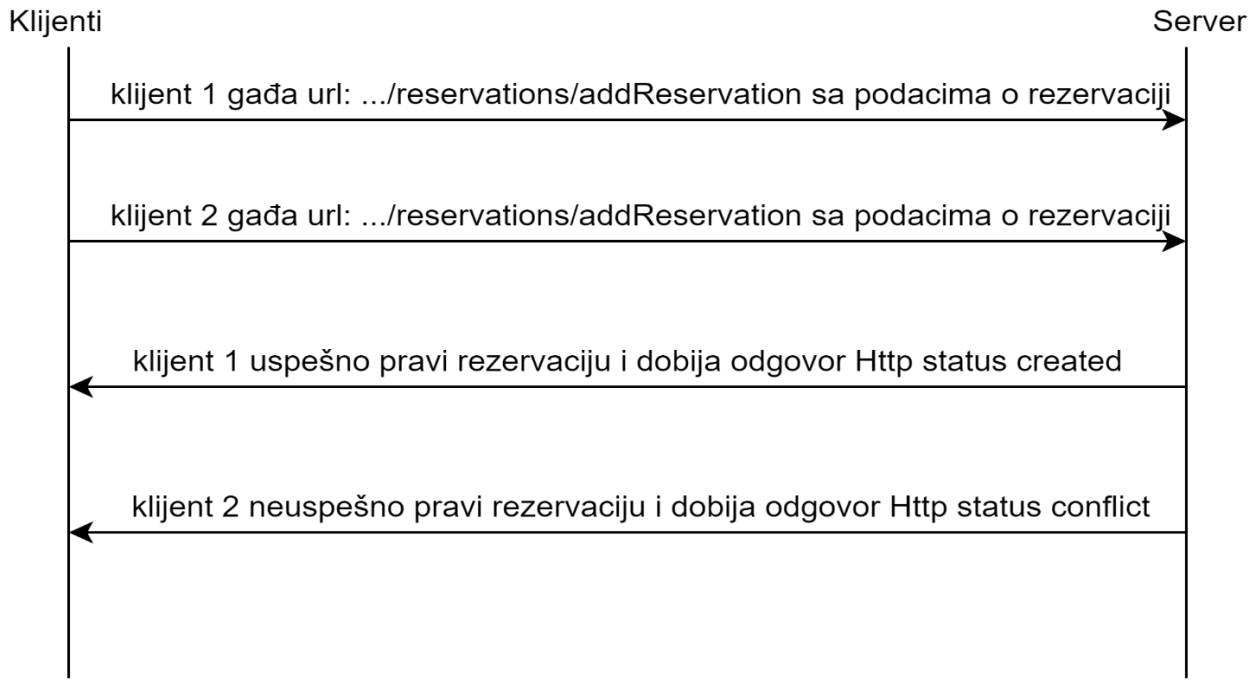
Student 1 (Mihal Sabadoš)

Konfliktna situacija 1

Više istovremenih klijenata pokušavaju da naprave rezervaciju istog entiteta u isto Ili preklapajuće vreme. Konflikt se rešava korišćenjem verzija odnosno optimističko zaključavanje entiteta nad kojim klijent želi da izvrši rezervaciju i time se na vrlo jednostavan način obezbeđuje da ne dođe do konfliktne situacije koja bi dovela do zakazivanja rezervacija dva ili više klijenta u isto vreme koje bi se preklapale. Sama implementacija optimističnog zaključavanja se odvija dodavanjem polja version i anotacije iznad polja u bean klasi entiteta kao I dodavanjem polja zaključanosti kojim bi hibernate shvatio da je došlo do promene stanja entiteta I automatski bi povećao verziju.

Kontroler koji se gađa prilikom rezervacije je **ReservationController** a metoda je **createReservation** koja prihvata u telu zahteva wrappovanu rezervaciju tipa **ReservationDTO** koja sadrži sve informacije koje su neophodne za kreiranje rezervacije.

Prilikom dodavanja nove rezervacije u metodi addReservation u reservationServisu se popunjavaju svi podaci iz wrappovane rezervacije i menja se polje zaključanosti u entitetu nad kojim se vrši rezervacija I čuvaju se promene čime se podiže verzija entiteta, nakon uspešne rezervacije polje zaključanosti se vraća na staro I time se čime se verzija entiteta vraća na prethodnu, ukoliko dođe do konflikta tada se baca **OptimisticLockException** i ne nastavlja se čuvanje podataka.

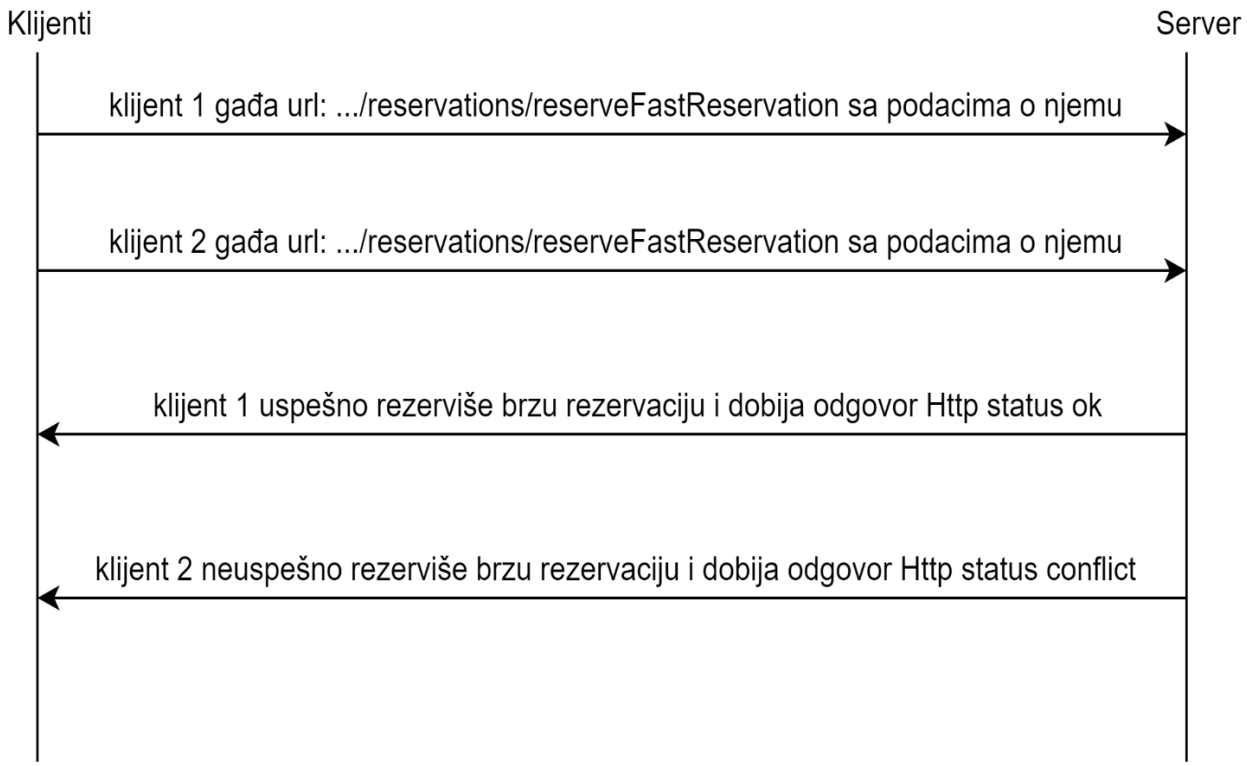


Konfliktna situacija 2

Više istovremenih klijenata pokušavaju da naprave brzu rezervaciju istog entiteta u isto Ili preklapajuće vreme. Konflikt se rešava korišćenjem verzija odnosno optimističko zaključavanje rezervacije nad kojim klijent želi da izvrši brzu rezervaciju i time se na vrlo jednostavan način obezbeđuje da ne dođe do konfliktne situacije koja bi dovela do zakazivanja brze rezervacija dva ili više klijenta u isto vreme koje bi se preklapale. Sama implementacija optimističnog zaključavanja se odvija dodavanjem polja version i anotacije iznad polja u bean klasi rezervacije, pošto se prilikom rezervacije brze rezervacije menja stanje već postojeće rezervacije, nije potrebno imati još polje zaključanosti kao kod prethodnog primera jer će hibernate automatski prepoznati promenu I time povećati verziju same rezervacije.

Kontroler koji se gađa prilikom rezervacije brze rezervacije je **ReservationController** a metoda je **reserveFastReservation** koja prihvata u telu zahteva wrappovanu rezervaciju tipa **ReservationDTO** koja već postoji u bazi i sadrži informacije o klijentu koji je rezervisao brzu rezervaciju.

Prilikom dodavanja klijenta već postojećoj rezervaciji u metodi **reserveFastReservation** u **reservationServisu** se menja se polje klijenta koje je bilo nepostojeće (**null**) i čuvaju se promene čime se podiže verzija entiteta, ukoliko dođe do konflikta tada se baca **OptimisticLockException** i ne nastavlja se čuvanje podataka.



Konfliktna situacija 3

Više istovremenih klijenata pokušavaju sa istim nalogom da ismene informacije o njima.

Konflikt se rešava korišćenjem verzija odnosno optimističko zaključavanje klijenta nad kojim se vrši izmena podataka i time se na vrlo jednostavan način obezbeđuje da ne dođe do konfliktne situacije koja bi dovela do ismene informacija u isto vreme koje bi se preklapale. Sama implementacija optimističnog zaključavanja se odvija dodavanjem polja version i anotacije @Version iznad polja u bean klasi klijenta, pošto se prilikom ismene podataka o klijentu menja stanje već postojećeg klijenta, hibernate automatski prepoznati promenu I time povećati verziju samih podataka o klijentu.

Kontroler koji se gađa prilikom izmene podataka o korisniku je **UserController** a metoda je **updateUser** koja prihvata u putanji identifikacioni broj korisnika (**id**) I u telu zahteva wrappovane podatke o korisniku tipa **UserDTO** koja sadrži izmenjene podatke o korisniku.

Prilikom izmene podataka o korisniku u metodi **updateUser** u **UserServisu** se menjaju podaci o korisniku i čuvaju se promene čime se podiže verzija entiteta, ukoliko dođe do konflikta tada se baca **OptimisticLockException** i ne nastavlja se čuvanje podataka.