

# Konkurentni pristup resursima u bazi

Student 1

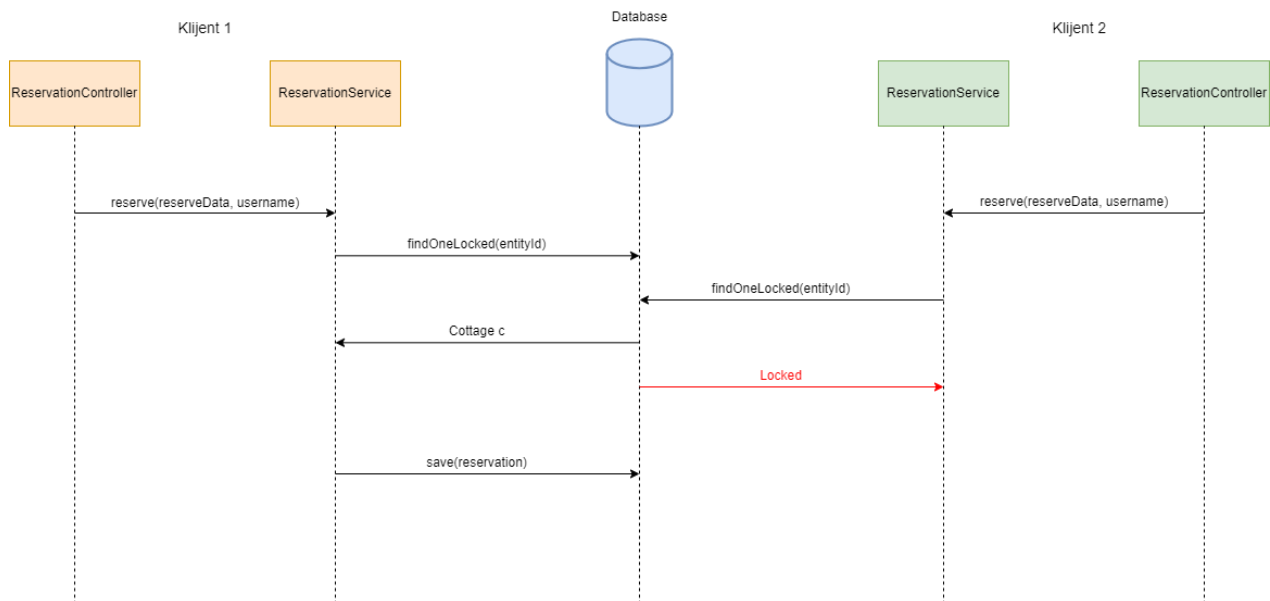
Ivana Janković SW5/2019

## 1. Konflikt – Više istovremenih klijenata ne mogu da naprave rezervaciju istog entiteta u isto (ili preklapajuće) vrijeme

Potencijalni konflikt nastaje kada više klijenata u isto vrijeme pokuša da rezerviše entitet u istom ili preklapajućem periodu. Ukoliko ne obezbijedimo mehanizam za zaključavanje entiteta, može nastati problem da oba korisnika uspješno rezervišu entitet za isti period.

Problem je riješen korištenjem pesimističkog zaključavanja, a postupak je ekvivalentan za sva tri tipa entiteta, pa ćemo demonstrirati samo slučaj rezervacije vikendice.

Klasa *ReservationService* je označena anotacijom *@Transactional*, a umjesto metode repozitorijuma *findOne*, korištena je metoda *findOneLocked* u *CottageRepository* i na njoj se koristi pesimističko zaključavanje. Na slici 1 je prikazan tok zahtjeva u slučaju kada dva klijenta istovremeno šalju zahtjev za rezervaciju entiteta. Klijent koji prvi pozove metodu *findOneLocked*, uspješno rezerviše vikendicu, dok drugi klijent dobija obavještenje da je entitet već rezervisan.



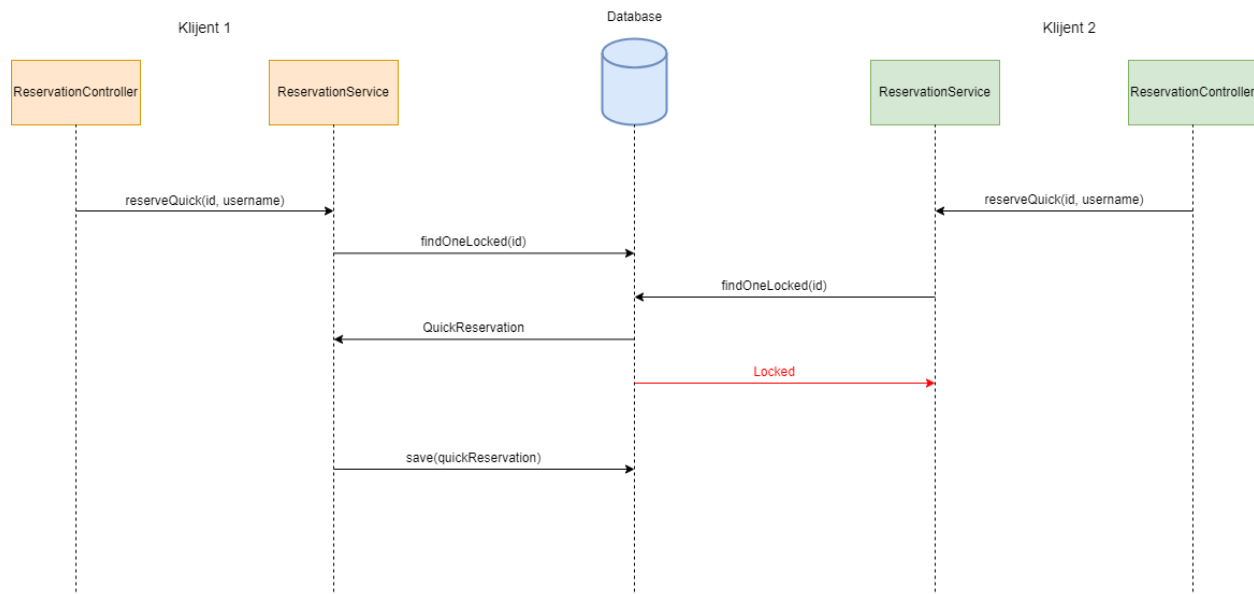
**Slika 1** – Klijenti 1 i 2 istovremeno šalju zahtjev za rezervaciju entiteta za isti period

## 2. Konflikt – Više istovremenih klijenata ne mogu da naprave rezervaciju istog entiteta na akciji u isto vrijeme

Potencijalni konflikt nastaje kada više klijenata u isto vrijeme pokuša da rezerviše entitet na akciji u istom periodu. Ukoliko ne obezbijedimo mehanizam za zaključavanje entiteta, može nastati problem da oba korisnika uspješno rezervišu istu akciju.

Problem je riješen korištenjem pesimističkog zaključavanja.

Klasa *ReservationService* je označena anotacijom *@Transactional* i korištena je metoda *findOneLocked* u *QuickReservationRepository* za koju se koristi pesimističko zaključavanje. Na slici 2 je prikazan tok zahtjeva u slučaju kada dva klijenta istovremeno šalju zahtjev za rezervaciju entiteta na akciji. Klijent koji prvi pozove metodu *findOneLocked*, uspješno rezerviše akciju, dok drugi klijent dobija obavještenje da je entitet već rezervisan.



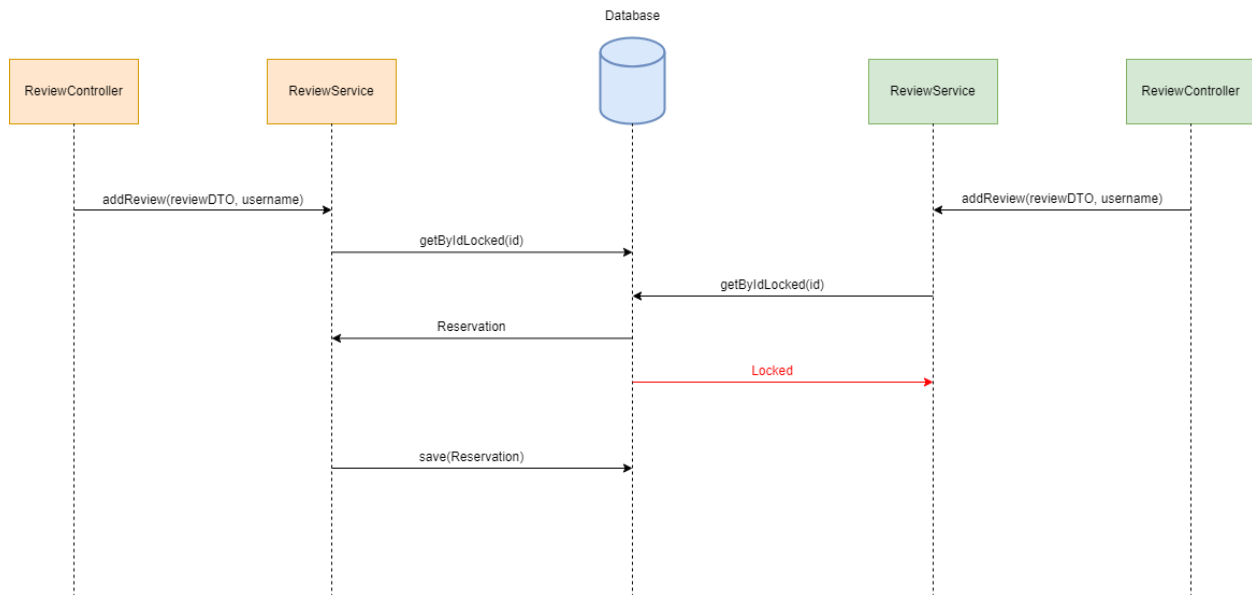
**Slika 2** – Klijenti 1 i 2 istovremeno šalju zahtjev za rezervaciju iste akcije

## 3. Konflikt – Za jednu rezervaciju se ne može poslati više revizija u isto vrijeme

Potencijalni konflikt nastaje kada više korisnika sa istog naloga u isto vrijeme pokušaju da pošalju reviziju za entitet i korisnika za istu rezervaciju.

Problem je riješen korištenjem pesimističkog zaključavanja.

Klasa *ReviewService* je označena anotacijom *@Transactional* i korištena je metoda *getByIdLocked* u *ReservationRepository* za koju se koristi pesimističko zaključavanje. Na slici 3 je prikazan tok zahtjeva u slučaju kada se istovremeno šalju dvije revizije za istu rezervaciju. Klijent koji prvi pozove metodu *getByIdLocked*, uspješno šalje reviziju, dok drugi klijent dobija obavještenje o konfliktu.



**Slika 3** – 2 korisnika istovremeno šalju zahtjev za reviziju iste rezervacije