



SREDIŠNJE
KLIRINŠKO
DEPOZITARNO
DRUŠTVO

TEHNIČKA DOKUMENTACIJA

**za korištenje aplikativnog sučelja
Središnjeg klirinškog depozitarnog društva d.d.**

rujan 2023.

POVLJEST DOKUMENTA	4
UVOD	6
ŠTO JE APLIKATIVNO SUČELJE	6
OSNOVNE ZNAČAJKE	6
WEB SERVIS	6
PRIJE SAMOG KORIŠTENJA	6
STRUKTURE ZA RAZMJENU PODATAKA	7
ULAGATELJI.....	7
POZICIJA	8
ZAHTJEV ZA REGISTRACIJU	9
INSTRUKCIJA ZA NAMIRU ISPORUKE / PRIMITKA	10
TRANSAKCIJA	12
ALOKACIJA	13
RAČUN	14
KORIŠTENJE I PRIMJERI.....	15
PRIJAVA I ODJAVA	16
DOHVAT PODATAKA	17
DOHVAT PODATAKA O ULAGATELJU	17
PREGLED SVIH POZICIJA REGISTRIRANIH S ČLANOM ZA POJEDINI RAČUN.....	17
PREGLED SVIH POZICIJA NA OSNOVNOM RAČUNU ČLANA.....	18
PREGLED SVIH POZICIJA REGISTRIRANIH S ČLANOM	18
PREGLED PORUKA SUSTAVA	19
PREGLED INSTRUKCIJA ZA NAMIRU ISPORUKE/PRIMITKA ČLANA	20
PREGLED INSTRUKCIJA ZA NAMIRU ISPORUKE/PRIMITKA SUPROTNE UGOVORNE STRANE	20
PREGLED ZAHTJEVA ZA REGISTRACIJU.....	21
PREGLED TRANSAKCIJA ČLANA.....	22
PREGLED ALOKACIJA PO TRANSAKCIJI	23
DOHVAT BROJA TRANSAKCIJE U SUSTAVU SKDD-A	23
UNOS I BRISANJE PODATAKA	24
UPIS ALOKACIJE.....	24

UPIS VIŠESTRUKIH ALOKACIJA	25
UPIS PREDBILJEŽBE ULAGATELJA.....	28
UPIS ZAHTJEVA ZA REGISTRACIJU	29
UPIS INSTRUKCIJE ZA NAMIRU.....	30
UPIS IZRAVNOG ODOBRENJA	32
VERIFIKACIJA PROČITANIH PORUKA (POSLANIH ČLANU KROZ SUSTAV SKDD-A)	33
BRISANJE ALOKACIJE	34
OTKAZIVANJE ZAHTJEVA ZA REGISTRACIJU.....	35
UPIS RAČUNA.....	35
BRISANJE RAČUNA.....	36
PROVEDBA ZBIRNOG IZMIRENJA.....	37
STATUSNI KODOVI	39

Povijest dokumenta

Verzija	Opis	Datum
3.1.	Strukture za razmjenu podataka – Proširena struktura Instrukcija za namiru isporuke / primitka (DCAAccount, OptOutMC, TradeExecutedMC) i Račun (CAPaymentAccount) Upis instrukcije za namiru – Izmjena primjera Upis računa – Izmjena primjera	01.07.2023
2.0	Strukture za razmjenu podataka - proširena struktura Transakcije (MIC, DateMatched, DeliveryType, PriceCurrency, SettlementCurrency, BuyInContinue, BuyInExtensionLastDate, BuyInLastdate, BuyInDeferralLastDate i PlaceOFClear) i Instrukcije isporuke/primitka (SettlementInstructionType, TransactionType, DealPriceCurrency, BuyInContinue i PlaceOFClear). SDA_GetInstruction - dopuna podataka u strukturi za slanje upita. SDA_GetTransactions - dopuna podataka u strukturi za slanje upita. SDA_AddInstruction - izmjena poziva u dijelu obaveznih i neobaveznih parametara.	01.02.2022
1.9	Strukture za razmjenu podataka - u strukturi Transakcije dužina polja ExchangeExecutionID se povećava s 16 na 22 znaka SDA_GetCDSReference - dodatno pojašnjenje za transakcije sklopljene na Zagrebačkoj burzi od uključivši 09.12.2019.	09.12.2019
1.8	Nadopunjeni statusni kodovi	14.11.2018
1.7	Novi URL servisa za produkcijsko okruženje	13.09.2018
1.6	Novi URL servisa za testno okruženje	27.06.2018
1.5	SDA_TFT_payment – nova funkcija za provedbu zbirnog izmirenja	25.01.2018
1.4	InstitutionalDelivery - proširena struktura podataka (TradeAllocation, TradeDate, ForeignBIC, ForeignAccountId), te su dodane dvije nove vrijednosti za polje Status SDA_GetInstruction - proširen poziv trima neobaveznim parametrima SDA_AddInstruction - proširen poziv četirima neobaveznim parametrima SDA_AddAllocations - nova funkcija za upis višestrukih alokacija	27.03.2015

1.3	Client – dodan IBAN SDA_AddPendingClient – dodan neobavezan parameter IBAN	11.07.2012
1.2	SDA_AddPendingClient – dodani podaci o adresi, ukinuta je mogućnost isplate iz korporativnih akcija u gotovini Client - struktura preimenovana u „Ulagatelj” PendingAccount - struktura podataka preimenovana u „Račun“ SDA_AddPendingAccount - koristi za upis računa portfelja i skrbničkih računa SDA_DeleteAccount – nova funkcija za brisanje računa	15.01.2012
1.1	Inicijalna verzija	04.01.2012

Uvod

Što je aplikativno sučelje

Zbog omogućavanja bržeg i pouzdanijeg protoka informacija između informacijskih sustava članova te sustava Središnjeg klirinškog depozitarnog društva (dalje: SKDD) eksponirani su određeni poslovni procesi kroz tehnologiju web servisa tj. korištenjem SOAP protokola. Time je u osnovi omogućena izravna komunikacija aplikativnih rješenja članova sa sustavom SKDD-a, tj. implementacija određene poslovne logike vezane uz rad s SKDD-om direktno u aplikativna rješenja članova.

Osnovne značajke

S tehnološkog gledišta, ova funkcionalnost je izvedena kao web servis, pod kojim se misli na modularnu aplikaciju koja se poziva preko intra/interneta. Korisnici web servisa su uobičajeno druge aplikacije koje preko HTTP transportnog protokola komuniciraju koristeći XML standarde kao što su SOAP, WDSL i UDDI. Time je omogućena komunikacija između različitih aplikativnih rješenja na standardan način bez obzira na programski alat i/ili operativni sustav. U konkretnom slučaju ovo sučelje je eksponirani web servis kojem se može pristupiti preko HTTP protokola te se njime omogućava dohvat i upis određenog podskupa podataka, tj. obavljanje određenih poslovnih procesa kao što su registracije pozicija, instrukcije za namiru, alokacije itd.

S razvojnog gledišta, aplikativno sučelje skup je funkcija i procedura koje korisnik usluga SKDD-a može koristiti izravno iz svog aplikativnog rješenja.

Web servis

Prije samog korištenja

Sam web servis je isključivo dostupan preko privatne računalne mreže SKDD-a. Prvi korak u korištenju web servisa u vlastitom aplikativnom rješenju jest dohvat definicije samog web servisa preko WSDL datoteke koja je dostupna s adrese web servisa <http://soa.skdd.hr:2222/SDAInterface?wsdl>. Korištenjem ove datoteke dostupne su sve definicije funkcija, procedura i struktura koje se mogu koristiti.

Moguće je doći do testnog web servisa putem WSDL datoteke koja je dostupna s adresom <https://servis.skdd.hr:2221/SDAInterface?wsdl>.

Cjelokupni web servis izrađen je tako da pamti sesije (HTTP Session Enabled) te time omogućava da se tijekom rada jednom prijavite te nadalje koristite ponuđene funkcionalnosti sve dok sesija ne istekne zbog neaktivnosti ili odjavom korisnika. Servis koristi SOAP1.2 protokol za razmjenu poruka.

Što se tiče sigurnosti podataka, sav mrežni promet se kriptira na identičan način kao i do sada „pametnim“ karticama i Checkpoint softverom pa nema dodatnih potreba za sigurnošću prijenosa podataka.

Da bi član mogao koristiti ovaj način rada sa sustavom SKDD-a, potrebno je prethodno aktivirati DEP-CLN nalog.

Strukture za razmjenu podataka

Većina funkcija i procedura iz web servisa kao ulazne i/ili izlazne parametre imaju strukturirane podatke koji upravo opisuju pojedini logički entitet u samom poslovnim procesu.

Strukture koje se koriste su sljedeće:

Ulagatelji

Struktura opisuje ulagatelje iz evidencije SKDD-a.

Entitet	Ulagatelj		
Naziv strukture	Client		
Naziv polja	Tip	Dužina	Opis
ClientID	Int32		Oznaka ulagatelja
Status	String	2	Status ulagatelja (DP,OP,OT,SS)
LocalClient_ID	String	13	MBG ili MB
Individual	String	1	Identifikator fizičke osobe
ClientType	String	2	Vrsta ulagatelja (DF,SO itd...)
Nationality	String	2	Oznaka države za državljanstvo
Name	String	160	Ime/tvrtka
Surname	String	50	Prezime/Skraćena tvrtka
Address1	String	60	Adresa
Address2	String	30	Adresa
Address3	String	15	Adresa
Town	String	50	Mjesto
PostCode	String	15	Broj pošte
CountryCode	String	2	Oznaka države za dopisnu adresu
DateOfIncorporation	Date	-	Datum rođenja/osnutka
PaymentType	String	1	Način isplate (B-račun)

BankVBDI	String	7	VBDI banke
BankAccountNo	String	10	Broj računa
BankSubAccountNo	String	22	Partija – polje se više ne koristi
DisposalInstruction	String	80	Uputa za plaćanja
EntitlementUnassigned	int	-	Novi VP na neregistriranu vl. poziciju ili ne
PreferredLanguage	String	1	Jezik za izvješćivanje (E-engleski C-hrvatski)
TaxMunicipality	int	3	Grad/općina rezidentnosti
TaxCategory	int	2	Oporezivanje primitaka od kapitala
TaxNumber	String	13	Porezni broj
OIB	String	11	OIB
DomicileAddress	String	80	Adresa prebivališta
DomicileTown	String	50	Grad prebivališta
DomicilePostCode	String	15	Poštanski broj grada prebivališta
DomicileCountryCode	String	2	Oznaka države prebivališta
IBAN	String	34	IBAN

Pozicija

Struktura koja opisuje pozicije na kojima su ubilježeni svi vrijednosni papiri u depozitoriju.

Entitet	Pozicija		
Naziv strukture	OwnershipPosition		
Naziv polja	Tip	Dužina	Opis
AccountID	Int32	13	Oznaka računa
ISIN	String	12	Vrijednosni papir
Balance	Double	-	Stanje
ParticipantID	String	4	Oznaka člana registracije pozicije
ParticipantType	String	1	Vrsta člana registracije pozicije
PositionType	String	2	Vrsta pozicije
PositionSubType	String	1	Status otplate (O,N)
Status	String	2	Status pozicije (DP,OP,OT,SS)

Zahtjev za registraciju

Struktura opisuje zahtjev za registraciju pozicije.

Entitet	Zahtjev za registraciju		
Naziv strukture	RegistrationRequest		
Naziv polja	Tip	Dužina	Opis
RequestNumber	Int32	10	Broj zahtjeva za registraciju
RequestDate	Date	-	Datum zahjteva
Status	Int16	1	Statusi registracija: 1=Zahtjev podnesen 2=Djelomično izvršen 3=Izvršen u cjelosti 4=Istekao 5=Otkazan 6=Brisan
TransferUnassigned	String	1	Y-odregistracija N-registracija
ParticipantID	String	4	Oznaka člana
ParticipantType	String	1	Vrsta člana
ISIN	String	12	Vrijednosni papir
FromAccountID	Int32	13	Na teret računa
ToAccountID	Int32	13	U korist računa
AccType	String	2	Vrsta pozicije
AccSubType	String	1	Status otplate pozicije
RequestQty	Double	-	Neprenesena količina
TransferredQty	Double	-	Prenesena količina
ExpiryDate	Date	-	Datum isteka zahtjeva
ConfirmationMessage	String	1	Slanje poruke o potvrdi
Remarks	String	80	Napomena
AddedBy	String	30	Upisao
DateAdded	Date	-	Datum upisa

Instrukcija za namiru isporuke / primitka

Struktura instrukcije isporuke ili primitka u sustavu instrukcija za namiru.

Entitet	Instrukcija za namiru isporuke/primitka		
Naziv strukture	InstitutionalDelivery		
Naziv polja	Tip	Dužina	Opis
IdReference	Int32	6	Broj instrukcije
IdTradeDate	Date	-	Datum instrukcije
DeliverReceive	String	1	Isporuka/primitak
DeliveryType	String	1	Vrsta isporuke: F-BEZ PLAĆANJA P-ISPORUKA PO PLAĆANJU W-ISPORUKA UZ PLAĆANJE O-PLAĆANJE BEZ ISPORUKE D-ZAMJENA VRIJEDNOSNIH PAPIRA
SettlementInstructionType	String	4	Vrsta instrukcije: DFP -ISPORUKA BEZ PLAĆANJA RFP-PRIMITAK BEZ PLAĆANJA DVP-ISPORUKA PO PLAĆANJU RVP-PRIMITAK PO PLAĆANJU DWP-ISPORUKA S PLAĆANJEM RWP-PRIMITAK S PLAĆANJEM DFOD-TEREĆENJE PLAĆANJA BEZ ISPORUKE CFOD-KREDITIRANJE PLAĆANJA BEZ ISPORUKE
TransactionType	String	4	Vrsta transakcije, u skladu s ISO standardom (TRAD,...)
DeliveryDate	Date		Datum namire
ParticipantId	String	4	Oznaka člana
ParticipantType	String	1	Vrsta članstva
ISIN	String	12	Vrijednosni papir (ISIN)
AccType	String	2	Vrsta pozicije
AccSubType	String	1	Status otplate pozicije
Quantity	Double	-	Količina vrijednosnog papira
SettlementAmount	Double	-	Iznos za namiru
DealPrice	Double	-	Cijena
DealPriceCurrency	String	13	Valuta cijene (% ili ISO 4217)
DealAmount	Double	-	Vrijednost transakcije
Tax	Double	17,2	Porez
Commission	Double	17,2	Provizija
Fees	Double	17,2	Naknada
ParticipantIdBeneficiary	String	4	Oznaka člana suprotne ugovorne strane
ParticipantTypeBeneficiary	String	1	Vrsta članstva suprotne ugovorne strane
ExchangeTrade	String	1	Trgovanje s tržišta
SettlementType	int	1	1-PN,2-UN
Status	String	1	Status instrukcije 1-Upisana 2-Odobrena 3-Uparena 4-Otkazana 5-Istekla

Entitet	Instrukcija za namiru isporuke/primitka		
Naziv strukture	InstitutionalDelivery		
Naziv polja	Tip	Dužina	Opis
			6-Namirena 7-Odobrena u čekanju 8-Otkazana u čekanju
EnteredUserId	String	30	Korisnik koji je upisao instrukciju
AuthorisedUserId	String	30	Korisnik koji je odobrio instrukciju
DateAdded	Date	-	Datum upisa
ApprovedDate	Date	-	Datum odobrenja
ModifiedBy	String	30	Korisnik koji je izmjenio instrukciju
DateModified	Date	-	Datum izmjene
AccruedInterest	Double	-	Stečena kamata
DateMatched	Date	-	Datum uparenja
CdsReference	Int32	10	Broj transakcije u SKDD-u nastale uparenjem
CertifiedSell	String	1	Osigurana prodaja
IdReferenceBeneficiary	Int32	6	Broj instrukcije s kojom se ista uparila
IdTradeDateBeneficiary	Date	-	Datum instrukcije suprotne ugovorne strane
ReturnDate	Date	-	Datum povrata
ReturnSettlementAmount	Double	-	Iznos za namiru povrata
AccountID	Int32	13	Račun vrijednosnih papira za alokaciju
AccountIdBeneficiary	Int32	13	Račun vrijednosnih papira za alokaciju suprotne ugovorne strane
SettlementCurrency	String	13	Valuta namire (ISIN kod zamjene vrijednosnih papira, ISO 4217)
Remarks	String	120	Napomena
TradeAllocation	String	1	Raspoređivanje na više računa
TradeDate	Date	-	Datum obavljene transakcije
ForeignBIC	String	11	SWIFT BIC krajnjeg kupca/prodavatelja
ForeignAccountId	String	35	Račun krajnjeg kupca/prodavatelja
BuyInContinue	Int	1	Oznaka da je transakcija nastavak provedbe prisilne kupnje iz druge nenamirene transakcije: 0-Ne, 1-Da
PlaceOfClear	String	11	BIC CCP-a ako je transakcija poravnava u CCP-u, inače prazno
DCAAccount	String	34	Šifra DCA računa ukoliko se za instrukciju koristi DCA
OptOutMC	String	4	NOMC ukoliko se zabranjuje automatsko generiranje Market Claim, inače prazno
TradeExecutedMC	String	4	CCPN - Cum Coupon, XCPN – Ex Coupon, inače prazno

Transakcija

Struktura opisuje transakciju u sustavu poravnanja i namire.

Entitet	Transakcija		
Naziv strukture	Transaction		
Naziv polja	Tip	Dužina	Opis
ExchangeID	Int	1	Oznaka tržišta: 1-ZAGREBAČKA BURZA 2-VARAŽDINSKA BURZA 3-INSTRUKCIJA ZA NAMIRU 4-POKRIĆE OBVEZE S KUĆNOG RAČUNA 5-AUTOMATSKI ZAJAM 6-ZAJAM 7-IZDAVANJE 8-POVLAČENJE 9-PREKOGRA NIČNE TRANSAKCIJE 10-TRŽIŠTE NOVCA ZAGREB 11-AUKCIJA 12-ZAJAM
MIC	String	4	Mjesto trgovanja (ISO 10383)
DateMatched	Date	-	Datum uparenja
ExchangeExecutionId	String	22	Oznaka transakcije na tržištu
DeliveryType	String	1	Vrsta isporuke: F-BEZ PLAĆANJA P-ISPORUKA PO PLAĆANJU W-ISPORUKA UZ PLAĆANJE O-PLAĆANJE BEZ ISPORUKE D-ZAMJENA VRIJEDNOSNIH PAPIRA
CDSReference	Int32	10	Broj transakcije u SKDD-u
TradeDate	Date	-	Datum transakcije
TradeTime	Date	-	Vrijeme transakcije
ISIN	String	12	Vrijednosni papir
Quantity	Double	-	Količina
AllocatedQuantity	Double	-	Alocirana količina
Price	Double	-	Cijena
PriceCurrency	String	13	Valuta cijene (% ili ISO 4217)
tieCdsReference	Int32	10	Broj vezane transakcije
tieTradeDate	Date	-	Datum vezane transakcije
SettlementDate	Date	-	Predviđeni datum namire
DateSettled	Date	-	Datum namire
TransactionType	String	1	Normalna ili blok transakcija
SettlementType	Int32	1	Vrsta namire: 1-pojedinačna 2-ugovorna 3-namira zamjenom
SettlementAmount	Double	17,2	Iznos za namiru
SettlementCurrency	String	13	Valuta namire (ISO 4217)
AccruedInterest	Double	17,2	Iznos stečene kamate
BuySell	String	1	Kupnja (B) ili prodaja (S)
TransactionStatus	Int	1	Status transakcije 1-Nenamirena 2-Namirena

Entitet	Transakcija		
Naziv strukture	Transaction		
Naziv polja	Tip	Dužina	Opis
			5-Neuspjela 6-Otkazana
IdTrade	String	1	Instrukcija za namiru: Y-Da, N-Ne
FreeTrade	String	1	Namira bez (Y) ili uz plaćanje (N)
OrderId	String	13	Broj naloga kod člana
BrokerRef	String	13	PNB kod sudionika
Certified	String	1	Osigurana prodaja
DateCertified	Date	-	Datum osiguranja prodaje
CounterParty	String	6	Suprotna ugovorna strana
BuyInContinue	Int	1	Oznaka da je transakcija nastavak provedbe prisilne kupnje iz druge nenamirene transakcije: 0-Ne, 1-Da
BuyInExtensionLastDate	Date	-	Zadnji datum produljenog razdoblja
BuyInLastDate	Date	-	Zadnji datum primjenjivog roka (za namiru prisilnom kupnjom)
BuyInDeferralLastDate	Date	-	Zadnji datum roka odgode/moguće odgode namire prisilnom kupnjom
PlaceOfClear	String	11	BIC CCP-a ako je transakcija poravnava u CCP-u, inače prazno

Alokacija

Struktura opisuje alokaciju isporuke ili primitka iz transakcije

Entitet	Alokacija		
Naziv strukture	Allocation		
Naziv polja	Tip	Dužina	Opis
CdsReference	Int32	10	Broj transakcije u SKDD-u
ISIN	String	12	Vrijednosni papir
TradeDate	Date	-	Datum trgovanja
LineNumber	Int16	3	Broj alokacije u transakciji
BuySell	String	1	Kupovna ili prodajna strana u transakciji
BuySellAllocation	String	1	Kupovna ili prodajna alokacija
QuantitySettled	Double	-	Namirena količina
QuantityUnSettled	Double	-	Nenamirena količina
Status	Int16	1	Status alokacije 1-nenamireno 2-namireno
DeliveryUnassigned	String	1	Isporuka na neregistriranu poziciju
AccountID	Int32	13	Broj računa vrijednosnih papira
ParticipantID	String	4	Oznaka člana registracije pozicije
ParticipantType	String	1	Vrsta članstva registracije pozicije
AccountType	String	2	Vrsta pozicije

AccountSubType	String	1	Status otplate pozicije
----------------	--------	---	-------------------------

Račun

Struktura opisuje račune - račun portfelja (RP) i skrbničke račune (skrbnički račun na ime (IM), zbirni skrbnički račun (ZB) i skrbnički račun pod zaporkom (PZ)).

Entitet	Upis računa		
Naziv strukture	PendingAccount		
Naziv polja	Tip	Dužina	Opis
AccountType	String	2	Vrsta računa (RP, IM, ZB, TU)
JointClientID	Int32	13	Oznaka ulagatelja kod računa portfelja (RP), skrbničkog računa na ime (IM) i Transakcijski račun ulagatelja (TU)
Nationality	String	2	Oznaka države – državljanstvo ulagatelja (obavezno kod ZB računa)
Residence	String	2	Oznaka države – rezidentnost ulagatelja (obavezno kod ZB računa)
AccountName	String	200	Naziv računa (obavezno kod ZB računa, ne upisuje se kod RP i IM računa, opcionalno kod TU računa)
BankSubAccountNo	String	22	Partija u banci
Remarks	String	80	Napomena
CAPaymentAccount	Int	1	1 – Isplata korporativnih akcija na račun za isplatu ulagatelja, 0 – Isplata korporativnih akcija na račun za isplatu člana, 2 – Isplata korporativnih akcija na DCA račun člana
T2SAccountID	String	34	T2S račun člana

Korištenje i primjeri

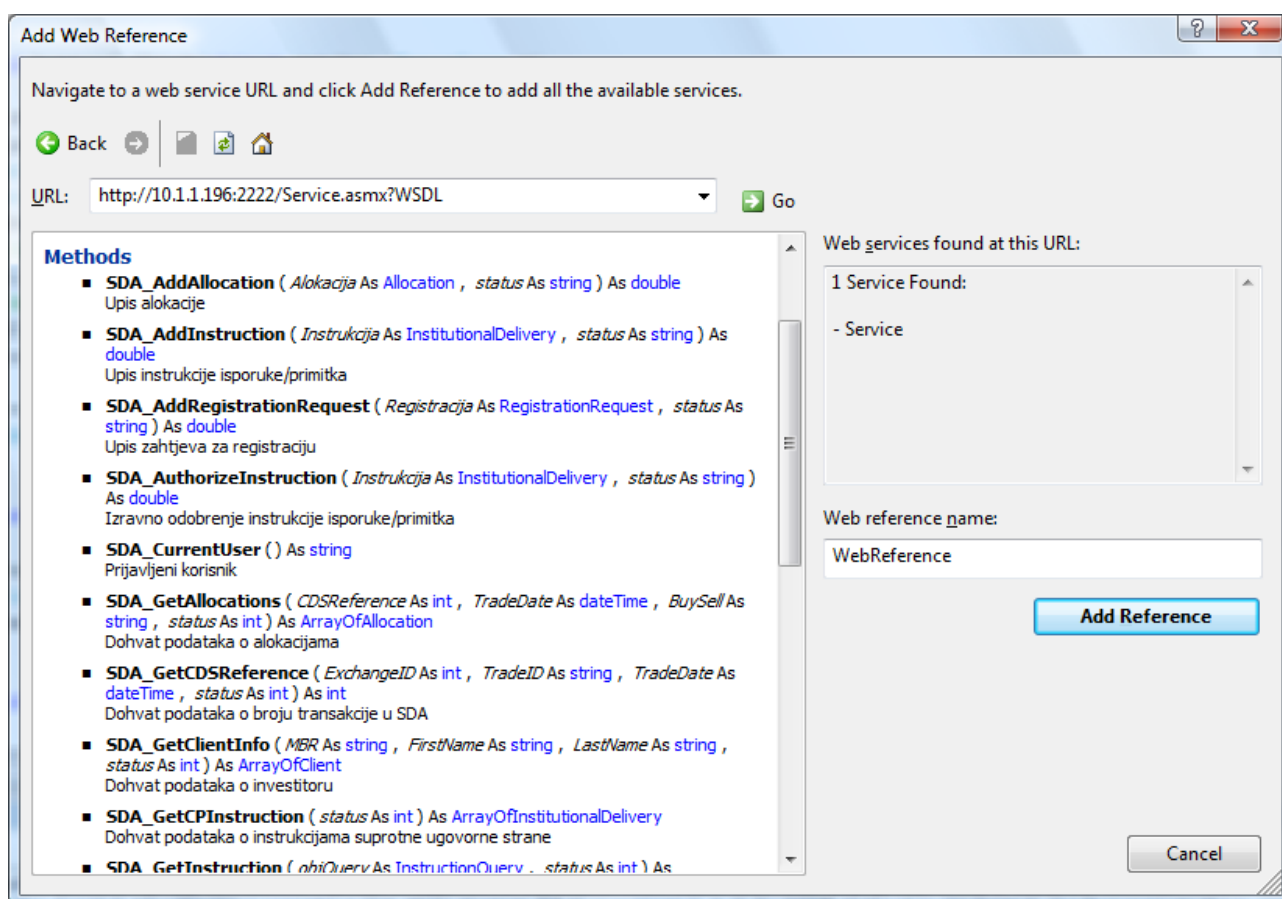
Svi primjeri koji su dani kroz ovu dokumentaciju odnose se na razvojni alat Microsoft Visual Studio uz korištenje C# programskog jezika. Osnovni principi rada s web servisima kroz ostale razvojne alate trebali bi biti vrlo slični, ako ne i isti.

Da biste mogli početi koristiti web servis potrebno je da u svom projektu dodate referencu na web servis (*Add web reference*) kao što je vidljivo na Slika 1.

Nakon toga ćete imati pristup klasi dotičnog web servisa (*SDAInterface*) i svim strukturama, funkcijama i procedurama koje ista sadrži. Nakon uspješnog dodavanja reference sama deklaracija treba izgledati ovako:

```
public SDAInterface moj servis = new SDAInterface();
```

Slika 1



Prijava i odjava

Prvi korak potreban za daljnji rad s web servisom je prijava korisnika. Za prijavu korisnika koristi se sljedeća funkcija:

SDA_Login (*username* As **string** , *password* As **string**) As **int**

Ulazni parametri: *username* – korisničko ime

password – zaporka

Povratna vrijednost: 0 – uspješna prijava korisnika

U primjeru prijava treba izgledati ovako:

```
SDAInterface mojservis = new SDAInterface();
if (mojservis.SDA_Login("BR01B", "BR01B") ==0 )
{
    MessageBox.Show("Prijava uspjela");
}
else
{
    MessageBox.Show("Probaj ponovo");
}
```

Da bi prijava bila funkcionalna mora se omogućiti održavanje i pamćenje sesija između klijentske aplikacije i Aplikativnog sučelja. To se postiže na sljedeći način:

```
SDAInterface mojservis = new SDAInterface();
// instanciranje Cookie container-a
public System.Net.CookieContainer Cookie=new
System.Net.CookieContainer();

mojservis.CookieContainer = Cookie;
```

Odjava se obavlja pozivom na sljedeću funkciju:

SDA_Logout () As **int**

Ulazni parametri: *-nema-*

Povratna vrijednost: 0 – uspješna odjava korisnika

Primjer

```
SDAInterface mojservis = new SDAInterface();
// prijava korisnika
//.....
//
if (mojservis.SDA_Logout() ==0 )
{
    MessageBox.Show("Odjava uspjela");
}
```


Dohvat podataka

Svrha funkcija i procedura ovog dijela web servisa je omogućavanje dohvata podataka kojima član trenutno ima pristup preko aplikativnog rješenja „Deprin“. Svi podaci se dobivaju u XML formatu u poljima struktura ranije navedenih u dokumentu.

Dohvat podataka o ulagatelju

Svrha funkcije je omogućavanje dohvata podataka o ulagateljima iz evidencije SKDD-a i to prema određenim uvjetima traženja. Pretraživanje je moguće zadati po matičnom broju, prezimenu i imenu ulagatelja. Barem jedan od kriterija traženja mora biti zadan jer u protivnom funkcija vraća praznu strukturu.

Definicija:

SDA_GetClientInfo (*OIB* As *string* , *FirstName* As *string* , *LastName* As *string* , *status* As *int*) As *ArrayOfClient*

Ulazni parametri: *OIB* – *Osobni identifikacijski broj*
 FirstName – *Ime i/tvrtka*
 LastName – *Prezime/skraćena tvrtka*
 Status – *status pozvane funkcije (0 – uspješno proveden upit)*

Povratna vrijednost: Polje strukture Client

Primjer:

```
// instanciranje klase te prijava korisnika
//...
int status=0;
Client[] ulagatelji = mojservis.SDA_GetClientInfo("", "Pero", "Perić", ref
status);
//prikaz podataka o ulagateljima u data grid-u
DataGrid1.DataSource = ulagatelji;
```

Pregled svih pozicija registriranih s članom za pojedini račun

Ova funkcija nudi mogućnost dobivanja podataka o svim trenutno registriranim pozicijama s članom za navedeni račun vrijednosnih papira. Poziv funkcije ostvaruje se prosljeđivanjem parametra oznaka računa vrijednosnih papira. Ukoliko navedeni račun postoji i na njemu su registrirane pozicije i navedeni korisnik ima pravo na te podatke, utoliko se ti podaci vraćaju u navedenom obujmu.

*Definicija***SDA_ReconcileAccount** (*AccountID* As **Int32** , *status* As **int**) As **ArrayOfOwnershipPosition****Ulazni parametri:** *AccountID*– oznaka računa vrijednosnih papira*Status* – status pozvane funkcije (0 – uspješno proveden upit)**Povratna vrijednost:** Polje strukture OwnershipPosition*Primjer*

```
// instanciranje klase te prijava korisnika
//...
int status=0;
OwnershipPosition[] pozicije = mojservis.SDA_ReconcileAccount(123456, ref
status);
//prikaz podataka o pozicijama u data grid-u
DataGrid1.DataSource = pozicije;
```

Pregled svih pozicija na osnovnom računu člana

Ova funkcija nudi mogućnost dobivanja podataka o svim pozicijama ubilježenim na osnovnom računu člana. Poziv funkcije ostvaruje se proslijeđivanjem PIN-a za osnovni račun člana.

*Definicija:***SDA_ReconcileHomeAccount** (*PIN* As **String** , *status* As **int**) As **ArrayOfOwnershipPosition****Ulazni parametri:** *PIN*– PIN za osnovni račun člana*Status* – status pozvane funkcije (0 – uspješno proveden upit)**Povratna vrijednost:** Polje strukture OwnershipPosition*Primjer:*

```
// instanciranje klase te prijava korisnika
//...
int status=0;
OwnershipPosition[] pozicije =
mojservis.SDA_ReconcileHomeAccount(„12345“, ref status);
//prikaz podataka o pozicijama u data grid-u
DataGrid1.DataSource = pozicije;
```

Pregled svih pozicija registriranih s članom

Ova funkcija nudi mogućnost dobivanja podataka o svim trenutno registriranim pozicijama s članom u sustavu SKDD-a.

*Definicija:***SDA_ReconcilePositions** (*status* As **int**) As **ArrayOfOwnershipPosition****Ulazni parametri:** *Status* – status pozvane funkcije (0 – uspješno proveden upit)**Povratna vrijednost:** Polje strukture OwnershipPosition*Primjer:*

```
// instanciranje klase te prijava korisnika
//...
int status=0;
OwnershipPosition[] pozicije =
mojservis.SDA_ReconcilePositions(ref status);
//prikaz podataka o pozicijama u data grid-u
DataGrid1.DataSource = pozicije;
```

Pregled poruka sustava

Ova funkcija nudi mogućnost dobivanja podataka o svim nepročitanim porukama sustava SKDD-a ili o određenoj poruci bez obzira na status. Poziv funkcije ostvaruje se prosljeđivanjem parametara broja poruke ako se želi dobiti određena poruka, a u slučaju da se žele dobiti sve nepročitane poruke, tada tom parametru pridružite vrijednost nula.

*Definicija:***SDA_GetMessages** (*MessageNo* As **int**, *status* As **int**) As **ArrayOfOwnershipPosition****Ulazni parametri:** *MessageNo* – broj poruke ili 0 za sve nepročitane*Status* – status pozvane funkcije (0 – uspješno proveden upit)**Povratna vrijednost:** Polje strukture Message*Primjer:*

```
// instanciranje klase te prijava korisnika
//...
int status=0;
Message[] poruke =
mojservis.SDA_GetMessages(0,ref status); // dohvat svih nepročitanih
//prikaz podataka o porukama u data grid-u
DataGrid1.DataSource = poruke;
```

Pregled instrukcija za namiru isporuke/primitka člana

Ova funkcija nudi mogućnost dobivanja podataka o svim instrukcijama za namiru primitka/iskoruke. Poziv funkcije ostvaruje se prosljeđivanjem parametra koji je kompletna struktura instrukcije.

Međutim, ne koriste se sva polja iz strukture za upit, nego samo određeni podskup koji je niže opisan. Barem jedan uvjet mora biti zadan, a u protivnom funkcija vraća praznu strukturu. Korisnik kao rezultat dobiva sve podatke o instrukcijama.

Definicija:

SDA_GetInstruction (*objQuery* As **InstitutionalDelivery**, *status* As **int**) As **ArrayOfInstitutionalDelivery**

Ulazni parametri: *objQuery*– struktura (instrukcija) za slanje upita a u kojoj se mogu koristiti sljedeća polja: *IdReference*, *SettlementType*, *DeliveryType*, *DeliverReceive*, *ParticipantIdBeneficiary*, *ParticipantTypeBeneficiary*, *ISIN*, *Status*, *TradeDate*, *DeliveryDate*, *ReturnDate*, *ForeignBIC*, *ForeignAccountID*, *TradeAllocation*, *SettlementInstructionType*, *TransactionType*, *BuyInContinue*, *PlaceOfClear*
Status – status pozvane funkcije (0 – uspješno proveden upit)

Primjer:

```
// instanciranje klase te prijava korisnika
//...
int status=0;
// kreiranje upita
InstitutionalDelivery upit = new InstitutionalDelivery();
upit.ISIN = "HRPLVARA0004";
upit.DeliverReceive="D";
upit.DeliveryType="F";
// poziv funkcije
InstitutionalDelivery [] instrukcije =
mojservis.SDA_GetInstruction(upit,ref status);
//prikaz podataka o instrukcijama u data grid-u
DataGrid1.DataSource = instrukcije;
```

Pregled instrukcija za namiru isporuke/primitka suprotne ugovorne strane

Ova funkcija nudi mogućnost dobivanja podataka o odobrenim instrukcijama za namiru isporuke/primitka upućenih prema članu od suprotne ugovorne strane. Poziv funkcije ostvaruje se bez dodatnih parametara te se kao rezultat dobivaju sve odobrene instrukcije suprotne ugovorne strane (ili više njih) u obliku pripadajuće strukture instrukcije.

*Definicija:***SDA_GetCPInstruction** (*status* As **int**) As **ArrayOfInstitutionalDelivery****Ulazni parametri:** *Status* – status pozvane funkcije (0 – uspješno proveden upit)**Povratna vrijednost:** Polje strukture *InstitutionalDelivery**Primjer:*

```
// instanciranje klase te prijava korisnika
//...
int status=0;
// poziv funkcije
InstitutionalDelivery [] instrukcije =
mojservis.SDA_GetCPInstruction(ref status);
//prikaz podataka o instrukcijama u data grid-u
DataGrid1.DataSource = instrukcije;
```

Pregled zahtjeva za registraciju

Ova funkcija nudi mogućnost preuzimanja podataka o zahtjevima za registraciju člana. Dohvat je moguće obaviti prema broju zahtjeva, statusu ili računu ulagatelja prema kojemu je zahtjev upućen. Mora biti zadan barem jedan uvjet jer u protivnom funkcija vraća praznu strukturu. Korisnik kao rezultat dobiva sve podatke o traženom zahtjevu za registraciju u obliku strukture zahtjeva za registraciju.

*Definicija:***SDA_GetRegistrationRequests** (*RequestNo* As **int**, *AccountID* As **int**, *RequestStatus* As **int**, *status* As **int**) As **ArrayOfRegistrationRequest****Ulazni parametri:** *RequestNo* – broj zahtjeva*AccountID* – oznaka računa ulagatelja*RequestStatus* – status zahtjeva*Status* – status pozvane funkcije (0 – uspješno proveden upit)**Povratna vrijednost:** Polje strukture *RegistrationRequest*

Primjer:

```
// instanciranje klase te prijava korisnika
//...
int status=0;
// poziv funkcije - dohvat svih djelomično izvršenih zahjete
RegistrationRequest [] zahtjevi =
mojservis.SDA_GetRegistrationRequests(0,0,2,ref status);
//prikaz podataka o zahtjevima u data grid-u
DataGrid1.DataSource = zahtjevi;
```

Pregled transakcija člana

Ova funkcija nudi mogućnost preuzimanja podataka o transakcijama člana. Poziv funkcije se ostvaruje prosljeđivanjem parametra koji je kompletna struktura transakcije. Međutim, ne koriste se sva polja iz strukture za upit, nego samo određeni podskup koji je niže opisan. Mora biti zadan barem jedan uvjet jer u protivnom funkcija vraća praznu strukturu. Korisnik kao rezultat dobiva sve podatke o transakcijama koje zadovoljavaju zadane uvjete u pripadajućoj strukturi.

Definicija:

SDA_GetTransactions (*objQuery* As **Transaction**, *status* As **int**) As **ArrayOfTransaction**

Ulazni parametri: *objQuery*– struktura (transakcija) za slanje upita a u kojoj se mogu koristiti sljedeća polja: *CDSReference*, *ExchangeID*, *SettlementType*, *TransactionStatus*, *ISIN*, *BuySell*, *TradeDate*, *SettlementDate*, *DateSettled*, *BuyInContinue*, *PlaceOfClear*
Status – status pozvane funkcije (0 – uspješno proveden upit)

Povratna vrijednost: Polje strukture *Transaction*

Primjer:

```
// instanciranje klase te prijava korisnika
//...
int status=0;
// kreiranje upita
Transaction upit = new Transaction();
upit.ISIN = "HRPLVARA0004";
upit.BuySell="B";
// poziv funkcije
Transaction [] transakcije =
mojservis.SDA_GetTransactions(upit,ref status);
//prikaz podataka o transakcijama u data grid-u
DataGrid1.DataSource = transakcije;
```

Pregled alokacija po transakciji

Ova funkcija nudi mogućnost preuzimanja podataka o alokacijama u transakcijama. Dohvat je moguće obaviti prema broju transakcije, datumu nastanka i strani u transakciji. Mora biti zadan barem jedan uvjet jer u protivnom funkcija vraća praznu strukturu. Korisnik kao rezultat dobiva sve podatke o alokacijama u obliku pripadajuće strukture.

Definicija:

SDA_GetAllocations (*CDSReference* As *int* , *TradeDate* As *DateTime* , *BuySell* As *string* , *status* As *int*) As *ArrayOfAllocation*

Ulazni parametri: *CDSReference* – broj transakcije

TradeDate – datum transakcije

BuySell – strana u transakciji

Status – status pozvane funkcije (0 – uspješno proveden upit)

Povratna vrijednost: Polje strukture *Allocation*

Primjer:

```
// instanciranje klase te prijava korisnika
//...
int status=0;
// poziv funkcije
//traženj alokacije za trx. broj 567234 sa datumom 01.08.2007
Allocation [] alokacije =
mojservis.SDA_GetAllocations(567234,DateTime.Parse("2007/8/1"),"", ref
status);
//prikaz podataka o alokacijama u data grid-u
DataGrid1.DataSource = alokacije;
```

Dohvat broja transakcije u sustavu SKDD-a

Ova funkcija nudi mogućnost dohvata informacije o broju transakcije u sustavu SKDD-a. Transakcije nastale na uređenom tržištu imaju svoj broj koji ih identificira unutar dana. Prilikom preuzimanja transakcija u sustav SKDD-a, svakoj se dodjeljuje jedinstveni broj (*CDSReference*). Da bi se omogućilo alociranje te druge funkcionalnosti vezane uz transakcije, ova funkcija prema podacima o transakciji s uređenog tržišta vraća broj transakcije u sustavu SKDD-a. Poziv funkcije se ostvaruje prosljeđivanjem svih parametara oznaka tržišta, broj transakcije i datum trgovanja. Za transakcije sklopljene na Zagrebačkoj burzi od uključivši 9.12.2019. broj transakcije (*TradeID*) mora sadržavati i ISIN vrijednosnog papira za koji je sklopljena transakcija u formatu {ISIN}.{broj transakcije} (npr. HRHT00RA0005.187). Korisnik kao rezultat dobiva broj transakcije u sustavu SKDD-a.

Definicija:

SDA_GetCDSReference (*ExchangeID* As **int** , *TradeID* As **string** , *TradeDate* As **dateTime** , *status* As **int**) As **int**

Ulazni parametri: *ExchangeID* – oznaka tržišta
 TradeID – oznaka transakcije na tržištu
 TradeDate – datum transakcije
 Status – status pozvane funkcije (0 – uspješno proveden upit)

Povratna vrijednost: Broj transakcije u sustavu SKDD-a

Primjer:

```
// instanciranje klase te prijava korisnika
//...
int status=0;
// poziv funkcije
//traženj alokacije za trx. broj 567234 sa datumom 01.08.2007
int CDSReference =
mojservis.SDA_GetCDSReference(1,"187",DateTime.Parse("2007/8/1"), ref
status);
```

Unos i brisanje podataka

Svrha ovog dijela poslovnog procesa je upis određenog skupa podataka koje član može unijeti korištenjem „Deprin“ aplikacije. Kao i kod pregleda podataka, za pokretanje svake funkcije potrebna je prethodna identifikacija s korisničkim imenom i zaporkom. Svi učinci i posljedice upisa podataka preko web servisa su identični kao i kod upisa podataka kroz postojeće aplikativno rješenje „Deprin“.

Upis alokacije

Ova funkcija nudi mogućnost upisa pojedine alokacije. Poziv funkcije se ostvaruje prosljeđivanjem svih podataka potrebnih za upis alokacije u pripadajućoj strukturi. Kao rezultat funkcije vraća se informacija o uspješnosti alokacije ili u slučaju neuspješnosti razlog zbog kojeg alokacija nije uspjela. Obavezni parametri iz strukture *Alokacija* su:

- ISIN
- TradeDate
- CdsReference
- BuySell
- BuySellAllocation

- AccountID
- QuantityUnSettled

Polje iz strukture *DeliveryUnassigned* je opcionalno. Osim strukture koristi se još jedan parametar (CertifySell) kojim se omogućava osiguravanje isporuke vrijednosnih papira.

Definicija:

SDA_AddAllocation (*Alokacija* As *Allocation* , *status* As *string*) As *double*

Ulazni parametri: *Alokacija*– struktura (*Alokacija*) za slanje podataka za upis a u kojoj se koriste sljedeća polja: *ISIN*, *TradeDate*, *CdsReference*, *BuySell*, *BuySellAllocation*, *AccountID*, *QuantityUnSettled*, *DeliveryUnassigned*
CertifySell – osiguranje prodaje
Status – status pozvane funkcije

Povratna vrijednost: Redni broj alokacije u transakciji.

Primjer:

```
// instanciranje klase te prijava korisnika
//...
string status;
//punjenje strukture
Allocation alokacija = new Allocation();
alokacija.CdsReference=83;
alokacija.ISIN="HRPLVARA0004";
alokacija.TradeDate=DateTime.Parse("2005/8/11");
alokacija.BuySell="B";
alokacija.BuySellAllocation="B";
alokacija.QuantityUnSettled=10;
alokacija.DeliveryUnassigned="Y";
alokacija.AccountID=5187770;
//poziv funkcije
double LineNum =mojservis.SDA_AddAllocation(alokacija,"N",ref status);
```

Upis višestrukih alokacija

Ova funkcija nudi mogućnost upisa višestrukih alokacija. Poziv funkcije se ostvaruje prosljeđivanjem svih podataka potrebnih za upis alokacije u listi koja sadrži pripadajuću strukturu podataka. Kao rezultat funkcije vraća se informacija o uspješnosti alokacije ili u slučaju neuspješnosti razlog zbog kojeg alokacija nije uspjela. Obavezni parametri iz strukture

Alokacija su:

- ISIN
- TradeDate

- CdsReference
- BuySell
- BuySellAllocation
- AccountID
- QuantityUnSettled

Polje iz strukture *DeliveryUnassigned* je opcionalno. Osim strukture koristi se još jedan parametar (CertifySell) kojim se omogućava osiguravanje isporuke vrijednosnih papira.

Definicija:

SDA_AddAllocations (*Allocations* As List <Allocation> , *status* As string) As double

Ulazni parametri: *Allocations* – lista strukturom (Alokacija) za slanje podataka za upis a u kojoj se koriste sljedeća polja: *ISIN*, *TradeDate*, *CdsReference*, *BuySell*, *BuySellAllocation*, *AccountID*, *QuantityUnSettled*, *DeliveryUnassigned*
 CertifySell – osiguranje prodaje
 Status – status pozvane funkcije

Povratna vrijednost: Vrijednost 0 ako je uspješno proveden upis.

Primjer:

```
// instanciranje klase te prijava korisnika
//...
string status;
//punjenje strukture

List<Allocation> allocations = new List<Allocation>();

Allocation alokacija = new Allocation();
alokacija.CdsReference=83;
alokacija.ISIN="HRPLVARA0004";
alokacija.TradeDate=DateTime.Parse("2005/8/11");
alokacija.BuySell="B";
alokacija.BuySellAllocation="B";
alokacija.QuantityUnSettled=10;
alokacija.DeliveryUnassigned="Y";
alokacija.AccountID=5187770;

allocations.Add(alokacija);

Allocation alokacija2 = new Allocation();
alokacija2.CdsReference=83;
alokacija2.ISIN="HRPLVARA0004";
alokacija2.TradeDate=DateTime.Parse("2005/8/11");
alokacija2.BuySell="B";
alokacija2.BuySellAllocation="S";
alokacija2.QuantityUnSettled=20;
alokacija2.DeliveryUnassigned="Y";
alokacija2.AccountID=45616;

allocations.Add(alokacija2);

//poziv funkcije

double LineNum =mojservis.SDA_AddAllocation(allocations,"N",ref status);
```

Upis predbilježbe ulagatelja

Ova funkcija nudi mogućnost upisa predbilježbe ulagatelja. Poziv funkcije se ostvaruje prosljeđivanjem svih podataka potrebnih za upis predbilježbe u pripadajućoj strukturi (Ulagatelj). Kao rezultat funkcije vraća se informacija o uspješnosti upisa predbilježbe (vrijednost 1) ili u slučaju neuspješnosti razlog zbog kojeg predbilježba nije uspjela u varijabli *status*. Obavezna polja iz strukture *Ulagatelj* su:

- OIB, Surname, Name, Nationality, Address1, Town, PostCode, CountryCode, ClientType, Individual, PreferredLanguage, TaxMunicipality, TaxCategory, DomicileAddress, DomicileTown, DomicilePostCode, DomicileCountryCode;

Neobavezna polja su:

- PaymentType, BankVBDI, BankAccountNo, IBAN, BankSubAccountNo, DateOfIncorporation, Address2, Address3

Definicija:

SDA_AddPendingClient (*Klijent* As Client , *status* As string) As double

Ulazni parametri: *Klijent– struktura (Client) za slanje podataka za upis u kojoj se koristite ranije navedena polja*
Status – status pozvane funkcije

Povratna vrijednost: Vrijednost 1 ako je uspješno proveden upis.

Primjer:

```
// instanciranje klase te prijava korisnika
//...
string status;
//punjenje strukture
Client ulagatelj = new Client();
ulagatelj.OIB = "12345678901";
ulagatelj.Surname = "Perić";
ulagatelj.Name = "Pero";
ulagatelj.Nationality = "HR";
ulagatelj.Address1 = "ULICA 4";
ulagatelj.Town = "Zagreb";
ulagatelj.PostCode = "10000";
ulagatelj.CountryCode = "HR";
ulagatelj.ClientType = "DF";
ulagatelj.Individual = "Y";
ulagatelj.PreferredLanguage = "C";
ulagatelj.TaxMunicipality = 133;
ulagatelj.TaxCategory = 1;
ulagatelj.DomicileCountryCode = "HR";
ulagatelj.DomicileAddress = "ULICA 5";
ulagatelj.DomicileTown = "SPLIT";
ulagatelj.DomicilePostCode = "21000";
ulagatelj.BankAccountNo = "12345241361";
ulagatelj.BankVBDI = "1234567";
ulagatelj.IBAN = "HR99123456712345241361";
ulagatelj.PaymentType = "B";
ulagatelj.DateOfIncorporation = DateTime.Parse("1970/01/01");
//poziv funkcije
double Result =mojservis.SDA_AddPendingClient(ulagatelj,ref status);
if (Result ==1)
{
    MessageBox.Show("Predbilježba uspjela");
}
```

Upis zahtjeva za registraciju

Ova funkcija nudi mogućnost upisa zahtjeva za registraciju. Poziv funkcije se ostvaruje prosljeđivanjem svih podataka potrebnih za upis zahtjeva za registraciju. Kao rezultat funkcije vraća se informacija o uspješnosti upisa zahtjeva za registraciju (broj zahtjeva) ili u slučaju neuspješnosti razlog zbog kojeg upis nije uspio.

Obavezna polja iz strukture potreban za upis su: ISIN, RequestQty, FromAccountID, ToAccountID, ConfirmationMessage.

- ReturnDate, AccountIdBeneficiary, Remarks, DealPrice, DealAmount, Tax, Commission, Fees, CertifiedSell, TradeAllocation, ForeignBIC, ForeignAccountId

Nespomenuta polja iz strukture *InstitutionalDelivery* se ne koriste kod upisa.

Kod upisa instrukcije za namiru preko aplikativnog sučelja, instrukcija se upisuje u statusu Odobrena.

Definicija:

SDA_AddInstruction (*Instrukcija* As *InstitutionalDelivery* , *status* As *string*) As *double*

Ulazni parametri: *Instrukcija*– struktura (*InstitutionalDelivery*) za slanje podataka za upis u kojoj se koristite ranije navedena polja
Status – status pozvane funkcije

Povratna vrijednost: Redni broj instrukcije ako je upis uspješan.

Primjer:

```
// instanciranje klase te prijava korisnika
//...
string status;
//punjenje strukture
InstitutionalDelivery upis = new InstitutionalDelivery();
upis.ISIN = "HRPLVARA0004";
upis.DeliverReceive = "D";
upis.DeliveryType = "F";
upis.SettlementInstructionType = "";
upis.TransactionType = "TRAD";
upis.TradeDate = DateTime.Parse("2007/11/01");
upis.DeliverDate = DateTime.Parse("2007/11/01");
upis.Quantity = 15;
upis.SettlementAmount = 0;
upis.SettlementCurrency="HRK";
upis.ParticipantIdBeneficiary="BR02";
upis.ParticipantTypeBeneficiary="B";
upis.AccountID = 5187753;

//poziv funkcije
double Result = mojservis.SDA_AddInstruction(upis, ref status);
if (Result >0)
{
    MessageBox.Show("Instrukcija upisana pod brojem:"+Result.ToString());
}
```

Upis izravnog odobrenja

Ova funkcija nudi mogućnost upisa instrukcije za namiru primitka/ispоруke kao odgovor na instrukciju suprotne strane (sa suprotnim uvjetima). Poziv funkcije se ostvaruje prosljeđivanjem parametara u strukturi (*InstitutionalDelivery*) koji jednoznačno određuju instrukciju za namiru za koju se želi upisati izravno odobrenje (broj instrukcije suprotne ugovorne strane, datum namire i oznaka primitak/ispоруka) i broj računa vrijednosnih papira na koji se obavlja primitak/ispоруka. Kao rezultat funkcije se vraća informacija o uspješnosti upisa instrukcije (broj instrukcije) ili u slučaju neuspješnosti razlog zbog kojeg upis nije uspio.

Obavezni parametri za upis instrukcije su:

- IdReference, DeliverReceive, DeliveryDate, AccountId

Ostala polja iz strukture *InstitutionalDelivery* se ne koriste kod upisa izravnog odobrenja.

Definicija:

SDA_AuthorizeInstruction (Instrukcija As InstitutionalDelivery , status As string) As double

Ulazni parametri: *Instrukcija– struktura (InstitutionalDelivery) za slanje podataka za upis u kojoj se koristite ranije navedena polja*
Status – status pozvane funkcije

Povratna vrijednost: Redni broj instrukcije ako je upis uspješan.

Primjer:

```
// instanciranje klase te prijava korisnika
//...
string status;
//punjenje strukture
InstitutionalDelivery upis = new InstitutionalDelivery();
upis.IdReference = 12345;
upis.DeliverReceive = "R";
upis.DeliverDate = DateTime.Parse("2007/11/01");
upis.AccountID = 5187753;

//poziv funkcije
double Result = mojserwis.SDA_AuthorizeInstruction(upis, ref status);
if (Result >0)
{
    MessageBox.Show("Instrukcija upisana pod brojem:"+Result.ToString());
}
```


Verifikacija pročitanih poruka (poslanih članu kroz sustav SKDD-a)

Ova funkcija nudi mogućnost verifikacije pročitanih poruka odnosno označavanje poruka koje su pročitane. Poziv funkcije se ostvaruje proslijeđivanjem broja poruke koja se želi označiti kao pročitana. Kao rezultat funkcije vraća se informacija o uspješnosti verifikacije poruke ili u slučaju neuspješnosti razlog zbog kojeg verifikacija nije uspjela.

Definicija:

SDA_SetMessageAsRead (*MessageNo* As **int**) As **int**

Ulazni parametri: *MessageNo* – broj poruke koju se želi označiti kao pročitano

Povratna vrijednost: 0 – ako je poziv uspješan

Primjer:

```
// instanciranje klase te prijava korisnika
//...

//poziv funkcije
int Result = mojservis.SDA_SetMessageAsRead(123456);
if (Result ==0)
{
    MessageBox.Show("Poruka označena kao pročitana");
}
```

Brisanje alokacije

Ova funkcija nudi mogućnost brisanja alokacije. Poziv funkcije se ostvaruje prosljeđivanjem podataka potrebnih za jednoznačnu identifikaciju alokacije korištenjem pripadajuće strukture (*Allocation*). Kao rezultat funkcije vraća se informacija o uspješnosti brisanja (vrijednost 2) alokacije ili u slučaju neuspješnosti razlog zbog kojeg brisanje nije uspjelo.

Obavezna polja su:

- TradeDate, CDSReference, BuySell, LineNumber

Definicija:

SDA_SetDeleteAllocation (*Alokacija* As **Allocation** , *status* As **string**) As **double**

Ulazni parametri: *Alokacija*– struktura (*Alokacija*) za slanje podataka za brisanje alokacije

Status – status pozvane funkcije

Povratna vrijednost: Vrijednost 2 ako je funkcija uspješno provedena.

Primjer:

```
// instanciranje klase te prijava korisnika
//...
string status;
//punjenje strukture
Allocation alokacija = new Allocation();
alokacija.CdsReference=83;
alokacija.TradeDate=DateTime.Parse("2005/8/11");
alokacija.BuySell="B";
alokacija.LineNumber=10;
//poziv funkcije
double Result =mojservis.SDA_DeleteAllocation(alokacija,ref status);
if (Result ==2)
{
    MessageBox.Show("Alokacija obrisana");
}
```

Otkazivanje zahtjeva za registraciju

Ova funkcija nudi mogućnost brisanja zahtjeva za registraciju. Poziv funkcije se ostvaruje prosljeđivanjem broja zahtjeva. Kao rezultat funkcije vraća se informacija o uspješnosti otkazivanja (vrijednost 0) zahtjeva ili u slučaju neuspješnosti razlog zbog kojeg otkazivanje nije uspjelo. Zahtjev je moguće otkazati samo ako ima inicijalni status (Zahtjev podnesen – 1).

Definicija:

SDA_ CancelRegistrationRequest (*RegistrationNumber* As Int32 , *status* As string) As double

Ulazni parametri: *RegistrationNumber* – broj zahtjeva koji se želi otkazati

Status – status pozvane funkcije

Povratna vrijednost: Vrijednost 0 ako je funkcija uspješno provedena.

Primjer:

```
// instanciranje klase te prijava korisnika
//...

//poziv funkcije
double Result = mojservis.SDA_CancelRegistrationRequest(123456);
if (Result ==0)
{
    MessageBox.Show("Zahtjev otkazan");
}
```

Upis računa

Ova funkcija nudi mogućnost upisa sljedećih vrsta računa:

- Račun portfelja (RP) od strane člana brokera i člana društva za upravljanje računima portfelja
- Skrbničkih računa (IM – skrbnički račun na ime, PZ – skrbnički račun pod zaporkom, ZB – zbirni skrbnički račun) od strane člana skrbnika.

Kao rezultat funkcije vraća se informacija o oznaci novonastalog računa ako je poziv uspješan (vrijednost > 0) ili u slučaju neuspješnosti razlog zbog kojeg poziv nije uspio (vrijednost <0).

Definicija:

SDA_AddPendingAccount(Racun As PendingAccount, status As string) As double

Ulazni parametri: PendingAccount – struktura s podacima za upis računa

Status – status pozvane funkcije

Povratna vrijednost: Vrijednost >0 ako je funkcija uspješno provedena.

Primjer:

```
// instanciranje klase te prijava korisnika
//...
string status;
//punjenje strukture
PendingAccount upis = new PendingAccount();
upis.AccountType = "ZB";
upis.Nationality = "HR";
upis.Residence = "HR";
upis.AccountName = "Zbirni račun XYZ";
upis.BankSubAccountNo = "1239686-123";
upis.Remarks = "Upis prebilježbe";
upis.CAPaymentAccount = 1;
upis.T2SAccountID = SHREURXXXXXXXXXX;
//poziv funkcije
double povrat = mySvc.SDA_AddPendingAccount(upis, ref status);
if (povrat > 0)
{
    MessageBox.Show("Upisan novi račun s oznakom: " + povrat.ToString());
}
```

Brisanje računa

Ova funkcija nudi mogućnost brisanja računa. Poziv funkcije se ostvaruje prosljeđivanjem broja računa koji se želi brisati. Kao rezultat funkcije vraća se informacija o uspješnosti brisanja računa (vrijednost 0) ili u slučaju neuspješnosti razlog zbog kojeg brisanje nije uspjelo.

Račun je moguće brisati samo ako na njemu ne postoje vrijednosni papiri i ako nije raspoređen u neku javnu ponudu.

Definicija:

SDA_DeleteAccount(*AccountID* As Int32 , *status* As int) As double

Ulazni parametri: *AccountID* – oznaka računa vrijednosnih papira

Status – status pozvane funkcije

Povratna vrijednost: Vrijednost 0 ako je funkcija uspješno provedena

Primjer:

```
// instanciranje klase te prijava korisnika
//...

//poziv funkcije
double Result = mojservis.SDA_DeleteAccount(123456);
if (Result ==0)
{
    MessageBox.Show("Račun izbrisan");
}
```

Provedba zbirnog izmirenja

Ova funkcija nudi mogućnost izmirenja novčanih obveza u poslovima namire transakcija razvrstanih u pojedinačnu namiru uz plaćanje i Naizmjeničnih transakcija.

Kao rezultat funkcije vraća se informacija o uspješnosti provedene transakcije:

- OK
- Obaveza po pozivu na broj poziv na broj je već podmirena
- Nepostojeći poziv na broj TFT obveze plaćanja poziv na broj

Definicija:

SDA_TFT_payment(*Racun* As string, *status* As string) As double

Ulazni parametri: *Poziv_na_broj* – transakcija

Status – status pozvane funkcije

Povratna vrijednost: Vrijednost >0 ako je funkcija uspješno provedena.

Primjer:

```
// instanciranje klase te prijava korisnika
//...

//poziv funkcije
double povrat = mySvc.SDA_TFT_payment(transakcija, ref status);
if (povrat > 0)
{
    MessageBox.Show("OK");
}
```

Statusni kodovi

Status	Brojčana oznaka	Opis
errLoginFailed	-100	Nije dobar username/password
errUnauthorized	-105	Poziv procedure kada korisnik nije logiran
errAlreadyLogged	-110	Pokušaj prijave kada je korisnik već prijavljen
errRefused	-111	Greška u bazi
errWorkingHours	-106	Poziv funkcije van radnog vremena
errServiceNotActivated	-101	Član nije aktivirao uslugu korištenja aplikativnog sučelja ili je status člana u sustavu SKDD-a različit od „Dozvoljeno poslovanje“
errParamMissing	-1	Nedostaje obavezan parametar ili je neispravna vrijednost parametra
responseOK	0	Sve je u redu
responseNothingChanged	1	Pozvana funkcija SDA_SetMessageAsRead je pokušala označiti poruke pročitanim, ali nije uspjela.
errUnexistentAccount	-200	Ne postoji račun za koji se traži podatak
errUnexistentClient	-205	Ne postoji klijent za koji se traži podatak
errUnexistentTransaction	-210	Ne postoji transakcija za koju se traži podatak
errUnexistentSecurity	-215	Ne postoji VP za koji se traži podatak
errUnexistentRegistrationReq	-220	Ne postoji zahtjev za registraciju
errUnexistentAllocation	-225	Ne postoji alokacija za koju se traži podatak

Ova Tehnička dokumentacija za korištenje aplikativnog sučelja SKDD-a, koju je Uprava SKDD-a 11.02.2022. donijela na svojoj 35. sjednici, stupa na snagu i primjenjuje se od 11. rujna 2023. Stupanjem na snagu ove Tehničke dokumentacije stavlja se van snage Tehnička dokumentacija od 11. veljače 2012. godine.

UPRAVA
 Broj: U-150/23-6
 Zagreb, 7. 9. 2023.

PREDSJEDNICA UPRAVE

mr. sc. Dora Matošić