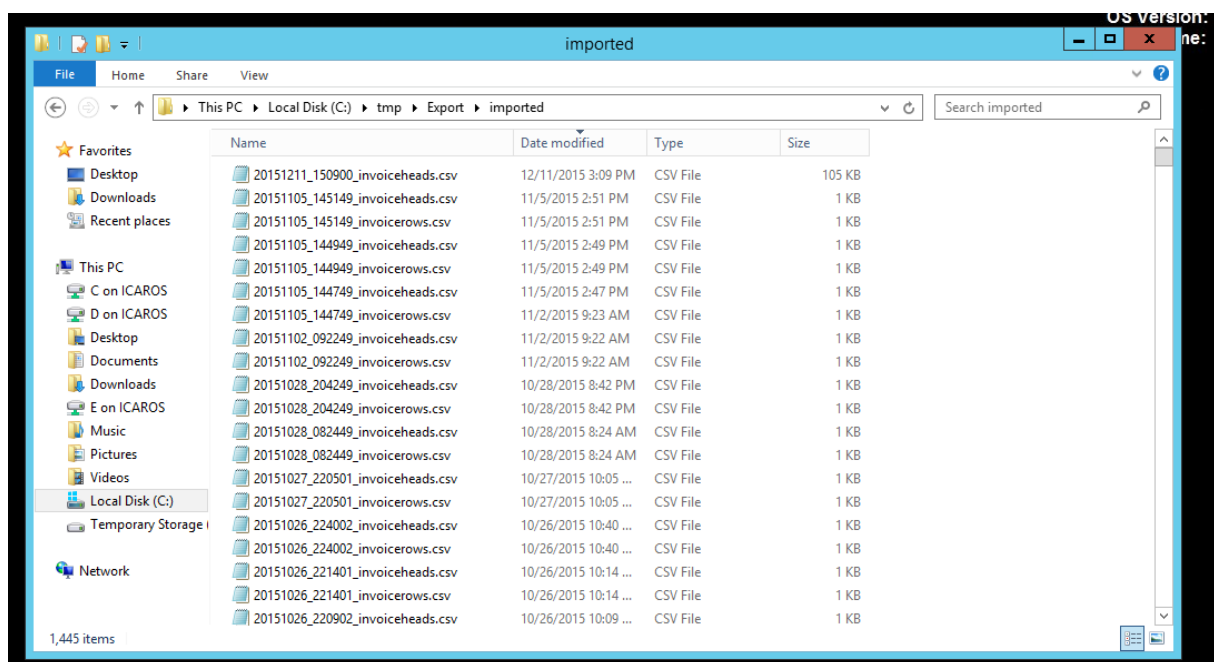


Manuell import till Lime Pro

Ibland kan det vara nödvändigt att göra en manuell import till Lime Pro, t.e.x. om en fil inte kan importeras och man behöver mer information för att felsöka. I den vanliga tjänstens loggar så visas endast om importen lyckades eller misslyckades. Exakt vilket felmeddelande som servern lämnade ifrån sig ser man först vid en manuell import.

Importen till Lime Pro använder sig av två filer. Dels en .csv-fil som innehåller datan som ska importeras och dels en .py-scriptfil som innehåller scriptet som laddar upp filen till servern, konfigurerar importjobbet, och startar importen. I vårt exempel så kommer vi göra en manuell import av .csv-filen som innehåller orderhuvud men principen är densamma även om vi vill importera orderrader eller kunder.

Börja med att leta upp den .csv-fil som du vill importera. Du hittar den under ”\export\imported”.



I vårt exempel så vill vi importera filen 20151211_150900_invoiceheads.csv. Kopiera filen till en annan mapp och kom ihåg var du placerade den.

Nästa steg är att sätta ihop en .py-fil för importen. .py-filerna som tjänsten använder sig av innehåller platshållare för limeserver, limedatabas, användarnamn, och lösenord eftersom de värdena styrs av programmets konfigurationsfil. Navigera till ”VismalIntegrationService\Scripts” och kopiera ImportInvoiceHeads.py till samma mapp som du placerade .csv-filen. Öppna upp .py-filen i t.ex. Anteckningar.

```
print ('argument 0:', sys.argv[1])
client = limeclient.LimeClient('[LimeApiUri]', '[LimeApiDb]', False, False)
with client.login('[LimeApiUser]', '[LimeApiPassword]') as c:
    with open(sys.argv[1], encoding='utf-8') as content:
        f = limeclient.ImportFiles(c).create(filename='invoiceheads.csv',content=content)
        f.delimiter = ';'
        f.save()
    invoiceHeads = limeclient.LimeTypes(c).get_by_name('invoice')
    config = limeclient.ImportConfigs(c).create(lime_type=invoiceHeads, importfile=f)
    config.behavior = limeclient.ImportConfig.CreateAndUpdate
```

Där ser du platshållarna för [LimeApiUrl], [LimeApiDb], [LimeApiUser] och [LimeApiPassword]. Dessa måste bytas ut mot de värden som är giltiga för miljön du arbetar med. Enklaste sättet att hitta värdena är att navigera till mappen "VismalIntegrationService" och öppna filen "VismalIntegrationService.exe.config" i Anteckningar.

```
<add key="PythonExecutable" value="c:\tmp\py\venv\>ci
<add key="InvoiceExportNextExecution" value="3/11/20:
<add key="InvoiceExportIntervalSeconds" value="10" /:
<add key="LimeApiUri" value="https://provisma:443" /:
<add key="LimeApiDb" value="visma_test_2" />
<add key="LimeApiUser" value="vismauser" />
<add key="LimeApiPassword" value="potatis" />
</appSettings>
:/configuration>
```

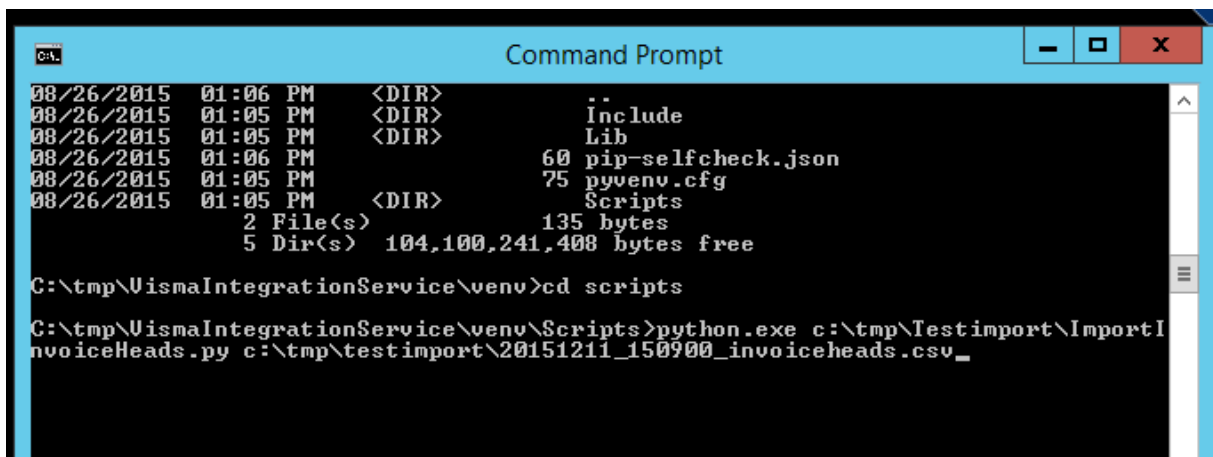
Ta motsvarande värde från .config-filen och ersätt platshållarna i .py-filen. När du är färdig så borde din .py-fil se ut ungefär såhär.

```
print ('argument 0:', sys.argv[1])
client = limeclient.LimeClient('https://provisma:443', 'visma_test_2', False, False)
with client.login('vismauser', 'potatis') as c:
    with open(sys.argv[1], encoding='utf-8') as content:
        f = limeclient.ImportFiles(c).create(filename='invoiceheads.csv', content=content)
        f.delimiter = ';'
        f.save()
```

Notera att tecknena "[" och "]" också är en del av platshållarvärdet och ska tas bort.

Öppna en kommandoprompt och navigera till mappen "VismalIntegrationService\venv\Scripts" och skriv enl följande mall:

Python.exe [sökväg till din .py-fil] [sökväg till din .csv-fil]



The screenshot shows a Windows Command Prompt window titled "Command Prompt". The window displays the following text:

```
08/26/2015 01:06 PM <DIR>      .
08/26/2015 01:05 PM <DIR>      Include
08/26/2015 01:05 PM <DIR>      Lib
08/26/2015 01:06 PM          60 pip-selfcheck.json
08/26/2015 01:05 PM          75 pyvenv.cfg
08/26/2015 01:05 PM <DIR>      Scripts
                2 File(s)      135 bytes
                5 Dir(s)  104,100,241,408 bytes free

C:\tmp\VismaIntegrationService\venv>cd scripts

C:\tmp\VismaIntegrationService\venv\Scripts>python.exe c:\tmp\Testimport\ImportI
nvoiceHeads.py c:\tmp\testimport\20151211_150900_invoiceheads.csv_
```




Tryck Enter för att starta importen. Du kommer att se en hel del varningar som har med HTTPS att göra. De är helt normala och inget du behöver oroa dig för. Är det något annat fel, t.ex. felaktiga användaruppgifter, felaktiga rättigheter på servern, korrupt .csv-fil, osv, så kommer du att få ett felmeddelande från Lime Pro-servern och importen avbryts. Det felmeddelandet kan du sen använda för att fortsätta din felsökning på Lime Pro server-sidan.

Uppstår inget fel så kommer importen att köra och programmet avslutas automatiskt när importen är helt färdig.

Spara kund till Visma

Om importen till Lime Pro fungerar men det inte går att spara eller uppdatera en kund via integrationsknappen i Lime Pro-klienten så tyder det på att anropen till integrationstjänsten inte går fram korrekt.

Integrationstjänsten genererar två typer av loggar. Dels den vanliga programloggen – [datum]_logg.txt – och dels en trafiklogg – [datum]_HttpLogg.txt.

 20151202_Logg.txt	12/2/2015 11:59 PM	Text Document	730 KB
 20151222_HttpLogg.txt	12/2/2015 4:27 PM	Text Document	3 KB
 20151201_...	12/1/2015 11:50 PM	Text Document	735 KB

Alla ingående anrop från Lime Pro-klienten loggas i HttpLogg.txt-filen. Varje gång användaren klickar på integrationsknappen i Lime Pro så går det iväg minst ett anrop till integrationstjänsten och du borde se att HttpLogg.txt-filen för dagens datum växer. HttpLogg.txt-filen innehåller information om tidpunkt för anropet, vilken typ av anrop det var (POST = skapa, PUT = uppdatera), vilken resurs som anropet skedde mot (oftast /api/v1/customer) och sist på raden kan du se vilken HTTP statuskod som anropet resulterade i. 200 = OK, 401 = rättighetsproblem, 500 = internt fel i applikationen. Vid statuskod 500 så finns det alltid mer information i den vanliga Logg.txt-filen om vad som gått fel.

Timestamp	Method	Url	UserAgent	Ip
12/2/2015 2:04:24 PM	POST	/api/v1/customer		
12/2/2015 2:07:18 PM	POST	/api/v1/customer		
12/2/2015 2:08:56 PM	PUT	/api/v1/customer		
12/2/2015 2:15:20 PM	POST	/api/v1/customer		
12/2/2015 2:17:42 PM	PUT	/api/v1/customer		

Slutet av raden:

727; .NET CLR 3.0.30729)"	fe80::dc37:e82c:50cd:c0c9%16	200
727; .NET CLR 3.0.30729)"	fe80::dc37:e82c:50cd:c0c9%16	200
727; .NET CLR 3.0.30729)"	fe80::dc37:e82c:50cd:c0c9%16	200
727; .NET CLR 3.0.30729)"	fe80::dc37:e82c:50cd:c0c9%16	200
727; .NET CLR 3.0.30729)"	fe80::dc37:e82c:50cd:c0c9%16	200
727; .NET CLR 3.0.30729)"	fe80::dc37:e82c:50cd:c0c9%16	200

Om användaren trycker på integrationsknappen och får ett felmeddelande men det inte kommer någon rad i HttpLogg.txt-filen så tyder det på att Lime Pro-klienten inte lyckas skicka anropet till integrationstjänsten. Oftast beror detta på felaktiga inställningar i Integrationstjänsten.

Navigera till mappen "VismalIntegrationService" och öppna filen VismalIntegrationService.exe.Config i Anteckningar. Hitta raden där Key-värdet är "ApiUrl". Värdet på den raden bestämmer vilken ipadress/maskin som integrationstjänsten ska lyssna efter inkommande anrop på. Om den är satt till <http://localhost:8194> så kommer den bara kunna ta emot lokala anrop, d.v.s. anrop från andra maskiner än den egna ignoreras. Korrekt konfiguration ska vara [http://\[maskinens_namn\]:8194](http://[maskinens_namn]:8194). T.ex. <http://provisma.mindomän.local:8194> eller <http://AppServer01:8194>.

Stoppa integrationstjänsten (VismaIntegrationService), ändra i .config-filen, och starta sedan integrationstjänsten igen för att ändra vilken adress som tjänsten api finns tillgängligt på. Får användaren fortfarande ett felmeddelande så måste du kontrollera i Lime Pro-klientens filer/integrationsknappens filer att du där angivit rätt sökväg till api'et.

Angående vilken port som används; i standardinställningarna så är det port 8194 som används för kommunikation mellan Lime Pro-klienten och integrationstjänsten. Vilken port som används är valfri och byter du port i inställningarna för integrationstjänsten så måste du också komma ihåg att ändra inställningarna i Lime Pro/Integrationsknappens filer.

Att en port används för API-kommunikation mellan Lime Pro-klienten och integrationstjänsten innebär också att maskinen som Lime Pro-klienten kör på måste kunna kommunicera över den porten med servern där integrationstjänsten kör. Eftersom Lime Pro-klienten och den server som integrationstjänsten är installerad på i nästan 100% av alla situationer ligger på samma lokala nät bakom en brandvägg så utgör den här kommunikationen ingen säkerhetsrisk och är tekniskt att likställa med att man kör Visma Administration-klienten ute på lokala maskiner men har Visma Administration-databasen utdelad på en server.

Vissa servrar har en mjukvarubrandvägg installerad lokalt och eftersom Integrationstjänsten kör på en icke-standard port så måste ibland serverns lokala brandvägg – den server som integrationstjänsten är installerad på – öppnas för att tillåta trafik till integrationstjänsten. Förfarandet för hur man öppnar upp en port i en brandvägg skiljer sig mellan olika brandväggar, men poängen är att klientmaskinerna som kör Lime Pro måste kunna kommunicera över den konfigurerade porten med den server som integrationstjänsten finns installerad på.