

Konkurentnost - Student 2 Strahinja Popović – SW51-2019

Rešavane konfliktne situacije:

1. Više klijenata ne može istovremeno da napravi rezervaciju za isti entitet u preklapajuće vreme
2. Više klijenata ne može istovremeno da napravi akcijsku rezervaciju za isti entitet u preklapajuće vreme
3. DODATNA SITUACIJA: Ne može da se desi da klijent menja svoje podatke u isto vreme kad se vrši automatsko resetovanje penala svakog prvog dana u mesecu u ponoć

Rešenja:

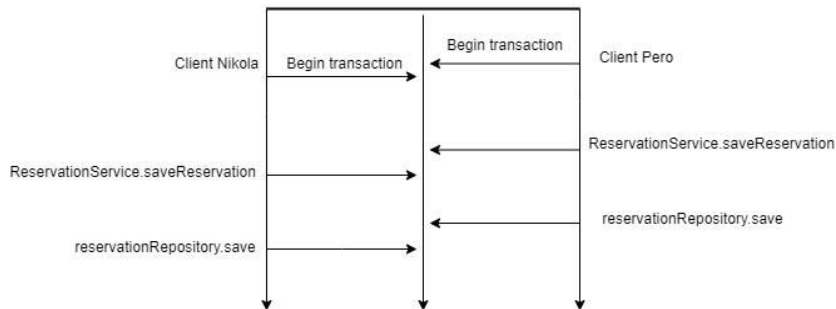
1. Više klijenata ne može istovremeno da napravi rezervaciju za isti entitet u preklapajuće vreme

Scenario problema: Dva ili više klijenata rezervišu isti entitet za isto ili preklapajuće vreme. Kako će oba korisnika imati isto stanje sistema pri procesu rezervacije desiće se da postoje dve ili više rezervacije za isti entitet koje se dešavaju u isto vreme što po specifikaciji nije dozvoljeno.

```
@PostMapping(value="/make-reservation")
@PreAuthorize("hasRole('CLIENT')")
@CrossOrigin(origins = ServerConfig.FRONTEND_ORIGIN)
public ResponseEntity<HttpStatus> makeReservation (@RequestBody
```



```
@PostMapping(value="/make-reservation")
@PreAuthorize("hasRole('CLIENT')")
@CrossOrigin(origins = ServerConfig.FRONTEND_ORIGIN)
public ResponseEntity<HttpStatus> makeReservation (@RequestBody
```



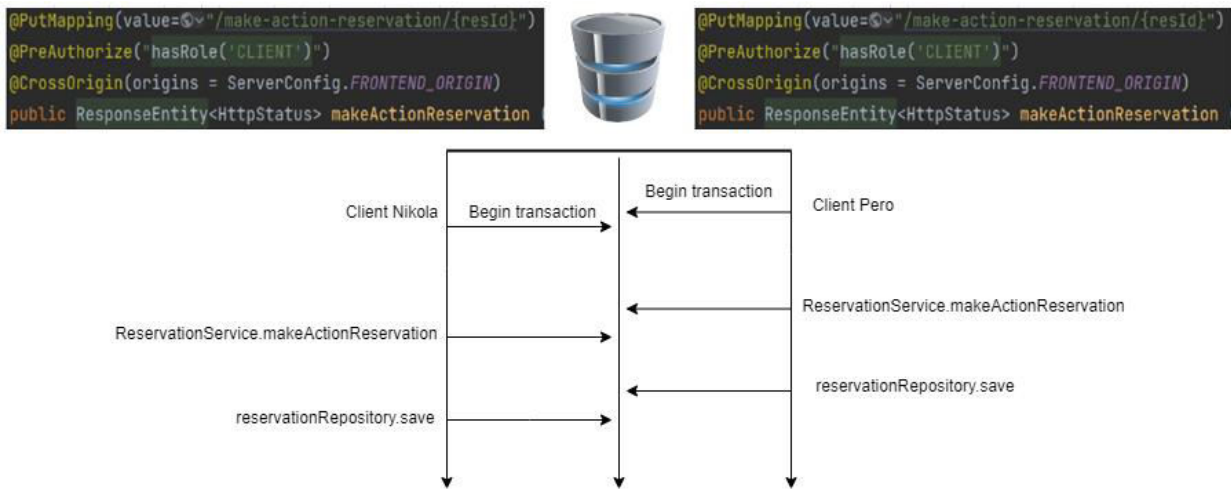
Rešenje problema: Problem je rešen korišćenjem optimističnog zaključavanja tako što je u klasu RentalService smo ubacili kolone *version* i *isChanged* i za svaku novu transakciju invertovali vrednost atributa *isChanged* i pokušavali cuvanje entiteta za koji se vrši rezervacija. Takođe sve servisne metode uključene u ovaj proces dobile su anotaciju *@Transactional*. Ovim se sprečava istovremena rezervacija istog entiteta od strane dva klijenta zbog različitih verzija entiteta kod svakog klijenta koji vrši rezervaciju, čime se rešava dati problem.

```
@Version
@Column(columnDefinition = "integer DEFAULT 0", nullable = false)
private Long version;

@Column(name = "is_changed", columnDefinition = "boolean DEFAULT false", nullable = false)
private boolean isChanged;
```

2. Više klijenata ne može istovremeno da napravi akcijsku rezervaciju za isti entitet u preklapajuće vreme

Scenario problema: Dva ili više klijenata pokušava da rezerviše istu akcijsku rezervaciju. Desiće se da će svi klijenti koji učestvuju u procesu da imaju isto stanje sistema u tom trenutku. Ovo rezultuje u tome da će svi ti klijenti biti u mogućnosti da naprave akcijsku rezervaciju. Ali samo jedan od njih, onaj koji poslednji bude radio upisivanje entiteta u bazu, će stvarno napraviti rezervaciju.



Rešenje problema: Rešenje ovog problema opisano je u rešenju problema pod brojem 1.

3. DODATNA SITUACIJA: Ne može da se desi da klijent menja svoje podatke u isto vreme kada se vrši automatsko resetovanje penala svakog prvog dana u mesecu u ponoć

Scenario problema: Klijent želi da izmeni svoje personalne podatke u isto vreme kada se odigrava i automatsko resetovanje penala za sve klijente. Dešava se da su klijent i sistem u tom trenutku izloženi istom stanju sistema. I da će se izmene entiteta primeniti samo za onaj proces koji će se izvršiti poslednji.

Rešenje problema: Rešenje za ovaj problem je slično kao i kod problema 1 i 2. Kod klase *User* dodaju se dva nova atributa, *version* i boolean *isChanged*. Verzioniranjem klijentorskog entiteta omogućava se provera njegove verzije pri upisivanju izmenjenog entiteta nazad u bazu. Ukoliko se verzije ne poklope baca se *ObjectOptimisticLockingFailureException*. Sve servisne metode koje učestvuju u ovom procesu anotirane su sa `@Transactional`.

```
@Scheduled(cron="0 0 0 1 * *")
public void resetClientPenalties()
```



```
@PostMapping(consumes="application/json", value="/data-update")
@PreAuthorize("hasRole('CLIENT')")
@CrossOrigin(origins = ServerConfig.FRONTEND_ORIGIN)
public ResponseEntity<ClientProfileDTO> updatePersonalData(@RequestBody
```

