# Komunikacija sa FIX inicijatorom TCP protokolom

## Uvod

* Softver za slanje FIX poruka, u daljem tekstu FIX inicijator, sakriven je od kranjeg korisnika i njemu se pristupa preko posrednika (proxy) koji predstavlja server sa kojim se prica iskljucivo TCP protokolom
* Server pruza 2 vrste usluga:

1. Usluge na zahtev (RMI)
2. Notifikacije (Data streams)

* Da bi se konzumirale bilo koje od ovih usluga nephodno je ispuniti sledece preduslove:

1. Poslati kredencijale koji su predefinisani u bazi podataka
2. Poslati ispravan **Secret**. (U slucaju ispravnih kredencijala [korak 1], klijent dobija Secret koji jedinstveno odredjuje njegovu sesiju)

* Ispunjavanjem ovih preduslova server kreira sesiju koja jedinstveno odredjuje konekciju izmedju kranjeg korisnika i FIX incijatora. Ovo znaci da nije moguce da 2 klijenta u isto vreme logicki koriste jednu konekciju (iako bi tehnicki to bilo izvodljivo).

## Komunikacija sa TCP serverom

### Uspesna komunikacija prolazi kroz 3 faze:

1. Pravljenje veze ka serveru, slanje karaktera “c” i Slanje kredencijala (server prekida vezu kad izda Secret)
2. Pravljenje veze ka serveru, slanje karaktera “s” a zatim slanje Secret-a (server odrzava ovu vezu dok postoji Fix konekcija)
3. Slanje zahteva / Primanje notifikacije

### Strukture poruka

* Kredencijali  
  <username>0<password>0
* Secret  
  n random bajta
* Zahtev  
  <string koji je ime metode>0<parametar0>0<parametar1>0...<parametar*n*>0
* Notifikacija  
  <fix poruka>0
* Tag-value sekvenca  
  <tag0>0<vrednost taga0>0...<tag*n*>0<vrednost taga*n*>00
* FIX poruka  
  <header tag-value sekvenca><tags tag-value sekvenca>  
  a grupe cemo da resavamo malo posle
* Session poruka  
  <sessionID tag-value sekvenca>

## Raskidanje veze

* U slucaju prirodnog raskidanja veze (klijent zatvori konekciju) brise se sesija koju je klijent posedovao.
* U slucaju naslinog raskidanja veze (prekid internet konekcije, nestanak struje,..) sesija se ne brise i klijent dobija 10(?) minuta da se ponovo uloguje. U tom vremenskom periodu poruke koje treba klijentu da se isporuce se baferuju (RAM) do odredjenog limita(?). U slucaju da se klijent ponovo uloguje za to vreme, njemu ce biti isporucene sve poruke iz bafera koje nije dobio za vreme prekida veze (10min). U suprotnom, sve poruke koje su bile u baferu se pisu u bazu podataka, brisu iz bafera i sesija se brise. Pri prvom sledecem logovanju sa istim kredencijalima dovlace se neisporucene poruke iz baze podataka, salju se klijentu a zatim se brisu iz baze podataka.

## Uspesan scenario komunikacije

1. Klijent uspostavlja ***obicnu*** konekciju sa TCP serverom na dogovorenoj adresi i portu
2. Klijent salje kredencijale u formatu: **c<name>0<password>0**
3. Server proverava kredencijale
4. Server odgovara slucajno generisanim 16-bitnim heksadecimalnim brojem u formatu: **r<tajni\_kljuc>**, koji predstavlja tajni kljuc za uspostavljanje ***zasticene*** konekcije
5. Server raskida konekciju
6. Klijent uspostavlja ***obicnu*** konekciju sa TCP serverom na dogovernoj adresi i portu
7. Klijent salje tajni kljuc u formatu: **s<tajni\_kljuc>**
8. Server proverava tajni kljuc
9. Server odgovara porukom **rcorrect\_secret**. Ovim je uspotavljena ***zasticena*** konekcija
10. Klijent prima poruku **rcorrect\_secret**, i ovim ostvaruje mogucnost da ***poziva metode*** i ***prima obavestenja*** od servera

## Alternativni scenario komunikacije

* 3a. Kredencijali su pogresni. Izvrsava se korak 5, povratak na korak 1.
* 8a. Tajni kljuc je pogresan. Izvrsava se korak 5, povratak na korak 1.
* 10a. Poruka za poziv metode nije ispravna. Izvrsava se korak 5, povratak na korak1.

## Pozivanje metoda i primanje obavestenja

* ***Metode*** se pozivaju u formatu: <ime\_metode>0<parametar\_0>0<parametar\_1>0...<paramentar\_n>0
* ***Metode*** koje implementira server:
* ***startFixInitiator*** – prima 1 parametar (String) koji predstavlja konfiguraciju FIX inicijatora (primer TODO). U slucaju uspesnog startovanja FIX inicijatora server odgovara sa **rfix\_initiator\_started**, u suprotnom odgovora greskom u formatu **e<greska>**
* ***sendFixMsg*** – prima 1 parametar (FIX poruka, anchor na strukturu gore TODO) koji predstavlja FIX poruku koja ce biti poslata FIX primaocu. U slucaju uspesnog slanja poruke server odgovara sa **rsuccessfully\_sent**, u suprotnom odgovara greskom u formatu **e<greska>**
* ***getStatistics*** – bez parametara. U slucaju uspesnog izvrsavanja server odgovara sa **r<statistika>** (dodati gore u tipove statistiku kad bude uradjena TODO), u suprotnom odgovara greskom u formatu **e<greska>**
* ***Obavestenja*** se primaju u formatu: o<ime\_obavestenja>0<argument\_1>0<argument\_2>...<argument\_n>0
* ***Obavestenja*** koja server salje:
* ***connectionEstablished –*** salje se 1 argument (Session poruka) koja predstavlja stanje sesije koja je uspostavljena izmedju FIX inicijatora i primaoca
* ***acceptFixMsg –*** salje se 1 argument (FIX poruka, anchor na strukturu gore TODO) koji predstavlja FIX poruku koja je primljena od FIX primaoca.
* ***ConnectionClosed –*** TODO