπάλληλος πως 2 από αυτά είναι ανοιγμένα και τα στέλνει πίσω, 13. Ο υπεύθυνος πωλήσεων βρίσκει χαμένα στην αποθήκη άλλα 30 τεμάχια, 43. Τέλος, αγοράστηκαν 8 τεμάχια, 35.
- Create, Read, Update, Delete μοντέλο (CRUD model).
- Μειονεκτήματα: απόδοση, δυνατότητα κλιμάκωσης, προβλήματα ενημέρωσης δεδομένων από ταυτόχρονη αλλαγή δεδομένων, απώλεια δεδομένων.

Event Sourcing Pattern

Τι είναι ;

- Καταγραφή συμβάντων αντί για την τωρινή κατάσταση κάθε φορά.

Παραλαβή
Ποσότητα: 15, Ημερομηνία : 2020-01-20

Επιστροφή
Ποσότητα: 13, Ημερομηνία : 2020-01-23
Αιτιολογία: Ανοιγμένα προϊόντα

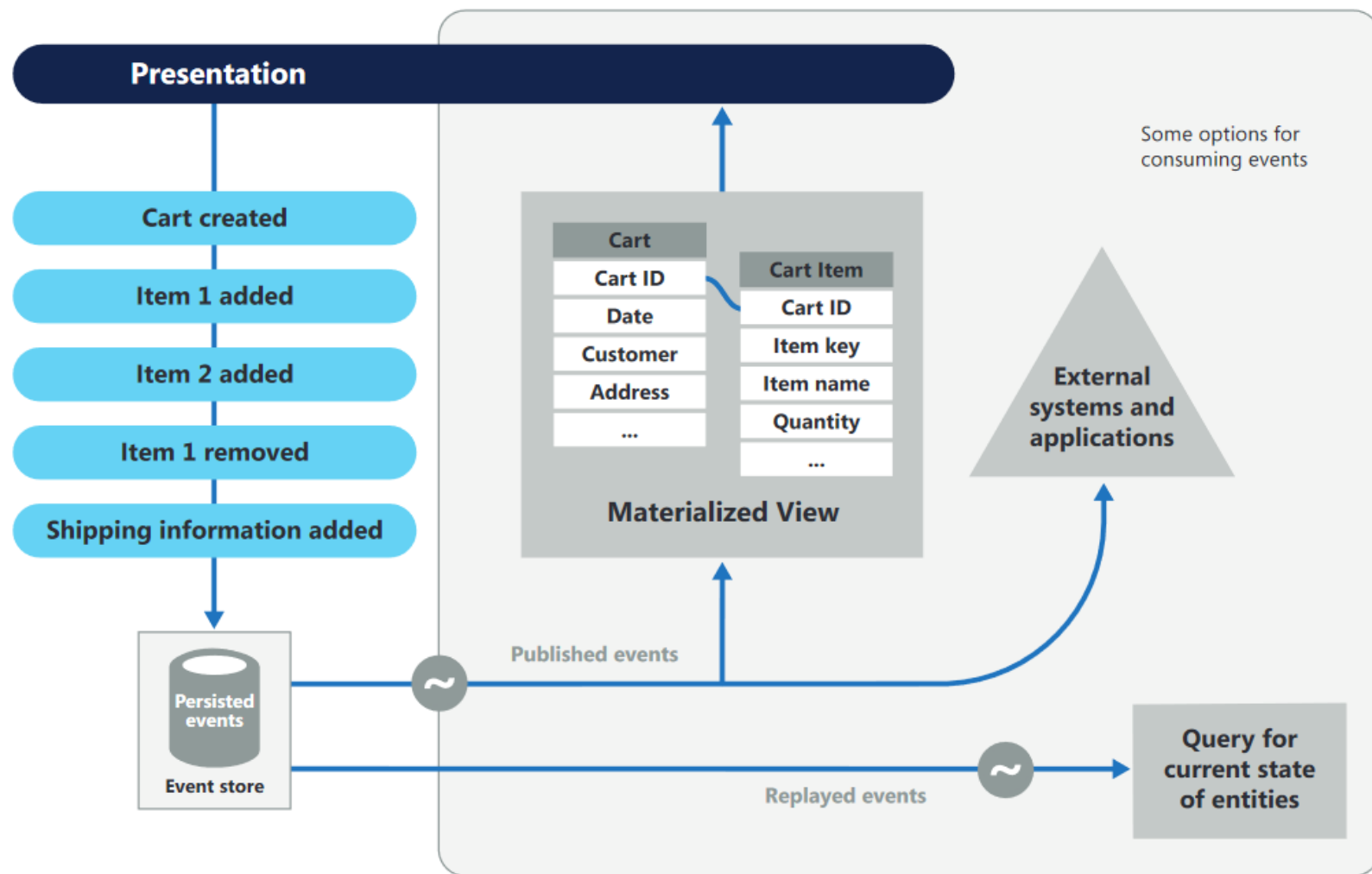
Αλλαγή
Ποσότητα: 43, Ημερομηνία : 2020-01-30
Αιτιολογία: Βρέθηκαν αποθήκη

Αποστολή
Ποσότητα: 35, Ημερομηνία: 2020-02-6

- Δεν χάνεται η πληροφορία του «γιατί» γίνεται η μετάβαση από μια κατάσταση σε μια άλλη.

Event Sourcing Pattern

To Event Sourcing Pattern με μια ματιά (Cloud Design Patterns by Microsoft)



Event Sourcing Pattern

Θετικά:

- Τα events είναι αμετάβλητες καταστάσεις: append-only λειτουργίες.
- Δεν ενημερώνουν άμεσα το πληροφοριακό σύστημα: καταγράφουν σειριακά τις καταστάσεις ώστε να ληφθούν υπόψιν αργότερα.
- Εξαγωγή χρήσιμων δεδομένων μέσα από τις λεπτομέρειες. π.χ. ένα στα τρία προϊόντα φθάνει ανοιγμένο στην αποθήκη. Χρήσιμη πληροφορία για έναν ειδικό.
- Αποφυγή σφαλμάτων κατά την ταυτόχρονη ενημέρωση μιας συγκεκριμένης πληροφορίας. π.χ. υπάλληλος front-desk με υπάλληλο αποθήκης (conflict).
- Εύρεση ενεργειών που έγιναν κατά της εφαρμογής καθώς το σύστημα είναι αυστηρά καταγραφής.
- Αναζήτηση χρήσιμων πληροφοριών: τάσεις της συμπεριφοράς των χρηστών και απόδοση της εφαρμογής.

Event Sourcing Pattern

Αρνητικά:

- **Σχετική καθυστέρηση:** μεταξύ της καταγραφή συμβάντων, της δημοσιοποίησης τους και της αντιμετώπισης τους από τους χρήστες.
- **Αμετάβλητη πηγή:** Η αποθήκευση των συμβάντων δεν μπορεί να μεταβληθεί. Μοναδικός τρόπος τροποποίησης ενός συμβάντος στο σύστημα είναι η προσθήκη ενός νέου event το οποίο αντισταθμίζει το λάθος συμβάν. π.χ. αφαίρεση 10 τεμαχίων ενός προϊόντος επειδή κατέφθασαν 20 τεμάχια στην αποθήκη αντί για 30 που καταγράφηκε στο event.
- **Πολύ γενική προσέγγιση:** Δεν υπάρχει συγκεκριμένος τρόπος υλοποίησης του μηχανισμού όπως π.χ. κάποια SQL queries για ανάγνωση των events. Τα μοναδικά δεδομένα που μπορούν να εξαχθούν είναι μια ροή συμβάντων με τη βοήθεια ενός αναγνωριστικού ως κριτήριο.

Retry Pattern

Εισαγωγή

- Δυνατότητα σε μια εφαρμογή να διαχειρίζεται αναμενόμενες προσωρινές αστοχίες όταν προσπαθεί να συνδεθεί σε μια υπηρεσία ή σε ένα δίκτυο.
- Επαναλαμβανόμενες προσπάθειες στο παρασκήνιο , οι οποίες νωρίτερα απέτυχαν, με την «ελπίδα» ότι ίσως το πρόβλημα που παρουσιάστηκε ήταν προσωρινό.
- Σταθερότητα

Retry Pattern

Ιδέα

- Μια εφαρμογή που επικοινωνεί με το cloud είναι ευαίσθητη σε προσωρινά σφάλματα που μπορεί να προκύψουν.
- Προσωρινή απώλεια δικτύου, η υπηρεσία βρίσκεται προσωρινά εκτός λειτουργίας ή timeout όταν μια υπηρεσία βρίσκεται κάτω από μεγάλη πίεση από requests.
- Εύκολο να αυτοδιορθωθούν τέτοιου είδους σφάλματα.
- Μετά από επαναπροσπάθεια της ενέργειας που προκάλεσε το σφάλμα είναι πολύ πιθανό να έχουμε επιτυχής σύνδεση.
- Η υλοποίηση ενός cloud-based project θα πρέπει να είναι σε θέση να διαχειρίζεται τέτοιου είδους σφάλματα έξυπνα ώστε να έχουν όσο το δυνατόν μικρότερο αντίκτυπο στις επιχειρηματικές διεργασίες.

Retry Pattern

Που χρησιμεύει;

- **Λάθος στοιχεία σύνδεσης:** στη περίπτωση αυτή είναι απίθανο να γίνει επιτυχής σύνδεση οπότε πρέπει να γίνει ακύρωση της διεργασίας (κλήση) και καταγραφή του συμβάντος σε κατάλληλο exception.
- **Σπάνιο είδος σφάλματος:** η εφαρμογή μπορεί να δοκιμάσει ξανά να συνδεθεί καθώς ένα πακέτο δικτύου μπορεί να έφτασε κατεστραμμένο ή ελλιπές. Τέτοιου είδους λάθη δύσκολο να επαναληφθούν και συνιστάτε η άμεση επαναπροσπάθεια.
- **Συχνό είδος σφάλματος/Busy:** εδώ η εφαρμογή θα χρειαστεί να περιμένει για μικρό χρονικό διάστημα πριν ξαναπροσπαθήσει. Ο διακομιστής στον οποίο προσπαθεί να συνδεθεί πολύ πιθανό να βρίσκεται εκτός λειτουργίας από μεγάλο αριθμό αιτημάτων.

Retry Pattern

Πως λειτουργεί;



- 1: Application invokes operation on hosted service. The request fails, and the service host responds with HTTP response code 500 (internal server error).
- 2: Application waits for a short interval and tries again. The request still fails with HTTP response code 500.
- 3: Application waits for a longer interval and tries again. The request succeeds with HTTP response code 200 (OK).

Retry Pattern

Σημαντικά σημεία

- Εάν ένα αίτημα εξακολουθεί να είναι ανεπιτυχής μετά από προκαθορισμένο αριθμών προσπαθειών τότε η εφαρμογή θα πρέπει να το διαχειριστεί αναλόγως.
- Η καταγραφή συχνών αποτυχιών ενός αιτήματος σε συγκεκριμένο διακομιστή μπορεί να σημαίνει ότι υπάρχει πολύ μεγάλη κίνηση αιτημάτων στη συγκεκριμένη δικτυακή περιοχή. Μια τέτοια πληροφορία είναι χρήσιμη για τον διαμοιρασμό των αιτημάτων σε τακτά χρονικά διαστήματα ή διαμοιρασμό σε διαφορετικούς διακομιστές αν αυτό είναι εφικτό.
- Είναι σημαντική η καταγραφή των προσπαθειών που αποτυγχάνουν καθώς έτσι μπορούμε να εντοπίσουν τυχόν προβλήματα με την εφαρμογή, τις υπηρεσίες ή τους πόρους της.