

DopinusApiWrapper DAW.EXE

Wrapper za klice DopinusE2 JSON-RPC API-jev, za aplikacije, ki tega ne zmorejo direktno:

Wrapper je napravljen v obliki .EXE datoteke (DopinusApiWrapper - DAW.EXE), ki jo pokličete iz svojih aplikacij. Prvi parameter .exe-ju vedno pove klic katerega dejanskega API-ja oziroma katero metodo naj simulira, ostali parametri podajajo ostale potrebne podatke za klic tega API-ja. Prvi parameter je vedno obvezen in je v tem dokumentu napisan s pokončnimi (nonitalic) črkami, ostali parametri so *italic* in so dodatno opisani. Parametri v oglatih oklepajih so opsijski. Response s serverske strani se zapiše v datoteko, navedeno v .config datoteki (primer na koncu dokumenta). Nastavite iz datoteke se lahko overwrite-ajo pri samem klicu avtentikacije. Response v datoteki je v splošnem popolnoma enak, kot bi API klicali klasično po JSON-RPC 2.0 specifikaciji, pri nekaterih »metodah« (ieracun, peracun in ier_niok) pa je response na željo programerjev, ki jim strukturirani zapisi povzročajo več težav kot koristi, »predelan« tako, kot bi šlo za .CSV datoteko.

Pri testiranju v ukazni vrstici lahko ob klicanju .EXE-ja vedno navedete še /console. V tem primeru se vam v ukazni konzoli na ekran izpišejo tako zahtevki, ki jih EXE pošlje serverju, kot response, ki se sicer še vedno zapiše v response datoteko.

Prvi klic je seveda prijava v sistem, ki jo v svojih programih lahko napravite povsem nevidno, je pa popolnoma enakovredna interaktivni prijavi v našo storitev.

avtentikacija: **DAW authenticate [username password] [responsefilepath]**

Podatki za avtentikacijo se lahko opsijsko vpišejo direktno kot parameter, sicer se preberejo iz .config fajla. Če se jih ne najde niti tam, se vrne authorisation failure. Geslo se nikoli ne prenaša po omrežju. On-line avtentikacija v servis Dopinus sicer podpira tudi Open-ID (google, facebook ipd. zaupanja vredni servisi) ter avtentikacijo s serverskimi ali lokalnimi certifikati. V primeru potrebe (delovanje na terminalskih strežnikih ipd.) je pri avtentikaciji možno podati pot in ime fajla za sprejem response-a, ki overwrite-a splošno nastavitve iz

.config fajla. Ponoven klic exeja v okviru veljavnosti te iste avtentikacije bo response vračal v tako nastavljen fajl. V fajlu je vedno le response zadnjega klica. Klici API-jev so sicer povsem asinhroni, toda zaradi načina komunikacije preko response fajla predlagamo izključno sinhrono uporabo (najprej preverite response prejšnjega zahtevka in šele nato ponovno izvedete nov API oziroma izvedete nov klic EXE-ja).

odjava : **DAW signout**

Po koncu dela je smiselno napraviti odjavo in s tem preprečiti nepooblaščen dostop do izdanih in prejetih faktur.

Preveri verzijo : **DAW version**

S tem ukazom lahko preverite trenutno verzijo daw.exe in aktualno verzijo, pripravljeno na strežnikih za avtomatski update. Tega v splošnem ne potrebujete, saj se nadgradnja zgodi povsem avtomatsko v okviru ukaza authenticate. Če zaradi kakršnegakoli razloga želite avtomatsko nadgrajevanje izklopiti, to storite v .config datoteki, tako, da nastavite "AutoUpdate" value="false". V tem primeru lahko nadgradnjo sprožite z ukazom

Osveži: **DAW update**

Help: **DAW help**

S tem ukazom se izpišejo vse metode daw.exe. Prijava ni nujno potrebna.

DAW help *ime_metode*

S tem ukazom se izpiše kratek opis metode in njenih parametrov (seznam in opis)

Za oddajo eRačuna se potrebuje naslednje:

oddaja eRačuna: **DAW ieracun filename [S#scanfile] [V#kvtxtfile] [open] [autosend]**

ieracun – kot izdani elektronski račun. Je specifično pripravljen za upload XML računa in hkraten zapis v tabelo računov. Operacija se zgodi kot nedeljiva celota, tako da smo lahko prepričani, da vsak zapis v tabeli dejansko dostopa tudi do uploadanega XML fajla. V bistvu gre za sestavljene tri metode – file upload, datacreate in xmlparse, ki se zgodijo kot celota: zgodi se create zapisa v tabelo izdanih faktur, zapisovanje url-ja za dostop do uploadanega računa, razparsanje XML-ja in prepis izbranih podatkov o fakturi in njenih detailov, morebitni dodatni specifični podatki iz ERP-a pošiljatelja pa se lahko dodatno zapišejo s pripravo *kvtxtfile-a*(*Key-Value txt fajl*).

Ker imamo pri metodi ieracun več opsijskih parametrov, ki jih lahko navedemo v poljubnem vrstnem redu, jih med seboj ločujemo s predponami (V#, S#, E#, F#, P#)

- *filename* je file z lokalnega diska, ki ga želimo uploadati v Dopinus Key-Value store. Pri eRačunu je to xml datoteka v formatu eSlog 1.5 ali 1.6. Url za dostop do file-a se bo avtomatsko zapisal v polje Racun. Parameter je obvezen, zato nima predpone.
- *S#scanfile* je opsijska pot in ime datoteke na lokalnem disku, pripravljene v različnih formatih, ki jih podpirajo programi za skeniranje. Najpogosteje se pripravijo PDF in JPG oblike vizualizirane fakture. Če je datoteka podana, se uploada v Dopinus Key- Value store, k fakturi pa se zapiše kot prva priloga, ki je kot taka tudi na ogled v uporabniškem vmesniku za ogled faktur, hkrati pa se URL do uploadane PDF/JPG datoteke zapiše tudi v polje PDF v tabelo IF_eracuni.

Parameter *scanfile* je po novem lahko seznam večih datotek, ločen z vejico (če je seznam brez presledkov, sicer mora biti cel seznam v narekovajih). Če bi se tako presegla omejitev ukazne vrstice (128 znakov) je mogoče seznam napisati v datoteko, katere ime je nastavljeno v .config in kot parameter podati to datoteko. Po dafaultu je nastavljena na ime »DAW_Seznam_prilog.txt«.

Primeri veljavnih klicev:

```
>daw ieracun eracun.xml S#prva_priloga.pdf,druga_priloga.txt  
>daw ieracun eracun.xml S#"prva_priloga.pdf, druga_priloga.txt"  
>daw ieracun eracun.xml S#DAW_Seznam_prilog.txt
```

Predvideva se, da je prva priloga vedno vizualizirana faktura (PDF, JPG ipd.). URL do te prve priloge se bo zapisal tudi k fakturi (v glavo fakture) v polje PDF.

Če je v seznamu »vizualizacija«, mora biti torej podana na prvem mestu.

Daw po novem podpira konvertiranje datotek v pdf format - tako za sam eračun preko vizualizacije, ki je navedena v eračunu, kot tudi za vse priloge. Pogoj je, da imate to storitev konvertiranja vklopljeno. Za konvertiranje se uporablja plačljiv servis, zato je potrebno vklop konvertiranja dogovoriti z nami. Vsak konvert datoteke se beleži in na osnovi tega seznama se bo naredil obračun konvertiranja. Zaradi konvertiranja bo prenos eračuna časovno malce bolj potraten – za en običajen eračun cca 4 sec, seveda pa je odvisno od velikosti izvorne datoteke.

Možna sta dva načina vključitve konverta:

- avtomatski

V tem primeru se v config datoteko doda nastavev za avtomatiko in se jo nastavi na true, kar pomeni, da se vse priloge, ki so dogovorjenih formatov (to je potrebno dogovoriti v času vklopa storitve konvertiranja), avtomatsko konvertirajo v pdf. Na ta način daw klicev ni potrebno nič spreminjati.

Izvirnega dokumenta ne hranimo, hranimo le ciljni dokument, torej pdf.

V kolikor pa bi želeli, da se tudi sam eračun konvertira preko vizualizacije (.xslt dokumenta) eračuna v pdf in da naj to postane dejansko pravi pdf oz. prva priloga, pa je

DopinusApiWrapper DAW.EXE

klic potrebno spremeniti in sicer s ključno besedo *c2pdf*. V tem primeru se datoteka xml (eračun) ohrani, naredi se dodatni pdf dokument.

Primer klicev:

- daw ieracun eracun.xml S#c2pdf

Eračun se pretvori v pdf, doda se zapis v priloge. Dokumenti, ki se naložijo: eracun.xml, eracun.pdf.

- daw ieracun eracun.xml S#c2pdf, priloga.rtf

Tako eračun kot priloga se pretvorita v pdf, če so nastavitve tako urejene. Dokumenti, ki se naložijo: eracun.xml, eracun.pdf, priloga.pdf

- ročni

Potrebno spremeniti klic daw-a in sicer s pomočjo kombinacije stringa *!c*. Na ta način sami določate katera datoteka naj se pretvori.

Izvornega dokumenta ne hranimo, shranimo le ciljni dokument, torej pdf.

Primer klica:

- daw ieracun eracun.xml S#priloga1.rtf!c

Ta priloga se bo konvertirala v pdf ne glede na dogovorjene formate za konvert, pogoj je le, da je konvert vklopljen. Dokumenti, ki se naložijo: eracun.xml, priloga1.pdf

Tudi pri ročnem lahko sam eračun konvertiramo v pdf in sicer na dva načina:

- daw ieracun eracun.xml S#c2pdf, priloga1.rtf!c, priloga2.pdf

ali

- daw ieracun eracun.xml S#eracun.xml!c, priloga1.rtf!c, priloga2.pdf

Če v eračunu ni navedene vizualizacije ali pa če je, a se je ne more doseči, vzame našo default vizualizacijo. V tem primeru se datoteka xml (eračun) ohrani, naredi se dodatni pdf dokument.

Podobno kot velja za navajanje posameznih prilog, velja tudi za seznam prilog, kjer lahko posamezni prilogi iz seznama dodelimo ključno besedo *!c*. **Kombinacija DAW_Seznam_prilog.txt!c ni podprta!**

- *V#kvtxtfile* je opsijsko ime txt fajla, v katerem so specifični `FieldName : Value` pari, ki se jih ne da sparsati iz XML fakture, ampak obstajajo samo v informacijskem sistemu, ki je poslal fakture, jih pa programer želi zapisati k fakturi ali pa želi izjemoma overwrite-atikakšen podatek iz XML fakture.
(trenutno servis pravilno interpretira le šumnike po UTF-8 in 1250 encoding-u, po potrebi lahko dodamo še kakšnega – recimo 852 ali celo stari, modificirani 437)
- Beseda *open* je dodatni opsijski parameter, ki pove ali se naj takoj po uploadu odpre še UI v internet browserju, ki omogoča pregled pravkar uploadane fakture in dodajanje prilog nanjo. Parameter je opsijski, saj se postopek z uporabniškim vmesnikom za dodajanje prilog in pošiljanje k naslovniku lahko odpre kasneje, ko je naloženih že več faktur (glej opis metode/parametra *open*).
- Beseda *autosend* je dodatni opsijski parameter, ki pove ali se naj takoj po uploadu fakture (in priloge) takoj odpošlje. Faktura, ki pride v naš servis, je lahko podpisana ali nepodpisana. Če je nepodpisana, jo bo naš servis avtomatsko podpisal in takoj odposlal. Zanj se ne pričakuje ročno dodajanje dodatnih prilog, ravno tako se v tem primeru ne pričakuje postopek potrjevanja in podpisovanja v samem servisu. V primeru opcije *autosend* se faktura najprej postavi v status »Pripravljen«, da jo bo postopek za oddajo lahko prepoznal kot primerno. Če se bo postopek oddaje izvedel brez napak, dobi takšna faktura takoj status »Oddan«.

DopinusApiWrapper DAW.EXE

pregled IF: **DAW open url**

Parameter *url* je relativno napisan internetni naslov (URL) za proženje postopka ali prikaz (če je v browserju ni mogoče prikazati pa download) posamezne datoteke, ki se zažene v internetnem pregledovalniku. Za proženje postopka za pregled izdanih faktur se kot url praviloma navede postopek *gui/isedico/izdanieracuni*, ki uporabniku ne omogoča dostopa do avtentikacijske vrstice, ampak je avtomatsko avtentificiran tako kot ga je avtentificiral iz vašega programa prožen DAW.EXE.

Funkcionalnost postopka je odvisna od vloge oziroma pravic prijavljene osebe. Pooblaščen(a) osebe razporedijo izdane fakture po vnaprej predvidenih krogotokih drugim osebam, ki morebiti še potrjujejo/podpisujejo fakturo ali prilagajajo priloge. V tem istem postopku osebe, ki imajo te pravice, potrjujejo ali podpisujejo fakturo in/ali odpošiljajo potrjene fakture naslovniku.

Zase in za uporabnike, ki sicer niso uporabniki vaše aplikacije, a bodo fakturam dodajali priloge ali kakorkoli drugače sodelovali pri postopku likvidacije ali so morda vzdrževalci, ki bodo pomagali različnim drugim uporabnikom, pa uporabite relativni naslov *web/isedico/izdanieracuni*, ki omogoča interaktivno prijavo v servis. Celoten naslov za prijavo v servis neodvisno od integriranega wrapperja je potrebno sestaviti iz podatkov v .config datoteki in tu navedenega URL – za postopek v zvezi z izdanimi računi recimo **<https://eracuni.dopinus.com/web/isedico/izdanieracuni>**

pregled statusa: **DAW ier_niok**

Parameter *ier_niok* pove EXE-ju, da naj sproži metodo, ki bo vrnila seznam izdanih eračunov, s katerimi je nekaj narobe (Izdani ERAčuni, ki Niso OK). V response datoteko se vrne seznam v obliki *id;status;opis_napake* za vse fakture, ki so v statusu »Napaka-*« ali »Zavrnilen-*«. Tako enostavno lahko napačne in zavrnjene račune označite tudi v svojih sistemih (glej dokument v zvezi s statusi izdanih računov) in se nato odločite kaj z njimi.

To je pravzaprav vse kar se potrebuje za običajno delo z izdanimi eRačuni. Za boljše razumevanje postopka naj razložimo, da »metoda« ieracun napravi najprej upload XML datoteke, nato še PDF/JPG slike fakture (če je prisotna oziroma, če je pri klicu navedena kot parameter), nato se kreira zapis v tabelo izdanih faktur. Vse to se izvede kot nedeljiv paket. Če se ta paket zahtevkov uspešno izvede, se nato XML datoteka še razparsa in se izbrana množica podatkov prepíše v novokreiran zapis, nato se ti podatki morebiti še dopolnijo ali overwrite-aajo še s podatki iz datoteke *kvtxtfile* (ki jo predhodno pripravi programer, če želi zapisati še kakšne specifične podatke iz ERP/PIS, recimo kakšno dodatno interno številko izdane fakture, drugačno od tiste v XML).

V primeru parametra *autosend* se v istem klicu pripravi še ovojnica in faktura se skupaj z morebitnimi prilogami še avtomatsko odpošlje. V tem primeru se v response datoteko zapiše še enovita identifikacijska številka fakture, ki jo je dodelil bodisi eden od UJP servisov ali pa jo je dodelila storitev *eracuni.dopinus.com*, če je bila faktura namenjena iz enega gospodarskega subjekta k drugemu gospodarskemu subjektu.

Rezultati izvedbe metod se v takšnem vrstnem redu zapišejo v response datoteko, da jih lahko programer v ERP/PIS pregleda oziroma prebere in po želji prepíše v »svoje« tabele (in tako ustvari stalno povezavo med podatki fakture v ERP/PIS in vizualizirano fakturo s prilogami v našem servisu).

OPIS oziroma razlaga RESPONSE DATOTEKE pri paketnem parametru »ieracun«

- **Authentication;OK;User.Name:<ime_osebe>;Organization.Title:<ime_organizacije>**

Če ima daw.exe veljavno avtentikacijo, se izpiše Authentication;OK; in v nadaljevanju podatki o prijavljeni osebi in organizaciji v katero je prijavljen. Če avtentikacija ni veljavna, se nadaljnje izvajanje prekine. Po preverjanju avtentikacije se zgodi še preverjanje ali morda na strežniku obstaja novejša verzija daw.exe.

DopinusApiWrapper DAW.EXE

Checking for updates...

Found version 1.0.0.0. No update available.

Če verzija obstaja in nastavitve "AutoUpdate" value="true", se na tem mestu avtomatsko izvede nadgradnja in vsi nadaljnji ukazi se nati izvajajo z najnovejšo verzijo.

- **UploadXML;OK;File/ID_datoteke (v key-value store-u)**

Del URL-ja (za drugim podpičjem) se zapiše k fakturi in preko njega se XML in morebiten PDF/JPG prikazujeta v uporabniškem vmesniku storitve. Če želite to vključiti v lastno aplikacijo, morate sestaviti absolutni URL do uploadane PDF/JPG datoteke (potrebno je združiti podatke iz .config datoteke in ta relativni URL (https://eracuni.dopinus.com/file/id_datoteke)).

- **UploadPDF; OK;File/ID_datoteke (v key-value store-u)**

Vse enako kot zgoraj, le da gre tu za morebitno prilogo »slike« fakture v PDF ali JPG obliki. Vrstica Upload (in končnica priloge) se zapiše za vsako od navedenih prilog.

- **IFCreate; OK;ID_fakture**

Najpomembnejši del. Glavni zapis fakture. Ta ID si je potrebno smiselno shraniti v svoje tabele, preko njega lahko kadarkoli pogledate vizualizirano fakture in priloge, preberete trenutni status fakture ipd.

- **IFCreateChild;OK;detail_tabela;število_dodanih_zapisov**

Kadar se ob parsanju fakture zgodijo še zapisi v druge (detail tabele, recimo v tabelo postavk) je za vsako tako tabelo not zapis, kjer se zapiše tabela in vanjo dodano število zapisov

- **KVModify;OK; ID_fakture**

Če je bila kot parameter podana še datoteka s Key-Value pari, s katero lahko programer v tabeli poslani v servis, zapiše še specifične podatke iz svoje aplikacije, se pojavi še takšna vrstica.

(ID je isti kot pri tretji alineji, saj gre za popravek podatkov te iste fakture)

Če je faktura poslana z opcijo autosend, so v response-u še naslednje vrstice

- **SignatureCheck;OK;NOTSIGNED; ID_fakture**

- **Signature;OK;**

- **Send;OK;ID_fakture**

Poslana faktura se v tem primeru najprej preveri ali je že podpisana. Če je, potem je na tretjem mestu SIGNED in naslednje vrstice ni, saj je ni bilo potrebno podpisati. Če NOTSIGNED, se podpiše in rezultat je v naslednji vrstici.

Če je v tretji vrstici namesto OK napisano SKIPPED, potem ta konkretna faktura ni bila avtomatsko poslana naprej, zato je potrebno v uporabniškem vmesniku preveriti njen status in polje opis napake, kjer je podrobno razložen vzrok.

Tudi, če vse fakture pošiljate z autosend, mora nekdo v podjetju imeti account, da se preko uporabniškega vmesnika lahko podrobneje ukvarja s problematičnimi in zavrženimi fakturami. Z ukazom »ier_niok« sicer lahko sami preberete statuse faktur in jih zapišete v svoje aplikacije.

Vsi ID so tipa GUID. Če se katerakoli operacija ne izvede pravilno, je namesto OK v response-u na tem mestu ERROR, namesto ID pa opis napake. (le pri Send je namesto ERROR navedeno SKIPPED, saj faktura, ki je prestala vse ostale teste in je podpisana ter pripravljena za oddajo, ostane v servisu, samo avtomatsko ni bila poslana in se po odpravi težave odpošlje ročno, preko uporabniškega vmesnika.

Primer (z vključenim autosend):

```
Authentication;OK;User.Name:Dušan Pondelak;Organization.Title:IS.Edico d.o.o.
```

```
Checking for updates...
```

```
Found version 1.0.0.0.
```

```
No update available.
```

```
UploadPDF;OK;File/ed41b38e-4421-4dcf-b579-ec19191dfadb
```

```
UploadXML;OK;File/a45a4e54-6f93-41ec-8a6d-b180019ef7fb
```

```
IFCreate;OK;99c1a3c1-df05-4799-9c3c-d40694fe4c8e
```

```
IFCreateChild;OK;IF_davki;2
```

```
KVModify;OK; 99c1a3c1-df05-4799-9c3c-d40694fe4c8e
```

```
SignatureCheck;OK;NOTSIGNED;99c1a3c1-df05-4799-9c3c-d40694fe4c8e
```

```
Signature;OK;
```

DopinusApiWrapper DAW.EXE

Send;SKIPPED;99c1a3c1-df05-4799-9c3c-d40694fe4c8e

Prejeti eRačuni

Mi smo storitev pripravili z mislijo na množico programerjev, ki bodo vsi izdelovali podobne postopke, ki bodo podpirali procese, ki so se do sedaj izvajali nad papirno obliko prejete fakture (logistika po podjetju, potrjevanje, podpisovanje, dodajanje prilog, opremljanje fakture s podatki stroškovnega razdelilnika ipd). Vse to jim želimo prihraniti, zato smo v storitev vključili postopek za ta namen. Poskušali smo ga zastaviti maksimalno splošno (univerzalno), zato postopek sloni na opisih delovnih tokov (workflow-ov), ki se lahko pripravijo za vsako podjetje posebej - hitro in enostavno. Da je ta postopek vključen v servis in dostopen v internetnem brskalniku je smiselno tudi zato, ker v tem postopku večkrat nastopajo tudi osebe, ki sicer nimajo dostopa do finančno računovodske aplikacije v podjetju (recimo tehnikalci, ki k fakturi dodajo razne tehnične priloge).

Ta postopek iz svoje aplikacije enostavno pokličete z naslednjim ukazom, ki bo internetnemu brskalniku podal avtentikacijsko sejo in v njem odprl naveden postopek.

pregled PF: **DAW open url**

Za proženje postopka za pregled in obvladovanje prejetih, a še ne v ERP/PIS prenesenih faktur, se kot url navede postopek *gui/isedico/prejetieracuni*. V tem postopku osebe, ki imajo ustrezne pravice, razporedijo fakture drugim osebam glede na predviden delovni tok dokumenta - za potrjevanje, podpisovanje, prilaganje ustreznih prilog in opremljanje fakture z delilnikom stroškov (stroškovna mesta, stroškovni nosilci, viri). Osebe, ki sodelujejo v krogotoku dokumenta, vidijo le fakture, ki so jim razporejene in lahko izvedejo operacije, ki se od njih pričakujejo v skladu s predhodno opisanim delovnim tokom. Ko je faktura ustrezno opremljena in predvidene osebe to potrdijo oziroma podpišejo, je takšna faktura pripravljena za prevzem v prejemnikov informacijski sistem.

Fakture se z UJP pobirajo ob 1.00h zjutraj (za prejšnji in tekoči dan) in ob 06:40h, 10:40h, 13:40h, ter ob 20:40h. Na tak način so v vmesniku avtomatsko vse fakture, ki so prispele pretekli dan ali pa so jih pošiljatelj poslali ponoči.

Če kdo želi bolj ažurne podatke, lahko »pobere« fakture »on-demand«, s klikom na ustrezno ikono v orodjarni (toolbar-u) v uporabniškem vmesniku. V seznam faktur bo tako dobil tudi fakture, ki so prispele v tekočem dnevu in so bile do zahteve že obdelane na UJP.

Ob vsakem prevzemu prejetih faktur iz UJP se avtomatsko osvežijo tudi statusi predhodno na UJP izdanih faktur.

Navodila za dejansko delo s tem postopkom so pripravljena med uporabniškimi navodili/dokumenti.

Ko se račun pridobi od izdajatelja v naš servis (po katerikoli elektronski poti že), dobi status »Nov«. Praviloma do konca procesa iz prvega odstavka pridobi status »Pripravljen« (pripravljen za prevzem v ERP/PIS). Če programerji svojim uporabnikom na strani prejemnika ne želijo integrirati tega postopka, lahko prevzamejo v ERP/PIS račun že kar v statusu »Nov«.

Za prevzem eračunov v ERP/PIS obstajata dva tehnično zelo različna načina. Prvi je namenjen programerjem, ki so (bodo) v svojih rešitvah sami pripravili postopke za uvoz XML računov, drugi pa programerjem, ki želijo to nalogo v celoti opraviti s pomočjo naših API-jev oziroma v tem primeru s pomočjo opisanega DopinusApiWrapperja (DAW.EXE).

V prvem primeru se fakture in priloge samo zapišejo na lokalni disk, pač nekam na zeleno mesto, kjer jih kasneje obdela (ustrezno razparsa in prepíše v svojo podatkovno bazo) poslovna programska oprema na strani prejemnika.

Za ta način potrebujete le izvesti naslednji klic:

prevzem XML: **DAW dl status dldir [XMLOnly]**

Parameter **dl** - kot okrajšava za download. Z njim enostavno vse prejete fakture v določenem statusu pretočimo na lokalni disk. Načelno se bodo v ERP/PIS prenašale fakture v statusu »Pripravljen«, ki ga bodo dobile, ko bo zaključen postopek likvidacije in potrjevanja. Vendar pa lahko z ukazom prenesete tudi fakture v kateremkoli drugem statusu, torej tudi fakture v

DopinusApiWrapper DAW.EXE

statusu »Nov«, kar je še posebej uporabno za podjetja, ki postopka likvidature v naši storitvi ne bodo uporabljale, ampak bodo fakture samo prevzele in postopek likvidaturo izvajale šele po importu v svojo aplikacijo.

status – z tem drugim parametrom poveste katere fakture želite prenesti na disk. Praviloma bodo to ali status »Pripravljen« ali status »Nov«.

dldir – tretji parameter je ime direktorija kamor se bodo prepisovale fakture in priloge (Primer: »c:\eracuni\prejeti«).

XMLonly – je opcijski parameter s katerim povemo, da se naj na lokalni disk prenesejo samo XML fakture, ne pa tudi njihove priloge.

Programerji preneseni XML račun sami razparsajo in podatke prepisujejo v podatkovno bazo, vzpostavi se povezava (preko id v imenu fakture, kot je razloženo kasneje) za dostop do virtualizirane fakture vključno z vsemi prilogami.

Fakture se pri zapisovanju na lokalni direktorij poimenujejo na naslednji način:

PFyyMMddhhmss_id_IDServisIdent_nnn_OrigImeDatoteke, pri čemer posamezni deli pomenijo naslednje:

- Predpona »PF«,
- yyMMddhhmss - leto, dan, mesec, ura, minuta in sekunda iz podatka (tipa datetime) o času prejema fakture v naš servis (ko se je odvirtela obdelava na UJP ali medbančna obdelava in se je faktura dejansko pojavila na strani prejelnika)
- id – enolična identifikacijska številka tipa GUID
- 'ID' + ServisIdent
 - ID je konstanta
 - ServisIdent je enolična številka, ki jo dokumentu priredi servis (recimo UJP, če gre dokument skozi UJP, servis EračuniEdico, če sta izdajatelj in prejemnik oba uporabnika naše storitve ipd.).
- nnn – trimestrna zaporedna številka, pri čemer ima račun vedno številko 000, prva priloga 001, druga 002...itd.
- za četrtem podčrtajem pa se nahaja originalno ime datoteke, pri čemer ima račun vedno ime »Faktura.xml«, priloge pa obdržijo originalno ime kot jim ga je določil izdajatelj ali kot ga je priloga določila oseba, ki je dodala prilogo na strani prejelnika. Tako smo se odločili, da je v primerih posebnih prilog olajšana komunikacija med izdajateljem in prejemnikom.

Primer:

PF150617211026_ede11d46-bf0c-4030-9658-99c2fc1114f2_IDUjpZapSt_000_Faktura.xml

Res, da je takšno ime prilično dolgo, vendar s tem zagotovimo, da programerjem na strani prejelnika, ki sami razparsajo fakture in podatke sami zapišejo v svoj ERP/PIS, ni potrebno pošiljati več nobenega drugega ukaza, saj lahko samo iz imena datoteke pridobijo čas prejema, predvsem pa id v našem servisu, preko katerega se lahko vzpostavi povezava z vizualizirano fakture in vsemi njenimi prilogami ter ostalimi podatki, ki jih o fakturi spremljamo v naši storitvi.

Vsekakor predlagamo, da po prevzemu fakture v ERP fakture in priloge preimenujete v vam bolj prijazno ime in iz tega direktorija prestavite na neko drugo mesto. Na tem direktoriju se naj vedno nahajajo le fakture, ki ste jih že prevzeli iz našega servisa, niste pa jih še prepisali v ERP/PIS.

Fakture, ki se prenesejo na disk, avtomatsko dobijo status »Prevzet-ERP«, zato se z naslednjim klicem ne bodo več prenesle.

prevzem XML: **DAW dlonly status dldir [XMLonly]**

Razlika napram dl parametru je le v tem, da se fakturam, ki se prenesejo na disk, ne spremeni status, zato jih lahko večkrat prenesemo.

Programerjem, ki delajo po drugem načinu (s pomočjo naših API-jev) in ne parsajo sami fakture, XML-ja in priloge ni potrebno nikoli prenašati na lokalni disk.

Prevzem faktur na drugi način – XML se razparsa v naši storitvi, nakar se lahko zgodijo zelo različne stvari:

1. Programer lahko preko response datoteke prevzame »Key-Value pare za vse svoje tabele in na njihovi osnovi sam pripravi zapis podatkov v kakršnikoli tehnologiji že
2. Podatki razparsanega XML se zapišejo v serversko kopijo lokalnih (ERP) tabel, nato:
 - 2.1 Programer prevzame popolne SQL-e (insert stavke) za vse svoje udeležene lokalne tabele
 - 2.2 Klicu Daw.exe se kot parameter poda connect string do lokalne podatkovne baze, ki pripravljene SQL (insert) stavke tudi izvede.
 - 2.3 Na željo nekaterih integratorjev se namesto SQL-ov pripravljajo CSV tekstovne datoteke.

V vseh primerih je predhodno potrebno pripraviti navodila za parsanje – xml datoteko v kateri so naštetih fieldi tabel in ustrezni xpath izrazi za pridobitev ustreznih podatkov iz podane XML fakture.

Prvi korak: **DAW perxmltodb status parsefile main_tabela [tofile]**

»Metoda« **perxmltodb** – kot prejeti elektronski račun, XML to DataBase. Je specifično pripravljen za konkretno pomoč pri prepisu prejetih faktur iz naše storitve v informacijski sistem uporabnika. Parameter **status** pove katere fakture želimo prenesti v ERP. Odvisno od tega ali ponudnik programske opreme v svoj SW integrira tudi uporabniške postopke naše storitve, je ta status »Nov« ali »Pripravljen«. V prvem primeru se poberejo fakture takoj po prejemu, v drugem po tem, ko so se v storitvi izvedli vsi postopki potrjevanja, dopolnjevanja...skratka likvidature...in se je status faktur spremenil v »Pripravljen« (za prevzem v ERP).

Priprava podatkov se zgodi na podlagi opisov v datoteki **parsefile**, v kateri se z uporabo xpath sintakse pove kateri podatek iz XML računa se zapiše v katero polje katere tabele. Specifična datoteka za vsakega ponudnika se predhodno pripravi in uploada v servis, pri čemer se pridobi unikatni id. Ponudnik ob klicu poda parameter v obliki relativnega url do datoteke predhodno naložene v Key-Value store naše storitve (File/<id_datoteke>). Datoteko pripravimo skupaj s ponudnikom informacijskega sistema. V primeru sprememb na strani informacijskega sistema se parsefile enostavno popravi in ponovno uploada.

Parameter **main_table** predstavlja master tabelo, v **parsefile** je lahko naštetih še poljubno število detail tabel. Master tabela in vse detail tabele (v bistvu kopija tabel, kot jih imajo v svojih informacijskih sistemih uporabniki konkretnega ponudnika) se materializirajo v naši storitvi in ob tem klicu napolnijo s podatki, ki jih želimo v lokalni bazi pri konkretnem uporabniku storitve.

V primeru opsijskega parametra, konstante **tofile**, materializacija tabel v storitvi ni potrebna, saj se v response napravijo le zapisi v obliki fieldName:Value parov, na podlagi katerih lahko programerji informacijskega sistema na strani prejelnika sami pripravijo zapisovanje teh podatkov v svojo bazo.

Ob uspešnem branju elektronskih faktur in zapisu v podatkovno bazo oziroma kreiranju response datoteke s pripravljenimi vrednostmi, se status prebranih faktur spremeni, zato se ob naslednjem klicu več ne upoštevajo.

Drugi korak: **DAW tolocaldb seznam_tabel SQL/EXE [connect_sting]**

»Metoda« **tolocaldb** – kot prenos poljubnih tabel pripravljenih v naši storitvi v lokalno, uporabniško podatkovno tabelo (To Local DataBase).

Pri parametru **seznam_tabel** se naštejejo tabele (ločene z vejico in cel parameter v narekovajih), katerih vsebino želimo naenkrat prenesti v poljubno uporabniško podatkovno bazo. To lahko naredimo na dva načina – da se (če pri klicu podamo parameter **SQL**) pripravijo SQL insert stavki, prirejeni za konkreten DBMS, ki jih preko response datoteke »primejo« programerji na lokalni strani in jih izvedejo sami, ali da (s parametrom **EXE**) povemo, da naj Daw.exe sam izvede tudi te stavke, za kar seveda potrebuje pridobiti **connect_string** za dostop do lokalne podatkovne baze.

Avtomatizirana izvedba prenosa podatkov prejetih faktur v podatkovno bazo uporabnika na tak način je zagotovljena le v primeru, da je organizacija podatkovnega modela v ERP/PIS taka, da vsebuje le eno master tabelo (praviloma glave faktur) in poljubno število njenih detail tabel (postavke, davki, delilnik stroškov itd...), vseh na tem istem, prvem nivoju. V primeru, da je podatkovni model gnezden večnivojsko (recimo tabela

DopinusApiWrapper DAW.EXE

postavk ima še nadaljnje detaile) se morajo za izvedbo vključiti programerji oziroma vzdrževalci ERP/PIS in za ta namen vključiti več ločenih klicev API-jev.

DopinusApiWrapper DAW.EXE

Še nekaj splošnih stvari:

Navodila za sestavo relativnih URL naslovov:

AppRoot Dopinus serverja se prebere iz konfiguracijske datoteke, zato se url-ji za proženje postopkov in morebiten ogled upload-anih datotek sestavljajo od tega mesta naprej, po naslednjem načinu:

-proženje postopka: gui/<provider>/<webname_postopka>

-klici predhodno uploadane datoteke: File/<file_id>

Če je datoteka takega tipa, da se ne more direktno prikazati/vizualizirati v internet browserju, bo browser forsiral download.

Za dodatno razumevanje avtentikacije:

»Metoda« authenticate se avtentificira na Dopinus serverju, zato lahko EXE nato izvaja vse serverske API-je.

Proženje postopkov, ki imajo UI, dvigne instanco enega od browserjev (trenutno forsiramo Chrome, ker postopki na IE še niso dovolj stestirani), saj uporabniški vmesnik Dopinus aplikacij teče izključno v internetnem pregledovalniku (browserju). To bi sicer zahtevalo ponovno avtentikacijo še v browserju, zato se v ozadju ob prvem klicu hiperlinka, ki zahteva UI v browserju, avtomatsko izvede prevzem te iste seje, ki jo je z avtentikacijo ustvaril EXE. Če je v tem browserju user že avtentificiran posebej/interaktivno, se prevzem seje ne zgodi in se postopki z UI prikazujejo v okviru njegove obstoječe seje.

Primer dela .config fajla za testno postavitev:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
<appSettings>
<!--BEGIN_Server url-->
<add key="ServerDomain" value="http://test.dopinus.com" />
<add key="Directory" value="" />
<!--END_Server url-->

<!--BEGIN_User credentials-->
<add key="Username" value="testuser" />
<add key="Password" value="testuser" />
<!--END_User credentials-->

<!--BEGIN_Daw basic settings-->
<add key="AuthCookiePath" value="" />           *If left empty default path is set (Users\XXXX\AppData\Roaming\Cookies)
<add key="AuthCookieName" value="" />         *If left empty default cookie name for E2 is set (dopinus.e2.session)
<add key="ResponsePath" value="c:\temp\DAWResponse.txt" />    ** glej spodaj
<add key="ResponseLogLevel" value="basic" />                *** glej spodaj
<add key="RequestTimeout" value="100000" />
<add key="AutoUpdate" value="true" />                    * veljavna vrednost je še »false«
<add key="ServerProxy" value="" />                       * možne vrednosti so še: null - brez proxy-ja,
auto - uporaba InternetExplorer default proxy-ja, ostalo -: uporabi podanega skozi config parameter
<add key="IgnoreAuthParams" value="" />                  * zagotavlja, da sta pri uporabi ukazov
version, update in help avtentikaciji potrebna.
<add key="IncludeResponseMarkup" value="" />              * določa, ali naj bo markup sintaksa v csv-ju
prisojna ali ne, torej samo podatki
<add key="ResponseEncoding" value="" />                  * določa kodno tabelo responsa pri ukazu
customrequest

<!--END_Daw basic settings-->

<!--BEGIN_Invoices specific settings-->
<add key="Browser" value="chrome" />                    *veljavne vrednosti so še »firefox«, »IE«
<add key="Ieracuni" value="web/isedico/izdanieracuni" />
<add key="parseFileUrl" value="public/IF_XmlPars_eRacun.xml" />
<add key="AttachmentListPath" value="DAW_Seznam_prilog.txt" />
<add key="sqlStatementEncoding" value="windows-1250" />
```

DopinusApiWrapper DAW.EXE

```
<add key="AutoConvertAllowedFormatsToPdf" value="false"/>  
vklopljena glavna nastavitve)  
<!--END_Invoices specific settings-->
```

**vklop avt. konverta za izdane eračune (pogoj, da je*

```
</appSettings>  
</configuration>
```

***v multiuser okoljih (terminalski strežniki ipd.) se mora določiti kot parameter pri avtentikaciji, ker mora vsak user imeti svoj response file, zato se praviloma usmeri na user direktorij*

**** veljavne vrednosti so »basic«, »advanced«, »intermediate«*

Aktualna .config datoteka je vedno priložena aktualni verziji DopinusApiWrapper.EXE-ja.

V njej se s strani ponudnika popravijo, ki integrira EXE, praviloma popravlja le en podatek – ResponsePath. Ostale podatke popravljajte le v predhodni komunikaciji z nami.

<http://test.dopinus.com> - je URL, kjer je pripravljeno testno okolje. Vsi ponudniki, ki nameravajo integrirati storitev morajo najprej vzpostaviti tehnično delovanje v tem okolju, šele nato jim odpremo dostop na produkcijsko okolje. Prijava na testno okolje je možna tudi brez certifikata, samo z uporabniškim imenom in geslom.

V testnem okolju se vse poslane fakture znajdejo v istem pogledu, saj je bistvena le tehnična pravilnost XML datoteke, klicev in elektronske poti. Dejanska razporeditev na produkciji se dogaja glede na pravice elektronskih potrdil (certifikatov).

<https://eracuni.dopinus.com> – URL, kjer je pripravljeno produkcijsko okolje, zanj je potrebno predhodno pridobiti veljaven dostop.

DopinusApiWrapper DAW.EXE

Dodatno, za bolj »sophisticirane« povezave, ki zapisujejo podatke še v kakšne druge tabele ali če morda ponudnik informacijskega sistema ne želi uporabljati našega že pripravljenega uporabniškega vmesnika v internetnem brskalniku, ampak želi morebiti napraviti vse postopke povsem integrirane v svojo programsko opremo, so na voljo še naslednje možnosti:

file upload : **DAW file filename ~~open~~ [stack[key,[source]]]**

response je ID (GUID), ki ga je file dobil v Dopinus Key-Value Store-u. Ta ID je potem del URL-ja preko katerega dosežemo datoteko.

- *filename* je file z lokalnega diska, ki ga želimo uploadati v Dopinus Key-Value store.
- *open* parameter je z ver. 33 odstranjen!!!
~~open je dodatni parameter, ki pove, da se naj file takoj po uploadu še odpre v browserju (sicer se ga lahko kasneje odpre z drugim klicem exe-ja in parametrom open (točka 5.), pri čemer pa mora programer takrat sam ustrezno pripraviti URL, za kar si mora predhodno zapomniti ID iz response-a)~~
- *stack* je nov opsijski parameter, ki pove, da se poleg uploada izvede še kreiranje zapisa v tabelo OTH_UploadDownloadExternalFiles (je na dopinus providerju). Vsebuje dva parametra (**obvezna navedba v oglatih oklepajih**):
 - *key*, ki je obvezen. Vsebina se zapiše v polje Key. Po tem polju se vedno izvaja filtriranje pri dfiles metodi in je tam ta podatek obvezen. Služi za razpoznavo zapisov.
 - *source*, ki ni obvezen. Vsebina se zapiše v polje Source. Po tem polju se lahko dodatno izvaja filtriranje pri dfiles metodi. Služi za dodatno razpoznavo zapisov še po enem kriteriju.

Primeri klicev: DAW file c:\filename.txt stack[ISEDICO,KPS]
DAW file c:\filename.txt "stack[ISEDICO,KPS]"
DAW file c:\filename.txt stack[ISEDICO]

file download : **DAW dfiles stack[key,[source]] dldir [overwrite]**

Ukaz »downloada« datoteke, ki so bile predhodno naložene v Dopinus Key-Value Store preko metode file in uporabe parametra stack.

- *stack* je obvezen parameter, ki pove, da naj se še neprenehane datoteke prenesejo iz Dopinus Key-Value Store na lokalni disk (preko OTH_UploadDownloadExternalFiles tabele). Vsebuje dva parametra (**obvezna navedba v oglatih oklepajih**):
 - *key*, ki je obvezen. Po tem polju se izvaja filtriranje. Služi za razpoznavo zapisov.
 - *source*, ki ni obvezen. Po tem polju se lahko dodatno izvaja filtriranje. Služi za dodatno razpoznavo zapisov še po tem kriteriju. V kolikor ni podan, se filter po tem polju ne izvaja, zajamejo se vsi zapisi ne glede na vrednost v Source.
- *dldir* je drugi parameter za ime direktorija, kamor se bodo prepisovale datoteke. Če ni naveden, se datoteke shranijo na trenutni direktorij (Primer: c:\DAWDownload)
- *overwrite* je opsijski parameter, ki pove ali naj se datoteka, ki se že nahaja na dldir, prepíše ali ne. Če je podan, se na konec imena datoteke doda še _YYYYMMDDThhmmss

Ko se datoteka enkrat uspešno prenese, se v sistemu označi kot prenesena in se naslednjič več ne prenaša.

Primer oblike zapisa imena datoteke:

OTH_27a07497-0d13-465b-a360-ef69588af3a1_Tečaji.xls. pri čemer pomeni:

- Predpona »OTH«
- id je enolična identifikacijska številka tipa GUID
- ime datoteke
- opsijsko še YYYYMMDDThhmmss - leto, mesec, dan, oznaka »T«, ura, minuta in sekunda zapisa na direktorij

DopinusApiWrapper DAW.EXE

Primeri klicev: DAW dfiles stack[ISEDICO,KPS] c:\temp
DAW dfiles "stack[IS EDICO,KPS]" c:\temp
DAW dfiles stack[ISEDICO]
DAW dfiles stack[ISEDICO] c:\temp overwrite

data create : **DAW datacreate entity kvtxtfile [postopek]**

v response datoteko vrne ID, ki ga je kreiran zapis dobil v tabeli, v katero smo pisali. Ta ID je potem pomemben za morebitno filtriranje (ali kot parameter pri metodi open).

- *entity* je ime tabele iz dopinusovega seznama tabel (ConTables) v formatu <provider/table_name>
- *kvtxtfile* je ime txt fajla, v katerem so fieldName : Value pari, ki jih je potrebno vpisati v navedeno tabelo

(trenutno šumnike pravilno interpretira le po UTF-8 in 1250 encoding-u, po potrebi lahko dodamo še kakšnega – recimo 852 ali celo 437)

- *postopek* je opcijski url za proženje poljubnega z Dopinusom razvitega postopka, ki se bo prožil takoj po uspešnem zapisu v navedeno tabelo. Za delo z eRačuni bodo na naši strani razviti vsi potrebni postopki za izdane in prejete fakture, za testne primere se uporabljata postopka »prejetieracuni« in »izdanieracuni«, ki se kliceta preko ustreznega URL (glej navodila za sestavo URL spodaj).

get data : **DAW getdata entity [Full]**

entity je ime tabele iz dopinusovega seznama tabel v formatu <provider/table_name>. Response je tak kot ga vrne metoda Data.GetEntityData. Po defaultu dobimo Flat paket, opcijsko se lahko zahteva Full paket, ki vključuje še polja iz konektanih tabel. Drugi parametri (filtri ipd. v tej enostavni varianti niso podprti, za vse bolj kompleksne primere je potrebno uporabiti customrequest).

dodatno pri ieracun:

Za kakšne bolj splošne potrebe so za »ieracun« pripravljene še naslednji opcijski parametri:

[E#entity] [F#field] [P#parserfile]

- *E#entity* je ime tabele iz dopinusovega seznama tabel napisana v notaciji provider/ime_tabele
- *F#field* je ime fielda iz gornje entitete, v katerega naj se zapiše URL za dostop do upload-ane datoteke
- *P#parserfile* je opcijski url (glej navodila spodaj) do predhodno uploadane txt datoteke z opisom pravil za razparsanje uploadane XML datoteke in prepis podatkov v novoustvarjen zapis v navedeni entiteti.

Datoteka se predhodno pripravi in uploada. Uporablja xpath sintakso za opis kateri podatek iz XML se zapiše v katero polje v bazo.

Ker imamo pripravljeno to tehnologijo odnosa XML faktura vs. podatkovna baza, ki deluje dvosmerno (podatki iz baze v XML ali parsanje XML in zapis podatkov v bazo) **lahko ponudnikom, ki še nimajo pripravljene XML fakture v eSlog 1.5/1.6 , to pomagamo pripraviti na zelo enostaven in hiter način.**

vse ostalo : **DAW customrequest JSONfile [true] [trans] [simple]**

Na ta način se da prožiti katerikoli API podprt s strani Dopinus serverja. V JSONfile mora biti v enostavni tekstovni obliki napisan zahtevek v pravilni strukturi/sintaksi. Kdor bo potreboval močnejšo integracijo od tu opisanega ali bo želel razviti lastne specifične postopke in jih integrirati v svojo aplikacijo, naj stopi v kontakt z nami in opiše svoje potrebe. V tehnologiji Dopinus se takšnim potrebam lahko ustreže zelo hitro, pripravi jih lahko tudi sam ponudnik PIS, ki je integriral servis ali celo uporabniki sami.

- ~~*true*~~ pomeni, da se zahtevek kliče kot transakcija (default je false)
- *namesto true se uporabi trans* - podpora transakcije pri pošiljanju requesta
- *simple*

Pripravi csv-like response (v configu nov podatek "IncludeResponseMarkup", ki določa, ali naj bo markup sintaksa v csv-ju prisotna ali ne, torej samo podatki).