

# Konkurentni pristup resursima u bazi

## student 2 - Zorica Vuković

Prvi problem: Vlasnik vikendice/broda ili instruktor ne može da napravi rezervaciju u isto vreme kad i drugi klijent. Ova situacija bi mogla biti realna u slučaju da istovremeno vlasnik vikendice/broda ili instruktor želi da zakaže rezervaciju klijentu koji ima aktivnu rezervaciju za neki period, a klijent želi da zakaže rezervaciju za isti taj entitet u preklapajućem periodu. Ovakva situacija bi dovela do toga da dve rezervacije budu kreirane za isti entitet, u preklapajućem periodu, različitim ili istim klijentima što može dovesti do problema u čitavom sistemu.

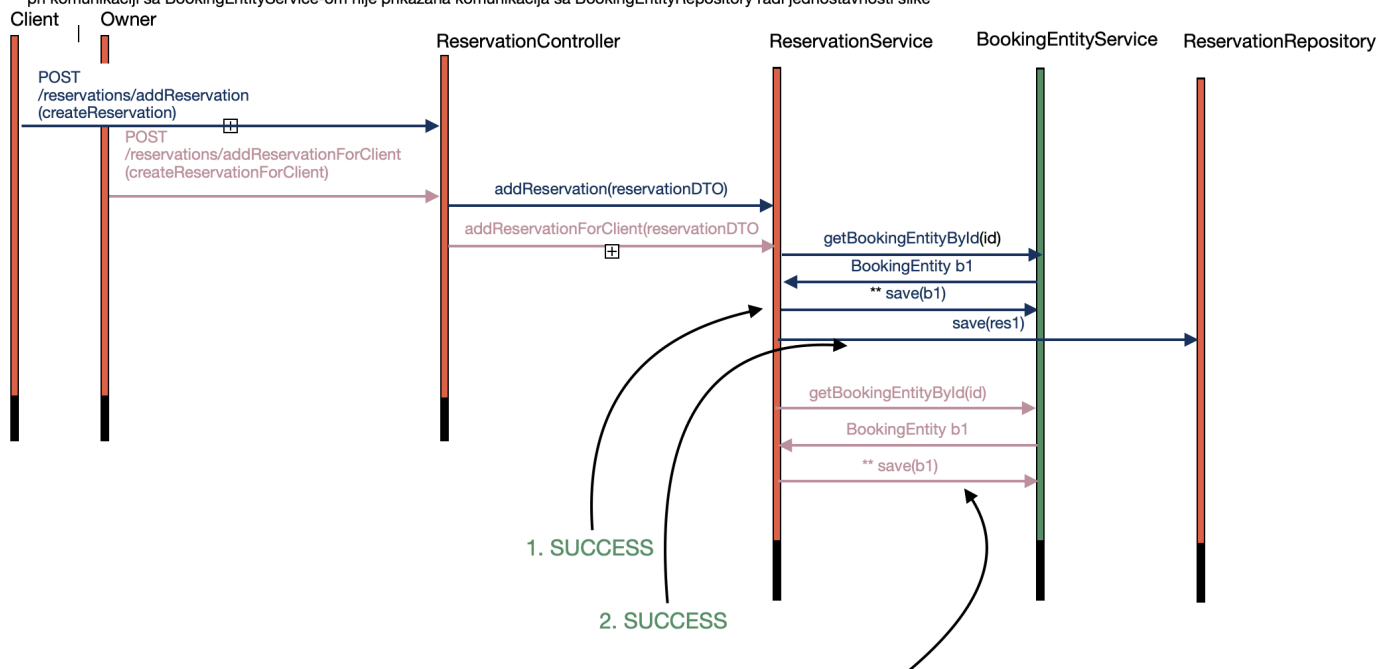
**Rešenje:** Kako bi sprečili da do ove situacije dođe, korist ćemo optimističko zaključavanje određenog entiteta koji rezervišemo u datom trenutku. Ovaj pristup je izabran jer entitet i kada zaljučamo, njega drugi mogu da dobavljaju radi čitanja pa samim tim to zaključavanje entiteta neće biti u većini slučajeva uopšte primetno. Optimističko zaključavanje je ovde bilo moguće jer bi do konfliktne situacije dolazilo kada se preklapaju datumi za **isti** entitet koji prethodno **postoji u bazi**, a upravo to nam omogućava da upravo taj entitet zaključamo sve dok se ne sačuva nova rezervacija u bazi podataka.

Kao što se vidi na slici ispod, pre čuvanja nove rezervacije u bazi podataka, menja se polje **locked** entiteta na **true** (na ovo polje je dodata anotacija **@Version**) i čuvanje te izmene u bazi će dovesti do automatskog inkrementiranja verzije tog entiteta. Zbog toga korisnik (klijent ili vlasnik) koji bude sporiji neće moći uspešno da izvrši rezervaciju, jer će u međuvremenu doći do promene verzije i hvata se **ObjectOptimisticLockingFailureExection** kao znak da ne radimo sa najnovijom verzijom entiteta (uopšte podataka).

```
BookingEntity entity = bookingEntityRepository.getEntityById(entityDTO.getId());

//resavanje konfliktne situacije student 2.
entity.setLocked(true);
bookingEntityRepository.save(entity);
res.setClient(client);
reservationRepository.save(res);
entity.setLocked(false);
bookingEntityRepository.save(entity);
```

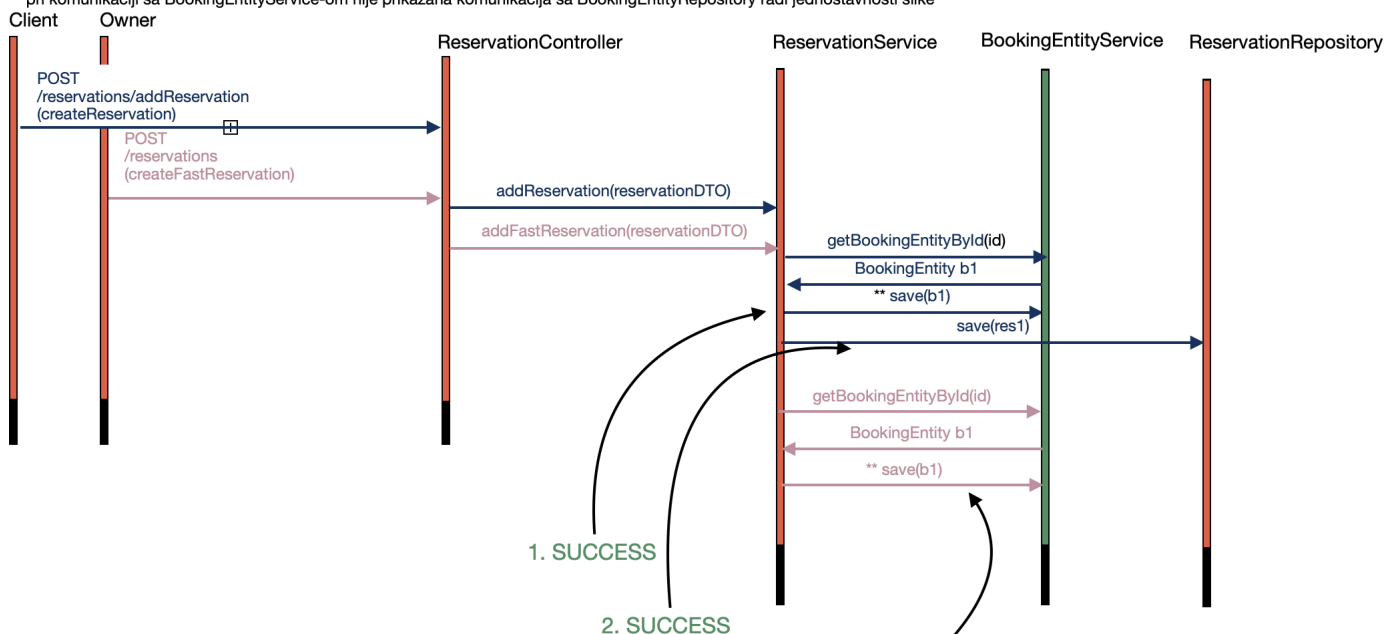
URL\_PATH = http://localhost:8092/bookingApp \*id = id od entiteta koji se rezerviše  
 \*\*pre čuvanja se izmeni polje locked kako bi se pri čuvanju izmene u bazi promenila verzija za taj entitet  
 \*\*\*pri komunikaciji sa BookingEntityService-om nije prikazana komunikacija sa BookingEntityRepository radi jednostavnosti slike



U ovom trenutku se dešava  
 ObjectOptimisticLockingFailureException  
 - konflikt se dešava jer se verzija za entitet b1 već promenila -

Drugi problem: Vlasnik vikendice/broda ili instruktor ne može da napravi akciju u isto vreme kad i drugi klijent vrši rezervaciju postojećeg entiteta. Ova situacija bi mogla biti realna u slučaju da istovremeno vlasnik vikendice/broda ili instruktor želi da kreira brzu rezervaciju (akciju) za neki period, a klijent želi da zakaže rezervaciju za isti taj entitet u preklapajućem periodu. Ovakva situacija bi dovela do toga da i akcija i rezervacije budu kreirane za isti entitet, u preklapajućem periodu, što može dovesti do problema u čitavom sistemu ukoliko izabrana akcija bude rezervisana od strane nekog klijenta.

URL\_PATH = http://localhost:8092/bookingApp \*id = id od entiteta koji se rezerviše  
 \*\*pre čuvanja se izmeni polje locked kako bi se pri čuvanju izmene u bazi promenila verzija za taj entitet  
 \*\*\*pri komunikaciji sa BookingEntityService-om nije prikazana komunikacija sa BookingEntityRepository radi jednostavnosti slike



U ovom trenutku se dešava  
 ObjectOptimisticLockingFailureException  
 - konflikt se dešava jer se verzija za entitet b1 već promenila -

**Treći problem:** Vlasnik vikendice/broda ili instruktor ne može da briše svoj entitet u isto vreme kada i klijent želi da rezerviše isti taj entitet. Ova situacija bi mogla biti realna u slučaju da istovremeno vlasnik vikendice/broda ili instruktor želi da obriše izabrani entitet, a klijent želi da zakaže rezervaciju za isti taj entitet. Ovakva situacija bi dovela do toga da u sistemu bude kreirana predstojeća rezervacija za entitet koji više ne postoji u sistemu, što može dovesti do problema u čitavom sistemu ukoliko klijent dođe u periodu rezeracije na željenu adresu, a taj entitet više nije deo sistema za izdavanje.

URL\_PATH = <http://localhost:8092/bookingApp> \*id = id od entiteta koji se rezerviše = entityId  
 \*\*pre čuvanja se izmeni polje lock kako bi se pri čuvanju izmene u bazi promenila verzija za taj entitet  
 \*\*\*pri komunikaciji sa BookingEntityService-om nije prikazana komunikacija sa BookingEntityRepository radi jednostavnosti slike

