Servant API

Technická dokumentace

Obsah

Úvod 4
Terminologie a základní fakta o datové výměně a zobrazených informacích 5
Testování datového napojení 6
Operace 7
Ověření spojení a přihlašovacích údajů 7
Načtení stavu objednávek na příjem 7
Načtení stavu objednávek na výdej 7
Načtení inventurního stavu 7
Načtení aktuálního seznamu dopravců 7
Odeslání seznamu partnerů 7
Odeslání sortimentu 8
Odeslání objednávky na příjem 8
Odeslání objednávky na výdej 8
Funkce 9
CanAccess 9
GetAllReceiptOrders 9
GetReceiptOrdersByDate 9
GetReceiptOrder 10
GetReceiptOrders 10
GetReceiptOrderByDocument 11
GetReceiptOrdersByDocument 11
GetAllIssueOrders 11
GetIssueOrdersByDate 12
GetIssueOrder 12
GetIssueOrders 13
GetIssueOrderByDocument 13
GetIssueOrdersByDocument 13
GetInventory 14

GetCarriers 14
GetImportResults 14
ImportPartners 15
ImportProducts 15
ImportReceipt 16
ImportIssue 16
ImportIssues 17
ImportFiles 17
Struktury 18
OrderFilterType 18
InventoryFilterType 18
ResultType 18
CredentialsType 19
DateTimeIntervalType 19
AttributesType 19
ActivityType 20
ItemType 20
OrderItemType 20
TransportItemType 21
CashOnDeliveryType 21
DimensionType 21
EanType 22
ReceiptOrderType 22
IssueOrderType 23
InventoryItemType 24
CarrierItemType 24
ImportIssueResultType 25
ImportRequestResultType 25
ImportResultType 26
InfoTextType 27
ImportBankType 27
ImportPartnerDefinitionType 27
ImportItemIdType 27
ImportProductItemDetailsType 28
ImportProductDimensionType 28
ImportProductEanType 28
ImportItemIdDetailsType 29
ImportLabelType 29

ImportActivityType 29
ImportProductActivityType 30
ImportProductType 30
ImportPartnerPointType 31
ImportPartnerType 31
ImportDetailsType 32
ImportDeliveryType 32
ImportItemType 33
ImportReceiptType 33
ImportIssueType 33
ImportIssueWithFilesType 33
Doručení do výdejního místa 34
ImportFileType 34
Použití 35
C# 35
PHP 36
SAPI Aplikace 36
Export z WSS 37
Import na WSS 41
Řízení přes příkazovou řádku 42
Externí nástroje a aplikace 44
Ověření XML dotazu proti WSDL předpisu 44

Úvod

Dokument obsahuje popis SW rozhraní webové služby umožňující provádět operace definované dále. Je určen pro vývojáře SW informačního systému na straně zákazníka společnosti Servant a.s. Rozhraní je postavené na protokolu SOAP a popisu rozhraní webové služby WSDL. Toto rozhraní je možné jednoduše použít v rámci datového napojení psaného v C#, PHP a dalších. Data lze přes rozhraní získat také pomocí SAPI Aplikace, která následně umožňuje data exportovat v několika nejběžnějších formátech.

URL: https://www.webskladservant.cz/new/impl/SAPI/V6/?wsdl
Verze SW rozhraní webové služby: V6.9
Verze dokumentu: V6.2026-07-10

Datová komunikace je šifrovaná pomocí TLS 1.2. **Tato verze šifrování je na straně klienta vyžadována** (připojení zabezpečené nižší verzí šifrování selže). Všichni uživatelé WSS dostupného přes webový server na adrese https://www.webskladservant.cz/ získávají také automaticky přístup přes SW rozhraní webové služby. Pro přihlášení se používá stejných přihlašovacích údajů, jakých se používá pro přihlášení do WSS.

SW rozhraní webové služby nabízí následující operace:

- Ověření spojení a přihlašovacích údajů
- Načtení stavu objednávek na příjem
- Načtení stavu objednávek na výdej
- Načtení inventurního stavu
- Načtení aktuálního seznamu dopravců
- Odeslání seznamu partnerů
- Odeslání sortimentu
- Odeslání objednávky na příjem
- Odeslání objednávky na výdej

POZOR: Maximální denní počet API dotazů může být limitován. V případě blížícího se limitu dochází při API dotazech, společně s daty, k návratu varování. Při překročení nastaveného denního limitu dojde do dalšího dne k zastavení možnosti API komunikace. Oba tyto stavy jsou indikované ve struktuře ResultType i na hlavní stránce WSS po přihlášení. V případě, že chcete předcházet této situaci, obzvláště při vyšší počtu dotazovaných objednávek, je potřeba využívat agregační funkce, tj. dotazovat se jedním dotazem např. na více objednávek, atd. Pro více informací kontaktuje operátora Servant.

Terminologie a základní fakta o datové výměně a zobrazených informacích

V rámci dokumentu i další komunikace se můžete setkat s následujícími termíny:

WSS	**W**eb**S**klad**S**ervant (https://www.webskladservant.cz) = webové stránky umožňující komunikaci (datovou výměnu) se střediskami společnosti Servant
Servant API	API (programové rozhraní) pro přístup do WSS
SAPI aplikace	Windows aplikace využívající Servant API

WSS je určen pro uživatele, kteří chtějí mít nepřetržitý přístup k objednávkám, inventuře zboží, a dalším přes webové stránky. Servant API poskytuje totožné možnosti jako WSS s tím rozdílem, že je primárně určený pro programové napojení na informační systém zákazníka. SAPI aplikace je nadstavba nad vlastní API (aplikace interně Servant API přímo používá) a může sloužit jak k testovacím účelům, tak i k uživatelskému přístupu z PC. Aplikaci lze rovněž ovládat z příkazové řádky bez zobrazení jakéhokoli okna a tím docílit napojení na další procesy na serveru zákazníka.

POZOR: Odeslání seznamu partnerů lze v současné době pouze přes Servant API. WSS takovou možnost nenabízí. Ostatní funkce importu jsou přístupné jak přes Servant API, tak i přes formuláře WSS. Pokud na WSS tuto volbu nevidíte a rádi byste formuláře využívali, kontaktujte operátora Servant.

Synchronizace dat na WSS / Servant API probíhá z několika zdrojů a k synchronizaci dochází průběžně tak, jak se vyřizují požadavky na import a zpracovávají se vlastní objednávky. Je tedy možné, že v určitý okamžik budou informace zobrazené na WSS / poskytované přes Servant API nekompletní či neaktuální. Typicky se jedná o následující situace:

Q Číslo přepravy již bylo přiděleno, nicméně stále nevidím informace o dopravě. Proč?
- **A** Tato situace je indikována rovněž varováním „Doprava pro tuto objednávku nebyla nalezena, informace o dopravě nebudou zobrazeny.“ a může být způsobena buď nenačtením dat o dopravě, případně chybou. Prosím počkejte minimálně 1 hodinu. Pokud chyba přetrvá, kontaktujte operátora Servant.

Dále je dobré vědět:

Q Servant API mi vrací chybu „Chyba přístupu k serveru“ s popisem „Chyba při komunikaci se serverem Servant, zkuste to, prosím, později“. Proč?
- **A** Chyba může být způsobena nedostupností serveru Servant. Pokud chyba přetrvá, kontaktujte operátora Servant.

Q Servant API mi vrací chybu „Chyba zadání importu (Servant server)“ s popisem „Failed - Neexistující ukladatel“. Proč?
- **A** S největší pravděpodobností ještě nemáte plně aktivovaný účet (na cílovém středisku Servant není tento účet známý). Kontaktujte operátora Servant.

Q Co mám vyplnit do atributu „warehouse“?
- **A** Jedná se o zkratku skladu. Zkratku skladu dostanete přidělenou podle toho, ve kterém skladě skladujete.

Testování datového napojení

Pro testování neexistuje žádné speciální testovací prostředí. Poté, co od nás obdržíte přístup do WSS (a nastavíte si heslo), je zároveň automaticky aktivované i Servant API. Ne všechny funkce, které jsou poskytované Servant API, jsou využitelné pro vaši společnost. Požadavky na funkční obsah přenášených dat a vhodnost jednotlivých funkcí pro vaši společnost prosím konzultujte s naším IT technikem. Ten vám také případně sdělí konstanty, které je nutné do komunikace zahrnout.

Doporučujeme začít s:

1. Sjednání přístupu do WSS a seznámení se s touto dokumentací
2. Vytvoření základů datové komunikace a Ověření spojení a přihlašovacích údajů
3. Odeslání sortimentu a sledování stavu importu sortimentu voláním GetImportResults
4. Odeslání seznamu partnerů a sledování stavu importu
5. Odeslání objednávky na příjem a sledování stavu importu
6. Načtení stavu objednávek na příjem
7. Načtení inventurního stavu
8. Odeslání objednávky na výdej a sledování stavu importu
9. Načtení stavu objednávek na výdej

V případě, že selže vlastní import nebo dojde k následné chybě při zpracování dat, dostanete informaci o chybě. Pokud selže vlastní import (např. neplatné přihlašovací údaje, chybný formát XML dat proti WSDL definici, apod.), chyba je reportovaná synchronně jako výsledek SOAP dotazu. Pokud selže následné zpracování dat (např. duplicitní objednávka, snaha vyskladnit nenaskladněný nebo neexistující sortiment, apod.), chyba je reportovaná asynchronně dotazem na stav importu (GetImportResults).

V případě nemožnosti odeslat nějaký XML dotaz si prosím vždy ověřte, že jste validní proti WSDL předpisu (viz kapitola Ověření XML dotazu proti WSDL předpisu). Kontrola validity probíhá i na straně serveru Servant a to ještě před vlastní zpracováním dotazu. V případě nevalidního dotazu je vrácena chyba s popisem umožňující identifikaci zdroje problému.

Po bezchybném přenosu testovací databáze sortimentu (Odeslání sortimentu), seznamu partnerů (Odeslání seznamu partnerů) a objednávky na příjem (Odeslání objednávky na příjem), prosím kontaktujte našeho IT technika, abychom v našem skladovém systému zpracovali testovací objednávku (potvrdíme testovací příjem a tím zboží fiktivně naskladníme). V tuto chvíli již máte na WSS informace o stavu této objednávky i o skladové zásobě (Načtení inventurního stavu). Pokračovat je možné vytvořením objednávky na výdej (Odeslání objednávky na výdej). Po jejím bezchybném přenosu nás opět informujte, abychom objednávku zpracovali. Stejně jako v případě objednávky na příjem, je po zpracování i této objednávky na WSS možné získat její stav. Stav se aktualizuje tak, jak je objednávka realizovaná a tedy je možné následnými dotazy například sledovat stav nakládky, doručení, atd. Aktualizovaný stav skladových zásob (Načtení inventurního stavu) si můžete stahovat vždy každý večer kolem 18 té hodiny.

Pokud budete potřebovat, fyzicky odešleme zkušební zásilky.

Z praxe máme vyzkoušeno, že pokud se držíte manuálu a respektujte normu a prostředí, na které se manuál odvolává, pak technickou část nastavení datové výměny lze provést bez naší spolupráce.

Před ostrým provozem všechna zkušební data smažeme a nebudou tak zatěžovat ostrý provoz, na který se všichni těšíme.

Operace

Ověření spojení a přihlašovacích údajů

Ověřit spojení a platnost přihlašovacích údajů lze přes funkci CanAccess. Volání této funkce před voláním dalších funkcí pro načtení objednávek nebo inventurního stavu není vyžadováno. Volání lze využít například pro jednorázové ověření přístupu na SW rozhraní webové služby nebo pro ověření dostupnosti serveru.

Načtení stavu objednávek na příjem

Objednávky na příjem lze načíst voláním jedné z následujících funkcí:

- GetAllReceiptOrders – vrátí všechny objednávky
- GetReceiptOrdersByDate – vrátí všechny objednávky, které odpovídají intervalu
- GetReceiptOrder – vrátí **jednu** konkrétní objednávku identifikovanou číslem objednávky
- GetReceiptOrders – vrátí **více** konkrétních objednávek identifikovaných čísly objednávek
- GetReceiptOrderByDocument – vrátí **jednu** konkrétní objednávku identifikovanou dokladem
- GetReceiptOrdersByDocument – vrátí **více** konkrétních objednávek identifikovaných doklady

Načtení stavu objednávek na výdej

Objednávky na výdej lze načíst voláním jedné z následujících funkcí:

- GetAllIssueOrders – vrátí všechny objednávky
- GetIssueOrdersByDate – vrátí všechny objednávky, které odpovídají intervalu
- GetIssueOrder – vrátí **jednu** konkrétní objednávku identifikovanou číslem objednávky
- GetIssueOrders – vrátí **více** konkrétních objednávek identifikovaných čísly objednávek
- GetIssueOrderByDocument – vrátí **jednu** konkrétní objednávku identifikovanou dokladem
- GetIssueOrdersByDocument – vrátí **více** konkrétních objednávek identifikovaných doklady

Načtení inventurního stavu

Položky inventurního stavu lze získat voláním funkce GetInventory.

Načtení aktuálního seznamu dopravců

Aktuální seznam dopravců platný pro daného ukladatele lze získat voláním funkce GetCarriers.

Odeslání seznamu partnerů

Voláním funkce ImportPartners se importují nové informace o partnerech. Partnerem je myšlen dopravce, ukladatel nebo odběratel. Pokud funkce skončí úspěchem, je předáno ID transakce, pomocí kterého je možné následně stav importu sledovat (vlastní import a navázané operace trvají netriviálně dlouhou dobu). V opačném případě je vrácen ihned důvod selhání.

Aktuální stav importu je možné získat voláním funkce GetImportResults.

POZOR: Přestože je indikované dokončení zpracování importu, může na pozadí k reálnému dokončení dojít až později bez ohledu na indikovaný stav zpracování. Zpracování a pronesení informací může reálně trvat až **30 minut**.

Odeslání sortimentu

Voláním funkce ImportProducts se importují nové informace o sortimentu. Funkce umožňuje nastavit jen některé atributy za účelem importu jen některých informací. Je tedy možné provést jen import EAN nebo prováděných VAS nad již definovaným zbožím. Pokud funkce skončí úspěchem, je předáno ID transakce, pomocí kterého je možné následně stav importu sledovat (vlastní import a navázané operace trvají netriviálně dlouhou dobu). V opačném případě je vrácen ihned důvod selhání.

Aktuální stav importu je možné získat voláním funkce GetImportResults. Zpracování a pronesení informací může trvat až **30 minut**.

Odeslání objednávky na příjem

Voláním funkce ImportReceipt se provede import objednávky na příjem. Pokud funkce skončí úspěchem, je předáno ID transakce, pomocí kterého je možné následně stav importu sledovat (vlastní import a navázané operace trvají netriviálně dlouhou dobu). V opačném případě je vrácen ihned důvod selhání.

Aktuální stav importu je možné získat voláním funkce GetImportResults. Zpracování a pronesení informací může trvat až **30 minut**.

Odeslání objednávky na výdej

Voláním funkce ImportIssue se provede import objednávky na výdej. K importu je možné připojit i dokument, který se automaticky vytiskne po zabalení objednávky (např. vaše faktura, návod na použití, dodací list, atp.), jehož data jsou kódována standardem Base64. Je možno vytisknout jen jeden dokument a jen jednou. Pokud tedy potřebujete přiložit např. 3 kopie jedné faktury, je nutné, aby PDF soubor obsahoval 3 stránky se stejným obsahem. Stejným způsobem je možné připojit i vlastní přepravní štítek. Soubor(y) je možné připojit i později voláním funkce ImportFiles. Pokud funkce skončí úspěchem, je předáno ID transakce, pomocí kterého je možné následně stav importu sledovat (vlastní import a navázané operace trvají netriviálně dlouhou dobu). V opačném případě je vrácen ihned důvod selhání.

V případě potřeby importu více objednávek na výdej najednou (v jednom dotazu), je možné využít funkci ImportIssues, která umožňuje dávkové odeslání až 100 objednávek.

Aktuální stav importu je možné získat voláním funkce GetImportResults. Zpracování a pronesení informací může trvat až **30 minut**.

Funkce

CanAccess

Funkce pro ověření dostupnosti serveru a přihlašovacích údajů. Funkce přijímá jako vstup strukturu **CanAccessParams** a jako výstup poskytuje strukturu **CanAccessParamsResult**.

Syntaxe:

```
CanAccessParamsResult CanAccess(CanAccessParams)
```

Vstup (CanAccessParams)	
Jméno	**Typ**
credentials	CredentialsType

Výstup (CanAccessParamsResult)	
Jméno	**Typ**
result	ResultType

GetAllReceiptOrders

Funkce pro získání všech objednávek na PŘÍJEM, které jsou k dispozici pod zadanými přihlašovacími údaji. Funkce přijímá jako vstup strukturu **GetAllReceiptOrdersParams** a jako výstup poskytuje strukturu **GetAllReceiptOrdersResult**. **POZOR:** Funkce může v rámci jednoho dotazu vrátit maximálně 1000 objednávek. V případě, že by dotaz měl vrátit objednávek více, bude vráceno prvních 1000 (seřazeno sestupně dle požadované expedice). V případě, že si přejete načíst konkrétní objednávky, je možné použít volání funkcí GetReceiptOrdersByDate, GetReceiptOrder nebo GetReceiptOrderByDocument.

Syntaxe:

```
GetAllReceiptOrdersResult GetAllReceiptOrders(GetAllReceiptOrdersParams)
```

Vstup (GetAllReceiptOrdersParams)	
Jméno	**Typ**
credentials	CredentialsType
filter	OrderFilterType

Výstup (GetAllReceiptOrdersResult)	
Jméno	**Typ**
result	ResultType
orders	ReceiptOrderType[]

GetReceiptOrdersByDate

Funkce pro získání všech objednávek na PŘÍJEM, které jsou k dispozici pod zadanými přihlašovacími údaji a zároveň vyhovují nastavenému intervalu data. Interval je aplikován na položku „načteno do systému“. Funkce přijímá jako vstup strukturu **GetReceiptOrdersByDateParams** a jako výstup poskytuje strukturu **GetReceiptOrdersByDateResult**. **POZOR:** Funkce může v rámci jednoho dotazu vrátit maximálně 1000 objednávek. V případě, že by dotaz měl vrátit objednávek více, bude vráceno prvních 1000 (seřazeno sestupně dle požadované expedice). Pokud k tomu dojde, je možné zúžit nastavený interval data, nebo v případě, že si přejete načíst konkrétní objednávku, je možné použít volání funkcí GetReceiptOrder nebo GetReceiptOrderByDocument.

Syntaxe:
```
GetReceiptOrdersByDateResult GetReceiptOrdersByDate(GetReceiptOrdersByDateParams)
```

Vstup (GetReceiptOrdersByDateParams)	
Jméno	**Typ**
credentials	CredentialsType
interval	DateTimeIntervalType
filter	OrderFilterType

Výstup (GetReceiptOrdersByDateResult)	
Jméno	**Typ**
result	ResultType
orders	ReceiptOrderType[]

GetReceiptOrder

Funkce vrátí detaily právě jedné objednávky na PŘÍJEM identifikované pomocí čísla objednávky. Pokud nebude objednávka nalezena, je indikován návratový kód operace 3 = NO_DATA (více viz ResultType). Funkce přijímá jako vstup strukturu **GetReceiptOrderParams** a jako výstup poskytuje strukturu **GetReceiptOrderResult**.

Syntaxe:
```
GetReceiptOrderResult GetReceiptOrder(GetReceiptOrderParams)
```

Vstup (GetReceiptOrderParams)	
Jméno	**Typ**
credentials	CredentialsType
order	string

Výstup (GetReceiptOrderResult)	
Jméno	**Typ**
result	ResultType
order	ReceiptOrderType

GetReceiptOrders

Agregační funkce pro hromadné získání detailu až 1000 objednávek na PŘÍJEM identifikovaných pomocí čísel objednávek. Pokud nebude nalezena ani jedna dotazovaná objednávka, je indikován návratový kód operace 3 = NO_DATA (více viz ResultType). Funkce přijímá jako vstup strukturu **GetReceiptOrdersParams** a jako výstup poskytuje strukturu **GetReceiptOrdersResult**.

Syntaxe:
```
GetReceiptOrdersResult GetReceiptOrder(GetReceiptOrdersParams)
```

Vstup (GetReceiptOrdersParams)	
Jméno	**Typ**
credentials	CredentialsType
orders	string[]
filter	OrderFilterType

Výstup (GetReceiptOrdersResult)	
Jméno	**Typ**
result	ResultType
orders	ReceiptOrderType[]

GetReceiptOrderByDocument

Funkce vrátí detaily právě jedné objednávky na PŘÍJEM identifikované pomocí dokladu. Pokud nebude objednávka nalezena, je indikován návratový kód operace 3 = NO_DATA (více viz ResultType). Funkce přijímá jako vstup strukturu **GetReceiptOrderByDocumentParams** a jako výstup poskytuje strukturu **GetReceiptOrderByDocumentResult**.

Syntaxe:

```
GetReceiptOrderByDocumentResult GetReceiptOrderByDocument(GetReceiptOrderByDocumentParams)
```

Vstup (**GetReceiptOrderByDocumentParams**)	
Jméno	**Typ**
credentials	CredentialsType
document	string

Výstup (**GetReceiptOrderByDocumentResult**)	
Jméno	**Typ**
result	ResultType
order	ReceiptOrderType

GetReceiptOrdersByDocument

Agregační funkce pro hromadné získání detailu až 1000 objednávek na PŘÍJEM identifikovaných pomocí dokladů. Pokud nebude nalezena ani jedna dotazovaná objednávka, je indikován návratový kód operace 3 = NO_DATA (více viz ResultType). Funkce přijímá jako vstup strukturu **GetReceiptOrdersByDocumentParams** a jako výstup poskytuje strukturu **GetReceiptOrdersByDocumentResult**.

Syntaxe:

```
GetReceiptOrdersByDocumentResult GetReceiptOrderByDocument(GetReceiptOrdersByDocumentParams)
```

Vstup (**GetReceiptOrdersByDocumentParams**)	
Jméno	**Typ**
credentials	CredentialsType
documents	string[]
filter	OrderFilterType

Výstup (**GetReceiptOrdersByDocumentResult**)	
Jméno	**Typ**
result	ResultType
orders	ReceiptOrderType[]

GetAllIssueOrders

Funkce pro získání všech objednávek na VÝDEJ, které jsou k dispozici pod zadanými přihlašovacími údaji. Funkce přijímá jako vstup strukturu **GetAllIssueOrdersParams** a jako výstup poskytuje strukturu **GetAllIssueOrdersResult**. **POZOR:** Funkce může v rámci jednoho dotazu vrátit maximálně 1000 objednávek. V případě, že by dotaz měl vrátit objednávek více, bude vráceno prvních 1000 (seřazeno sestupně dle požadované expedice). V případě, že si přejete načíst konkrétní objednávky, je možné použít volání funkcí GetIssueOrdersByDate, GetIssueOrder nebo GetIssueOrderByDocument.

Syntaxe:

```
GetAllIssueOrdersResult GetAllIssueOrders(GetAllIssueOrdersParams)
```

Vstup (GetAllIssueOrdersParams)	
Jméno	**Typ**
credentials	CredentialsType
filter	OrderFilterType

Výstup (GetAllIssueOrdersResult)	
Jméno	**Typ**
result	ResultType
orders	IssueOrderType[]

GetIssueOrdersByDate

Funkce pro získání všech objednávek na VÝDEJ, které jsou k dispozici pod zadanými přihlašovacími údaji a zároveň vyhovují nastavenému intervalu data. Interval je aplikován na položku „načteno do systému“. Funkce přijímá jako vstup strukturu **GetIssueOrdersByDateParams** a jako výstup poskytuje strukturu **GetIssueOrdersByDateResult**. **POZOR:** Funkce může v rámci jednoho dotazu vrátit maximálně 1000 objednávek. V případě, že by dotaz měl vrátit objednávek více, bude vráceno prvních 1000 (seřazeno sestupně dle požadované expedice). Pokud k tomu dojde, je možné zúžit nastavený interval data, nebo v případě, že si přejete načíst konkrétní objednávku, je možné použít volání funkcí GetIssueOrder nebo GetIssueOrderByDocument.

Syntaxe:

```
GetIssueOrdersByDateResult GetIssueOrdersByDate(GetIssueOrdersByDateParams)
```

Vstup (GetIssueOrdersByDateParams)	
Jméno	**Typ**
credentials	CredentialsType
interval	DateTimeIntervalType
filter	OrderFilterType

Výstup (GetIssueOrdersByDateResult)	
Jméno	**Typ**
result	ResultType
orders	IssueOrderType[]

GetIssueOrder

Funkce vrátí detaily právě jedné objednávky na VÝDEJ identifikované pomocí čísla objednávky. Pokud nebude objednávka nalezena, je indikován návratový kód operace 3 = NO_DATA (více viz ResultType). Funkce přijímá jako vstup strukturu **GetIssueOrderParams** a jako výstup poskytuje strukturu **GetIssueOrderResult**.

Syntaxe:

```
GetIssueOrderResult GetIssueOrder(GetIssueOrderParams)
```

Vstup (GetIssueOrderParams)	
Jméno	**Typ**
credentials	CredentialsType
order	string

Výstup (GetIssueOrderResult)	
Jméno	**Typ**
result	ResultType
order	IssueOrderType

GetIssueOrders

Agregační funkce pro hromadné získání detailu až 1000 objednávek na VÝDEJ identifikovaných pomocí čísel objednávek. Pokud nebude nalezena ani jedna dotazovaná objednávka, je indikován návratový kód operace 3 = NO_DATA (více viz ResultType). Funkce přijímá jako vstup strukturu **GetIssueOrdersParams** a jako výstup poskytuje strukturu **GetIssueOrdersResult**.

Syntaxe:

```
GetIssueOrdersResult GetIssueOrder(GetIssueOrdersParams)
```

Vstup (GetIssueOrdersParams)	
Jméno	**Typ**
credentials	CredentialsType
orders	string[]
filter	OrderFilterType

Výstup (GetIssueOrdersResult)	
Jméno	**Typ**
result	ResultType
orders	IssueOrderType[]

GetIssueOrderByDocument

Funkce vrátí detaily právě jedné objednávky na VÝDEJ identifikované pomocí dokladu. Pokud nebude objednávka nalezena, je indikován návratový kód operace 3 = NO_DATA (více viz ResultType). Funkce přijímá jako vstup strukturu **GetIssueOrderByDocumentParams** a jako výstup poskytuje strukturu **GetIssueOrderByDocumentResult**.

Syntaxe:

```
GetIssueOrderByDocumentResult GetIssueOrderByDocument(GetIssueOrderByDecumentParams)
```

Vstup (GetIssueOrderByDecumentParams)	
Jméno	**Typ**
credentials	CredentialsType
document	string

Výstup (GetIssueOrderByDocumentResult)	
Jméno	**Typ**
result	ResultType
order	IssueOrderType

GetIssueOrdersByDocument

Agregační funkce pro hromadné získání detailu až 1000 objednávek na VÝDEJ identifikovaných pomocí dokladů. Pokud nebude nalezena ani jedna dotazovaná objednávka, je indikován návratový kód operace 3 = NO_DATA (více viz ResultType). Funkce přijímá jako vstup strukturu **GetIssueOrdersByDecumentParams** a jako výstup poskytuje strukturu **GetIssueOrdersByDocumentResult**.

Syntaxe:

```
GetIssueOrdersByDocumentResult GetIssueOrderByDocument(GetIssueOrdersByDecumentParams)
```

Vstup (GetIssueOrdersByDecumentParams)	
Jméno	**Typ**
credentials	CredentialsType
documents	string[]
filter	OrderFilterType

Výstup (GetIssueOrdersByDocumentResult)	
Jméno	**Typ**
result	ResultType
orders	IssueOrderType[]

GetInventory

Funkce pro získání všech položek inventury, které jsou k dispozici pod zadanými přihlašovacími údaji. Funkce přijímá jako vstup strukturu **GetInventoryParams** a jako výstup poskytuje strukturu **GetInventoryResult**.

Syntaxe:

```
GetInventoryResult GetInventory(GetInventoryParams)
```

Vstup (GetInventoryParams)	
Jméno	**Typ**
credentials	CredentialsType
filter	InventoryFilterType
filterArg	string

Výstup (GetInventoryResult)	
Jméno	**Typ**
result	ResultType
inventoryItems	InventoryItemType[]

GetCarriers

Funkce pro získání všech aktuálních dopravců, které jsou k dispozici pod zadanými přihlašovacími údaji. Funkce přijímá jako vstup strukturu **GetCarriersParams** a jako výstup poskytuje strukturu **GetCarriersResult**.

Syntaxe:

```
GetCarriersResult GetCarriers(GetCarriersParams)
```

Vstup (GetCarriersParams)	
Jméno	**Typ**
credentials	CredentialsType

Výstup (GetCarriersResult)	
Jméno	**Typ**
result	ResultType
carrierItems	CarrierItemType[]

GetImportResults

Funkce pro získání jednoho nebo i více výsledků importu, které jsou k dispozici pod zadanými přihlašovacími údaji. Funkce přijímá jako vstup strukturu **GetImportResultsParams** a jako výstup poskytuje strukturu **GetImportResultsResult**. V případě, že je nastaven `includePDF`, budou

položky v odpovědi doplněny o PDF report daného importu. **POZOR:** Funkce může v rámci jednoho dotazu přijmout požadavek na získání maximálně 100 výsledků importu. V případě, že by dotaz obsahoval požadavek na získání více výsledků importu, bude zpracováno prvních 100 požadavků v seznamu transactionIds.

Syntaxe:

```
GetImportResultsResult GetImportResults(GetImportResultsParams)
```

Vstup (GetImportResultsParams)	
Jméno	**Typ**
credentials	CredentialsType
transactionIds	string[]
includePDF	bool

Výstup (GetImportResultsResult)	
Jméno	**Typ**
result	ResultType
importResults	ImportResultType[]

ImportPartners

Funkce pro import seznamu partnerů. Informace budou přeneseny na server Servant zvoleného střediska pod ukladatelem dle zadaných přihlašovacích údajů. Funkce přijímá jako vstup strukturu **ImportPartnersParams** a jako výstup poskytuje strukturu **ImportPartnersResult**.

Syntaxe:

```
ImportPartnersResult ImportProducts(ImportPartnersParams)
```

Vstup (ImportPartnersParams)	
Jméno	**Typ**
credentials	CredentialsType
warehouse	string
partners	ImportPartnerDefinitionType[]

Výstup (ImportPartnersResult)	
Jméno	**Typ**
result	ResultType
importRequestResult	ImportRequestResultType

ImportProducts

Funkce pro import sortimentu, jeho EAN a prováděných VAS. Informace budou přeneseny na server Servant zvoleného střediska pod ukladatelem dle zadaných přihlašovacích údajů. Funkce přijímá jako vstup strukturu **ImportProductsParams** a jako výstup poskytuje strukturu **ImportProductsResult**.

Syntaxe:
```
ImportProductsResult ImportProducts(ImportProductsParams)
```

Vstup (ImportProductsParams)	
Jméno	**Typ**
credentials	CredentialsType
warehouse	string
products	ImportProductType[]

Výstup (ImportProductsResult)	
Jméno	**Typ**
result	ResultType
importRequestResult	ImportRequestResultType

ImportReceipt

Funkce pro import objednávky na příjem. Informace budou přeneseny na server Servant zvoleného střediska pod ukladatelem dle zadaných přihlašovacích údajů. Funkce přijímá jako vstup strukturu **ImportReceiptParams** a jako výstup poskytuje strukturu **ImportReceiptResult**. **POZNÁMKA:** Přestože vstupní struktura umožňuje zadat soubory, není v současné době žádný soubor určený pro objednávky na příjem podporovaný. Jakékoli nastavení souboru bude mít za následek chybu importu.

Syntaxe:
```
ImportReceiptResult ImportReceipt(ImportReceiptParams)
```

Vstup (ImportReceiptParams)	
Jméno	**Typ**
credentials	CredentialsType
warehouse	string
receipt	ImportReceiptType
files	ImportFileType[]

Výstup (ImportReceiptResult)	
Jméno	**Typ**
result	ResultType
importRequestResult	ImportRequestResultType

ImportIssue

Funkce pro import objednávky na výdej. Informace budou přeneseny na server Servant zvoleného střediska pod ukladatelem dle zadaných přihlašovacích údajů. Funkce přijímá jako vstup strukturu **ImportIssueParams** a jako výstup poskytuje strukturu **ImportIssueResult**. V rámci importu je možné připojit právě jeden soubor expedice a právě jeden soubor s vlastním přepravním štítkem.

Syntaxe:

```
ImportIssueResult ImportIssue(ImportIssueParams)
```

Vstup (**ImportIssueParams**)	
Jméno	**Typ**
credentials	CredentialsType
warehouse	string
issue	ImportIssueType
files	ImportFileType[]

Výstup (**ImportIssueResult**)	
Jméno	**Typ**
result	ResultType
importRequestResult	ImportRequestResultType

ImportIssues

Funkce pro dávkový import až 100 objednávek na výdej v rámci jednoho dotazu. Informace budou přeneseny na server Servant zvoleného střediska pod ukladatelem dle zadaných přihlašovacích údajů. Funkce přijímá jako vstup strukturu **ImportIssuesParams** a jako výstup poskytuje strukturu **ImportIssuesResult**. V rámci importu je možné ke každé jedné objednávce připojit právě jeden soubor expedice a právě jeden soubor s vlastním přepravním štítkem.

Syntaxe:

```
ImportIssuesResult ImportIssue(ImportIssuesParams)
```

Vstup (**ImportIssuesParams**)	
Jméno	**Typ**
credentials	CredentialsType
warehouse	string
issues	ImportIssueWithFilesType[]

Výstup (**ImportIssuesResult**)	
Jméno	**Typ**
results	ImportIssueResultType[]

ImportFiles

Funkce pro import souboru(ů) k již existující objednávce. Funkce přijímá jako vstup strukturu **ImportFilesParams** a jako výstup poskytuje strukturu **ImportFilesResult**.

Syntaxe:

```
ImportFilesResult ImportFiles(ImportFilesParams)
```

Vstup (**ImportFilesParams**)	
Jméno	**Typ**
credentials	CredentialsType
order	string
files	ImportFileType[]

Výstup (**ImportFilesResult**)	
Jméno	**Typ**
result	ResultType
importRequestResult	ImportRequestResultType

Struktury

Povinné položky mají typ tučnou kurzívou (***string***). Nepovinné položky mají typ standardně (string).

OrderFilterType

Filtr aplikovaný na objednávky. Filtr může být nenastaven, nebo nabývat jedné z hodnot níže.

`MostRelevant` ..... v případě více záznamů se stejným číslem objednávky vrací pouze nejnovější (nejvíce relevantní) záznamy

`IncludeHidden` ..... vrací i objednávky, které jsou na WSS označené jako skryté

InventoryFilterType

Filtr aplikovaný na položky inventury. Filtr může být nenastaven, nebo nabývat jedné z hodnot níže.

`NonZero` ..... vrací pouze záznamy s nenulovým množstvím
- parametr `filterArg` není očekáván

`IncludeHidden` vrací i položky, které jsou na WSS označené jako skryté, a zároveň u těchto položek zahrnuje i skryté čárové kódy
- parametr `filterArg` není očekáván

`Movement` ..... vrátí pouze záznamy se změnou množství od určitého data
- parametr `filterArg` obsahuje datum ve formátu ISO 8601 (PHP `DateTime::ATOM`, např. 2026-05-21T12:00:00+02:00).

ResultType

Struktura obsahující výsledek operace.

Jméno	Typ	Význam	Platné hodnoty
resultCode	***int***	Návratový kód operace	0 = OK 1 = INTERNAL_EXCEPTION 2 = INVALID_CREDENTIALS 3 = NO_DATA 4 = LIMITED_DATA 10 až 20 = API_REQUESTS_WARNING_REACHED 20 až 30 = API_ACCESS_TIME_WARNING_REACHED 30 = API_REQUESTS_LIMIT_REACHED 40 = API_ACCESS_TIME_LIMIT_REACHED
resultString	***string***	Textová reprezentace návratového kódu operace	

`OK` – operace skončila bez chyby

`INTERNAL_EXCEPTION` – při zpracování dotazu došlo k interní chybě serveru

`INVALID_CREDENTIALS` přihlašovací údaje jsou neplatné, nebo nemá daný uživatel k informacím přístup

`NO_DATA` – operace byla dokončena, nicméně server neodeslal žádná data (parametrům dotazu neodpovídají žádná data; zkuste změnit parametry dotazu)

`LIMITED_DATA` – operace byla dokončena, nicméně server vrátil méně dat, než je na serveru k dispozici (parametrům dotazu odpovídá více dat, než je server schopný v rámci jednoho dotazu poskytnout; zkuste změnit parametry dotazu)

`API_REQUESTS_WARNING_REACHED` – varování – denní limit API dotazů se blíží; kód varování je kombinován s kódem `OK`, `NO_DATA`, nebo `LIMITED_DATA` – popisek i kód varování se sčítají

`API_ACCESS_TIME_WARNING_REACHED` – varování – denní limit přístupové času k serveru se blíží; kód varování je kombinován s kódem `OK`,

	`NO_DATA`, nebo `LIMITED_DATA` – popisek i kód varování se sčítají
`API_REQUESTS_LIMIT_REACHED`	– chyba – denní limit API dotazů byl dosažen; API komunikace je až do dalšího dne zastavena
`API_ACCESS_TIME_LIMIT_REACHED`	– chyba – denní limit přístupové času k serveru byl dosažen; API komunikace je až do dalšího dne zastavena

CredentialsType

Struktura obsahující informace o uživateli požadující provedení operace.

Jméno	Typ	Význam	Platné hodnoty
email	***string***	E-mailová adresa registrovaná pro přihlášení do WSS www.webskladservant.cz	Platná emailová adresa, maximální délka je 60 znaků
password	***string***	Heslo pro přihlášení pod zadanou e-mailovou adresou	
switchForUser	bool	Nastavte na `true`, pokud si přejete přihlásit jako jiný uživatel	`null` nebo `true`/`false`
switchForUserId	int	Pokud k tomu má přihlašovaný uživatel právo, přepne se po přihlášení interně uživatel na uživatele identifikovaného pomocí tohoto ID	`null` nebo `<-1,1000>`
accessId	string	Identifikátor přístupu	`null` nebo textový řetězec identifikující zdroj přístupu

Hodnoty switchForUser, switchForUserId a accessId jsou mimo výjimky použity pro vývoj vlastního SW rozhraní webové služby. Pro běžný přístup, prosím, hodnoty nenastavujte, nebo jej nastavte na `switchForUser = false`, `switchForUserId = -1` a `accessId = ""`.

DateTimeIntervalType

Struktura umožňující nastavení rozsahu data a času.

Jméno	Typ	Význam	Platné hodnoty
from	***dateTime***	Datum a čas „od“	Při implementaci vlastního serializéru / deserializéru je potřeba dodržet specifikace dateTime formátu definovaného XSD
to	***dateTime***	Datum a čas „do“	

AttributesType

Textové pole o velikosti 0 až 3 položky. Maximální délka první položky je 45 znaků. Maximální délka dalších dvou položek je 20 znaků.

ActivityType

Struktura obsahující informace o VAS prováděných nad položkou objednávky.

Jméno	Typ	Význam	Platné hodnoty
id	***int***	Číselný identifikátor VAS	≥ 0
name	***string***	Textové pojmenování VAS	Maximální délka je 120 znaků
value	***string***	Výstup VAS	Maximální délka je 120 znaků
count	***float***	Množství zboží	Při implementaci vlastního serializéru / deserializéru je potřeba dodržet specifikace decimal formátu definovaného XSD, platnost je na 2 desetinná místa
servantNumber	***string***	Balíkový štítek - číslo přidělené společností Servant a.s.	Maximální délka je 16 znaků
transportNumber	***string***	Číslo přidělené dopravcem (číslo dopravy)	Maximální délka je 32 znaků

ItemType

Struktura obsahující informace sdílené pro položky objednávky a položky inventury.

Jméno	Typ	Význam	Platné hodnoty
code	***string***	Kód sortimentu	Maximální délka je 50 znaků
name	***string***	Jméno sortimentu	Maximální délka je 120 znaků
attributes	***AttributesType***	Atributy	
expiration	***date***	Expirace	Při implementaci vlastního serializéru / deserializéru je potřeba dodržet specifikace date formátu definovaného XSD
batch	***string***	Šarže	Maximální délka je 45 znaků
unit	***string***	Měrná jednotka	Maximální délka je 8 znaků

OrderItemType

Struktura obsahující informace o jedné položce objednávky.

Jméno	Typ	Význam	Platné hodnoty
item	***ItemType***	Struktura obsahující informace sdílené pro položky objednávky a položky inventury	
requiredCount	***float***	Požadovaný počet kusů	Při implementaci vlastního serializéru / deserializéru je potřeba dodržet specifikace decimal formátu definovaného XSD, platnost je na 2 desetinná místa
count	***float***	Reálně vyskladněný počet kusů	
activities	***ActivityType[]***	VAS prováděné nad položkami objednávky	

TransportItemType

Struktura obsahující informace o jedné položce dopravy objednávky.

Jméno	Typ	Význam	Platné hodnoty
unit	***string***	Nákladová jednotka přepravy	Maximální délka je 30 znaků
servantNumber	***string***	Balíkový štítek - číslo přidělené společností Servant a.s.	Maximální délka je 60 znaků
transportNumber	***string***	Číslo přidělené dopravcem (číslo dopravy)	Maximální délka je 60 znaků
statusUrl	***string***	URL pro sledování stavu dopravy na stránkách dopravce	
packageType	***string***	Typ přepravy (pro Českou Poštu se může jednat např. o DR, NP, apod.)	
weight	***float***	Hmotnost	Při implementaci vlastního serializéru / deserializéru je potřeba dodržet specifikace decimal formátu definovaného XSD, platnost je na 2 desetinná místa
dimension	DimensionType	Rozměry	`null` nebo platná struktura

Při osobní (OSB), individuální (IND) nebo vlastní dopravě s přepravním štítkem (PDF) není URL pro sledování dopravy na stránkách dopravce k dispozici a tato položka je vždy nulová. Stav dopravy lze vyčíst také přímo ze struktury IssueOrderType z položky „state“ (není nutné stav zjišťovat ze stránek dopravce).

CashOnDeliveryType

Struktura obsahující informace o dobírce dopravy.

Jméno	Typ	Význam	Platné hodnoty
amount	***float***	Cena dobírky v nastavené měně v případě, že je dobírka nastavená. V opačném případě je položka nastavená na 0.	Při implementaci vlastního serializéru / deserializéru je potřeba dodržet specifikace decimal formátu definovaného XSD, platnost je na 2 desetinná místa
currency	***string***	Měna dobírky	Formát dle ISO 4217
variableSymbol	***string***	Variabilní symbol, případně další informace potřebné pro identifikaci platby (oddělené pomocí ^)	Maximální délka je 30 znaků

DimensionType

Definice je totožná s typem ImportProductDimensionType.

EanType

Struktura obsahující definici čárového kódu (balení) zboží.

Jméno	Typ	Význam	Platné hodnoty
ean	***string***	EAN	Maximální délka je 64 znaků
count	***float***	Počet MJ k EAN	Při implementaci vlastního serializéru / deserializéru je potřeba dodržet specifikace decimal formátu definovaného XSD
weight	***float***	Hmotnost v kg	
dimension	DimensionType	Rozměry	`null` nebo platná struktura
velocity	***float***	Objem v m^3	Při implementaci vlastního serializéru / deserializéru je potřeba dodržet specifikace decimal formátu definovaného XSD
primary	***bool***	Výchozí EAN pro výpočty a tiskové sestavy	`true` nebo `false`

ReceiptOrderType

Struktura obsahující informace o objednávce na PŘÍJEM.

Jméno	Typ	Význam	Platné hodnoty
order	***string***	Označení objednávky	Maximální délka je 64 znaků
document	***string***	Číslo dokladu (např. 20-HU2-0203814)	Maximální délka je 60 znaků
invoice	***string***	Faktura	Maximální délka je 60 znaků
orderItems	***OrderItemType[]***	Jednotlivé položky objednávky	
activities	***ImportActivityType[]***	VAS prováděné nad celou objednávkou	
partnerName	***string***	Jméno partnera	Maximální délka je 60 znaků
partnerCode	***string***	Kód partnera	Maximální délka je 60 znaků
loaded	***dateTime***	Datum a čas načtení do systému	Při implementaci vlastního serializéru / deserializéru je potřeba dodržet specifikace dateTime formátu definovaného XSD
received	***dateTime***	Datum a čas fyzického přijetí objednávky	
checkState	***string***	Popis stavu kontroly zboží	Maximální délka je 120 znaků
finished	***dateTime***	Datum a čas dokončení příjmu	Při implementaci vlastního serializéru / deserializéru je potřeba dodržet specifikace dateTime formátu definovaného XSD
error	***dateTime***	Datum a čas chyby	

Všechny položky typu „date“ nebo „dateTime“ mohou být nastavené nebo nenastavené. Pokud je položka nenastavená, znamená to, že k události ještě nedošlo. Nenastavenou položku lze identifikovat tak, že je hodnota `< 1.1.2000 00:00:00`.

Položky objednávky („orderItems“) jsou načteny a uvedeny v datové struktuře typicky (pokud nezvolil ukladatel jinak) až po 18. hodině toho dne, ve kterém došlo k jejich fyzickému naskladnění. Do té doby je indikováno 0 položek objednávky.

IssueOrderType

Struktura obsahující informace o objednávce na VÝDEJ.

Jméno	Typ	Význam	Platné hodnoty
order	***string***	Označení objednávky	Maximální délka je 64 znaků
document	***string***	Číslo dokladu (např. 20-HU2-0203814)	Maximální délka je 60 znaků
invoice	***string***	Faktura	Maximální délka je 60 znaků
orderItems	***OrderItemType[]***	Jednotlivé položky objednávky	
activities	***ImportActivityType[]***	VAS prováděné nad celou objednávkou	
transportItems	***TransportItemType[]***	Jednotlivé položky dopravy (přepravní balíky, palety, atd.)	
partnerName	***string***	Jméno partnera	Maximální délka je 60 znaků
partnerCode	***string***	Kód partnera	Maximální délka je 60 znaků
recipientAddress	***string***	Doručovací adresa	Maximální délka je 120 znaků
cashOnDelivery	***CashOnDeliveryType***	Informace o dobírce dopravy	
note	***string***	Poznámka	Maximální délka je 120 znaků
state	***string***	Poslední stav dopravy	Maximální délka je 60 znaků
warning	***string***	Varování signalizované API	Maximální délka je 120 znaků
error	***dateTime***	Objednávka obsahuje chybu (například, že sortiment objednávky nebylo možné vykrýt v požadovaném množství)	Při implementaci vlastního serializéru / deserializéru je potřeba dodržet specifikace dateTime formátu definovaného XSD
requiredDispatch	***date***	Požadovaný datum expedice	Při implementaci vlastního serializéru / deserializéru je potřeba dodržet specifikace date formátu definovaného XSD
dispatch	***dateTime***	Datum a čas expedice	Při implementaci vlastního serializéru / deserializéru je potřeba dodržet specifikace dateTime formátu definovaného XSD
loaded	***dateTime***	Datum a čas načtení do systému	
entered	***dateTime***	Datum a čas zadání do výroby	
packaged	***dateTime***	Datum a čas zabalení	
transportCarrier	***string***	Dopravce	Maximální délka je 30 znaků
transportEntered	***dateTime***	Datum a čas přidělení čísla/čísel dopravy	Při implementaci vlastního serializéru / deserializéru je potřeba dodržet specifikace dateTime formátu definovaného XSD
transportPicked	***dateTime***	Datum a čas vyzvednutí dopravcem	
transportDelivered	***dateTime***	Datum a čas doručení	

V případě, že objednávka obsahuje více položek dopravy, reflektuje položka „state“ stav všech těchto položek.

Dopravce může nabývat jedné z následujících hodnot:

`GLS`	GLS (https://gls-group.eu/CZ/cs/home)
`CP`	Česká Pošta (https://www.ceskaposta.cz/index)
`FOFR`	FOFR (https://www.fofrcz.cz/)
`TOPT`	TOPTRANS (https://www.toptrans.cz/)
`GEIS`	Geis (https://www.geis-group.cz/cs)
`ZAS`	Zásilkovna (https://www.zasilkovna.cz/)
`DPD`	DPD (https://www.dpd.com/cz/cs/)
`IND`	individuální doprava
`OSB`	osobní odběr
`PDF`	vlastní doprava s přepravním štítkem

Všechny položky typu „date“ nebo „dateTime“ mohou být nastavené nebo nenastavené. Pokud je položka nenastavená, znamená to, že k události ještě nedošlo. Nenastavenou položku lze identifikovat tak, že je hodnota `< 1.1.2000 00:00:00`.

Položky objednávky („orderItems“) jsou načteny a uvedeny v datové struktuře typicky (pokud nezvolil ukladatel jinak) až po 18. hodině toho dne, ve kterém došlo k jejich fyzickému vyskladnění. Do té doby je indikováno 0 položek objednávky.

Informace o dopravě (všechny položky „transport*“) je možné získat až ve chvíli, kdy dopravce převezme balíky, palety, atd. Do doby převzetí je indikováno 0 položek dopravy a veškeré další informace o dopravě nejsou nastavené.

Při individuální dopravě nebo osobním odběru je indikováno doručení ve chvíli, kdy dojde k převzetí zboží.

InventoryItemType

Struktura obsahující informace o položce inventury.

Jméno	Typ	Význam	Platné hodnoty
item	***ItemType***	Struktura obsahující informace sdílené pro položky objednávky a položky inventury	
count	***float***	Aktuální fyzické množství	Při implementaci vlastního serializéru / deserializéru je potřeba dodržet specifikace decimal formátu definovaného XSD, platnost je na 2 desetinná místa
requestedCount	***float***	Aktuální evidenční množství	
group	***string***	Skupina	Maximální délka je 60 znaků
eans	***EanType[]***	Seznam definic čárového kódu (balení) zboží	
checkExpiration	***bool***	Nastaveno na `true`, pokud se má u zboží kontrolovat expirace	`true` nebo `false`
checkBatch	***bool***	Nastaveno na `true`, pokud se má u zboží kontrolovat šarže	
checkSerialNumbers	***bool***	Nastaveno na `true`, pokud se má u zboží sledovat výrobní čísla	
printLabel	***bool***	Nastaveno na `true`, pokud se má na zboží se lepit štítek	
movement	***date***	Datum posledního pohybu (změny aktuálního počtu kusů)	Při implementaci vlastního serializéru / deserializéru je potřeba dodržet specifikace date formátu definovaného XSD

CarrierItemType

Struktura obsahující informace o dopravci.

Jméno	Typ	Význam	Platné hodnoty
id	***string***	ID dopravce	Viz seznam dopravců uvedený v definici struktury IssueOrderType
name	***string***	Jméno dopravce (lidsky čitelné jméno, nebo jméno dopravce dle interního pojmenování Servant)	

ImportIssueResultType

Struktura agregující informace o výsledku importu jedné objednávky na výdej.

Jméno	Typ	Význam	Platné hodnoty
order	***string***	Objednávka	
result	***ResultType***	Výsledek dotazu	
importRequestResult	***ImportRequestResultType***	Výsledek operace importu	

ImportRequestResultType

Struktura obsahující informace o výsledku operace vložení importu ke zpracování. Pokud je vložení úspěšné, vlastní výsledek importu obsahuje struktura ImportResultType.

Jméno	Typ	Význam	Platné hodnoty
transactionId	***string***	ID transakce, pod kterým lze následně ověřit výsledek importu	SHA256
success	***bool***	Nastaveno na `true`, pokud bylo vložení importu ke zpracování úspěšné	`true` nebo `false`

resultCode	***int***	Návratový kód operace	`0 = Success`
			`1 = PrerequestCheckError`
			`2 = ServerAccessError`
			`3 = ImportRequestError`
			`4 = PostrequestOperationError`

result	***string***	Textová reprezentace návratového kódu operace	
infoText	***string***	Doplňující informace (typicky nastavené, pokud dojde k chybě vložení importu)	

Význam návratového kódu operace:

`Success`	import úspěšně zadán ke zpracování
`PrerequestCheckError`	chyba při kontrole vstupních dat (webový server)
`ServerAccessError`	chyba přístupu k serveru
`ImportRequestError`	chyba zadání importu (Servant server)
`PostrequestOperationError`	chyba operací vykonávaných po úspěšném zadání importu (interní chyba webového serveru)

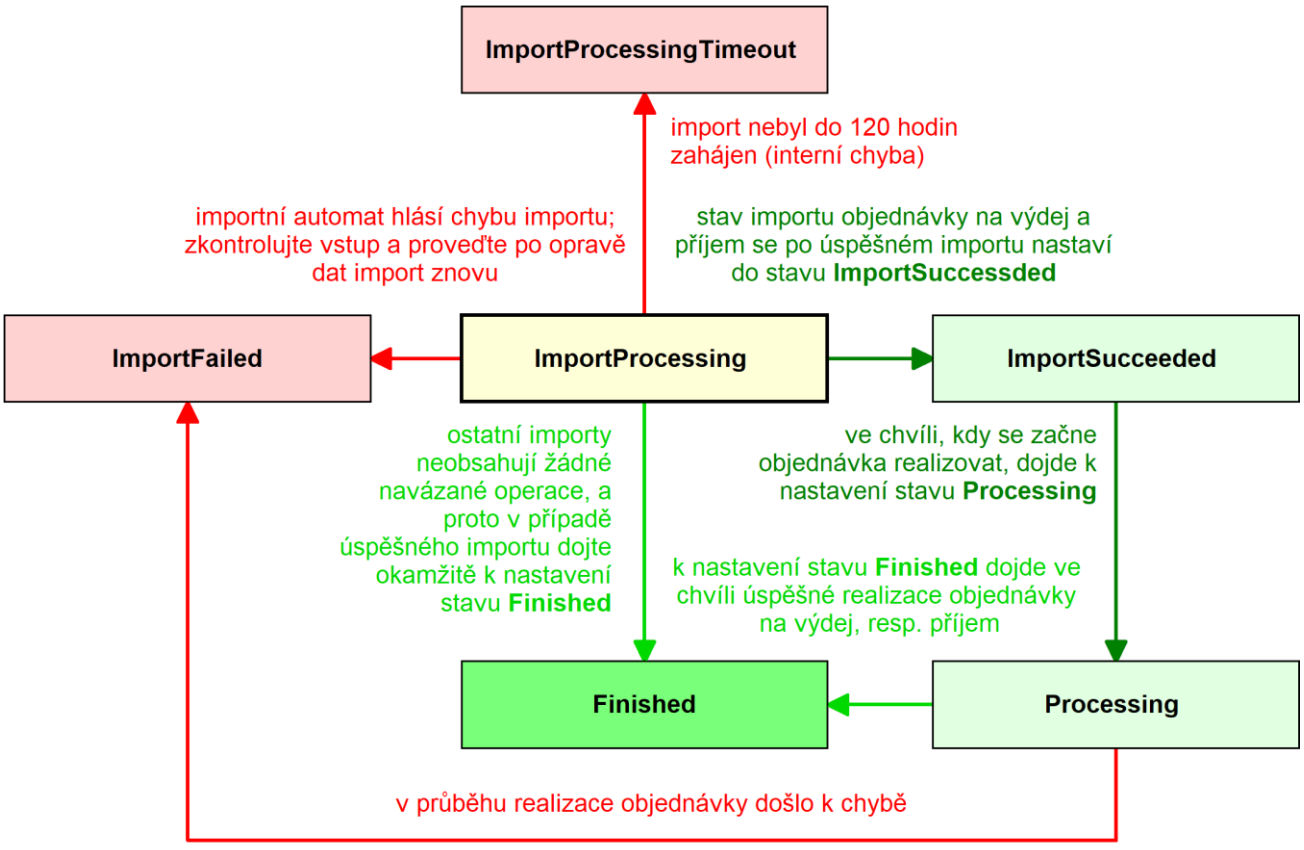
ImportResultType

Struktura obsahující informace o výsledku importu.

Jméno	Typ	Význam	Platné hodnoty
transactionId	***string***	ID transakce	SHA256
closed	***bool***	Nastaveno na `true`, pokud byl dokončen import i případné navázané operace	`true` nebo `false`
error	***bool***	Nastaveno na `true`, pokud v průběhu importu došlo k chybě	
state	***string***	Aktuální stav importu nebo případných navázaných operací	Platné hodnoty viz definice níže …
infoTexts	InfoTextType[]	Doplňující informace k danému stavu	`null` nebo ≥ 0 položek
encodedContentPDF	string	PDF report importu	`null` nebo string obsahující PDF data zakódovaná dle Base64 standardu

Aktuální stav může nabývat jedné z následujících hodnot:

`ImportProcessing`	Import právě probíhá …
`ImportSucceeded`	Import skončil úspěšně; očekávají se další navázané operace
`Processing`	Probíhají další operace (vlastní realizace objednávky) …
`Finished`	Import i případné další navázané operace byly úspěšně dokončeny
`ImportFailed`	Import skončil s chybou
`ImportProcessingTimeout`	Došlo k timeoutu importu ze strany Servant serveru (interní chyba)



InfoTextType

Struktura obsahující rozšiřující textové informace, které mohou být doplněny o JSON data zakódovaná dle Base64 standardu.

Jméno	Typ	Význam	Platné hodnoty
text	**string**	Textová informace	
data	string	Data	`null` nebo string obsahující JSON data zakódovaná dle Base64 standardu

ImportBankType

Struktura obsahující informace o bankovním spojení.

Jméno	Typ	Význam	Platné hodnoty
name	***string***	Jméno banky	Maximální délka je 64 znaků
code	***string***	Kód banky	Právě 4 ASCII číslice (např. 0100)
accountNumber	***string***	Číslo účtu	Maximální délka je 24 znaků
specificSymbol	string	Specifický symbol	`null` nebo string o maximální délce 4 znaky

ImportPartnerDefinitionType

Struktura obsahující informace o partnerovi.

Jméno	Typ	Význam	Platné hodnoty
type	***string***	Typ partnera	
partner	***ImportPartnerType***	Definice partnera	
legalForm	string	Právní forma partnera	`null` nebo string o maximální délce 48 znaků
payingVAT	bool	Plátce DPH	`null` nebo `true`/`false`
website	string	Webové stránky	`null` nebo string o maximální délce 64 znaků
bank	ImportBankType	Bankovní spojení na partnera	`null` nebo platná struktura
expirationDays	int	Neposílat zboží, jehož expirace končí za kratší počet dnů než zde uvedený	`null` nebo kladná celočíselná hodnota

Typ partnera může nabývat jednu z následujících hodnot:

`Carrier` pro dopravce
`Supplier` pro dodavatele
`Customer` pro odběratele

ImportItemIdType

Struktura obsahující informace definující zboží. Společně se strukturou ImportItemIdDetailsType tvoří kompletní identifikátor zboží.

Jméno	Typ	Význam	Platné hodnoty
code	***string***	Jedinečný kód sortimentu. Pokud není vyplněn, přidělíme zboží vlastní kód	Jedinečný kód v rámci ukladatele. Maximální délka je 50 znaků
attributes	AttributesType	Atributy sortimentu. Velikost, barva, rozměr apod. Pokud je tento údaj vyplněný, je povinný, a musí se používat!	`null` nebo platná struktura

ImportProductItemDetailsType

Struktura obsahující podrobnosti o zboží.

Jméno	Typ	Význam	Platné hodnoty
name	***string***	Název zboží	Maximální délka je 255 znaků
shortName	string	Krátký název zboží	`null` nebo string o maximální délce 32 znaků
description	string	Popis zboží	`null` nebo string o maximální délce 255 znaků
unit	***string***	Množstevní jednotka (MJ)	Maximální délka je 8 znaků
type	string	Typ sortimentu	`null` nebo string o maximální délce 128 znaků
group	string	Skupina sortimentu	`null` nebo string o maximální délce 64 znaků
note	string	Poznámka ke zboží	`null` nebo string o maximální délce 255 znaků
notePrint	string	Poznámka, která se tiskne na některých štítcích a tiskových sestavách	`null` nebo string o maximální délce 64 znaků
fifo	bool	Má se pro tento sortiment používat FIFO?	`null` nebo `true`/`false`
supplierCode	string	Kód dodavatele nebo výrobce zboží	`null` nebo string o maximální délce 64 znaků

ImportProductDimensionType

Struktura obsahující fyzické rozměry zboží.

Jméno	Typ	Význam	Platné hodnoty
x	***float***	Rozměr X v metrech	Při implementaci vlastního serializéru / deserializéru je potřeba dodržet specifikace decimal formátu definovaného XSD
y	***float***	Rozměr Y v metrech	
z	***float***	Rozměr Z v metrech	

ImportProductEanType

Struktura obsahující definici čárového kódu (balení) zboží.

Jméno	Typ	Význam	Platné hodnoty
ean	***string***	EAN	Jedinečný kód v rámci ukladatele. Maximální délka je 64 znaků
count	***float***	Počet MJ k EAN	Povinná hodnota (hodnoty ≤ 0 jsou neplatné).
eanCount	float	Počet MJ v balení, které patří k tomuto EAN	Může být `null`, 0, nebo libovolné kladné číslo. Při implementaci vlastního serializéru / deserializéru je potřeba dodržet specifikace decimal formátu definovaného XSD
countToPlacement	float	MJ balení do umístění na jednu pozici (0,8 x 1,2 x 1,6 m)	
expandability	float	Na kolik částí lze balení rozdělit	
weight	float	Hmotnost v kg	
dimension	ImportProductDimensionType	Rozměry	`null` nebo platná struktura
velocity	float	Objem v m^3	Může být `null`, 0, nebo libovolné kladné číslo. Při implementaci vlastního serializéru / deserializéru je potřeba dodržet specifikace decimal formátu definovaného XSD
primary	bool	Výchozí EAN pro výpočty a tiskové sestavy	`null` nebo `true`/`false`

Rozměr zboží musí být zadán buď formou fyzických rozměrů (`dimension`) nebo objemu (`velocity`). V případě nastavení formou fyzických rozměrů je možné nastavit `velocity` na `0`.

V případě nastavení formou objemu je možné nenastavovat `dimension`, případně všechny jeho rozměry nastavit na `0`.

Smazat EAN nelze (pouze zásahem na naší straně). EAN lze tedy jen přidávat.

Systém automaticky kontroluje unikátnost EAN kódu v rámci již existujícího sortimentu a přiřazených EAN kódů. V rámci kontroly může dojít ke dvěma situacím:

1. EAN kód není dosud nikde evidován = EAN je přiřazen sortimentu
2. EAN kód již je někde evidován (přiřazen nějakému již zadanému sortimentu) = provede se kontrola aktuálního množství sortimentu, ke kterému je EAN přiřazen
    a. V případě 0 (nulového množství) dojde k odstranění EAN kódu ze sortimentu s nulovým množstvím a přiřazení EAN kódu k právě zadávanému sortimentu
    b. V případě nenulového množství dojde k chybě

ImportItemIdDetailsType

Struktura obsahující informace upřesňující zboží. Společně se strukturou ImportItemIdType tvoří kompletní identifikátor zboží.

Jméno	Typ	Význam	Platné hodnoty
expiration	date	Expirace	Může být `null`. Při implementaci vlastního serializéru / deserializéru je potřeba dodržet specifikace date formátu definovaného XSD
batch	string	Šarže	`null` nebo string o maximální délce 45 znaků

Pokud nechcete expiraci uvádět, je možné nastavit libovolný datum `< 1.1.2000`

Pokud nebude některé z polí `expiration` nebo/a `batch` nastaveno, vybere konkrétní zboží náš systém. Pokud se uvede, systém vybere podle uvedené hodnoty.

ImportLabelType

Struktura obsahující definici štítku.

Jméno	Typ	Význam	Platné hodnoty
type	***int***	Lepit štítek z číselníku štítku ukladatele	Celé číslo mezi hodnotami 10 a 20 včetně
specification	string	Jazyková mutace nebo jiné upřesnění štítku	`null` nebo string o maximální délce 3 znaky

ImportActivityType

Struktura obsahující definici VAS prováděných nad zbožím.

Jméno	Typ	Význam	Platné hodnoty
checkExpiration	bool	Nastaveno na `true`, pokud se má u zboží kontrolovat expirace	`null` nebo `true`/`false`
checkBatch	bool	Nastaveno na `true`, pokud se má u zboží kontrolovat šarže	
checkSerialNumbers	bool	Nastaveno na `true`, pokud se má u zboží sledovat výrobní čísla	
printLabel	bool	Nastaveno na `true`, pokud se má na zboží se lepit štítek	
labels	ImportLabelType[]	Definice štítků	`null` nebo 10 ≥ položek ≥ 0
services	string[]	Provést služby dle číselníku VAS	`null` nebo 10 ≥ položek ≥ 0, maximální délka jedné služby jsou 3 číslice v rozsahu 021-999
documentsType	string	K sortimentu přiložit dokumenty dle číselníků	`null` nebo string o maximální délce 255 znaků

Pokud chceme k nějaké službě VAS doplnit data, můžeme tak učinit doplněním dvojtečky, která je následovaná vlastními daty (příklad: `96:BCX123456OK`)

K VAS službě je možné připojit i soubor a to ihned při odeslání, nebo pouze nastavit referenci na v budoucnu odeslaný soubor. V obou případech nastavte jako VAS data jméno souboru (příklad: `96:receipt.pdf`) a následně:

- V případě okamžitého odeslání společně s objednávkou připojte soubor jako ImportFileType
- V případě pozdějšího odeslání soubor odešlete pomocí funkce ImportFiles

… **POZOR:** že v obou případech musí být jméno souboru v datech VAS služby a ve struktuře ImportFileType nastavené stejně.

Seznam možných VAS služeb a očekávaných dat naleznete v tomto dokumentu.

ImportProductActivityType

Struktura obsahující definici VAS prováděných nad zbožím nebo celou objednávkou. Tato struktura je proti struktuře ImportActivityType použita výhradně pro definici VAS v rámci funkce pro import zboží.

Jméno	Typ	Význam	Platné hodnoty
idDetails	***ImportItemIdDetailsType***	Definice zboží	
activity	***ImportActivityType***	VAS prováděná nad zbožím	

ImportProductType

TOP struktura obsahující definici právě jednoho zboží, jeho EAN a VAS, které se mají nad zbožím provádět.

Jméno	Typ	Význam	Platné hodnoty
id	***ImportItemIdType***	Definice zboží	
details	ImportProductItemDetailsType	Podrobnosti o zboží	`null` nebo platná struktura
eans	***ImportProductEanType[]***	Definici EAN zboží	≥ 1 položek (minimálně jeden EAN k zboží je povinný)
activities	ImportProductActivityType[]	Definice VAS prováděných nad zbožím	`null` nebo ≥ 0 položek
files	ImportFileType[]	Obrázky zboží	`null` nebo ≥ 0 položek (jeden hlavní a až 2 další obrázky)

Definice zboží je povinná. Ostatní atributy je možné vyplnit nebo ponechat prázdné. Budou importovány jen ty části, které budou vyplněné. Při nevyplnění ani jednoho atributu (podrobnosti o zboží, definice EAN zboží nebo definice VAS prováděných nad zbožím), import selže.

Při importu sortimentu se stejným `code` ve struktuře ImportItemIdType nedojde k chybě duplicity, ale k aktualizaci položky s následujícím pravidlem - rozsah aktualizovaných atributů ze struktury ImportProductItemDetailsType se liší podle aktuálního množství skladem – při nulovém množství se aktualizují i základní identifikační údaje (`name`, `shortName`, `type` a `unit`), zatímco při množství větším než 0 se tyto údaje nemění.

ImportPartnerPointType

Struktura obsahující definici místa partnera. V současné době se využívá pouze pro zadání výdejního místa pro zásilky při objednávce na výdej[2].

Jméno	Typ	Význam	Platné hodnoty
type	string	Typ výdejního místa	`null`, `PP`, `PP-BOX` (pro výdejní box) nebo `PP-SHOP` (pro partnera - obchod, trafiku, apod.)
id	***string***	ID výdejního místa	
name	string	Název výdejního místa	`null` nebo string o maximální délce 128 znaků
street	string	Adresa výdejního místa	
city	string	Město výdejního místa	`null` nebo string o maximální délce 80 znaků
zipCode	string	PSČ výdejního místa	Pevná délka je 5 znaků
country	string	Město výdejního místa	`null` nebo string o maximální délce 80 znaků
latitude	decimal	Zeměpisná šířka výdejního místa	`null` nebo číslo platné na 15 desetinných míst. Při implementaci vlastního serializéru / deserializéru je potřeba dodržet specifikace decimal formátu definovaného XSD
longitude	decimal	Zeměpisná délka výdejního místa	
infos	string[]	Další informační položky související s výdejním místem	`null` nebo 2 ≥ položek ≥ 0, maximální délka jedné položky je 32 znaků
note	string	Poznámka	`null` nebo string o maximální délce 32 znaků
idEx	int	Označení exportu pro případné nastavení importního filtru	`null` nebo kladná celočíselná hodnota

ImportPartnerType

Struktura obsahující definici partnera k objednávce na příjem nebo výdej. Pokud se jedná o objednávku na příjem, je partnerem dodavatel. Při objednávce na výdej se jedná o odběratele.

Jméno	Typ	Význam	Platné hodnoty
code[1]	***string***	Kód partnera	Maximální délka je 64 znaků
title	***string***	Název partnera	
name	string	Jméno (zastupující osoby)	`null` nebo string o maximální délce 32 znaků
surname	string	Příjmení (zastupující osoby)	
ic_rc	string	IČO, rodné číslo nebo jiná identifikace	`null` nebo string o délce 8 až 10 znaků
dic	string	DIČ partnera	`null` nebo string o délce 10 znaků
street	**string**	Ulice, ve které sídlí partner	Maximální délka je 64 znaků
city	**string**	Město / Obec, ve které sídlí partner	Maximální délka je 32 znaků
zipCode	**string**	PSČ bez mezery	Pevná délka je 5 znaků
countryCode	**string**	Kód státu	Kód státu dle ISO 3166-1
phone	string	Telefon	`null` nebo string o maximální délce 16 znaků
mobile	**string**	Mobilní telefon	Maximální délka je 16 znaků
email	**string**	Email	Maximální délka je 64 znaků
note	string	Poznámka	`null` nebo string o maximální délce 255 znaků
point	ImportPartnerPointType	Místo partnera	Platné pouze pro objednávku na výdej, kde se jedná o výdejní místo[2]; ve všech ostatních případech má být `null`

[1] Kód je povinný pouze při importu partnera. V ostatních případech je možné nechat nastavené na „“, nebo vyplnit „Neukládá se do seznamu partnerů“.

[2] Původní způsob zadávání výdejního místa přes adresu partnera zůstal zachován. Toto je alternativní, lépe podporovaný, způsob.

Lze zvolit mezi uloženým partnerem a manuálním zadáním informací. V případě, že chcete použít uloženého partnera, vyplňte pouze parametr `code` (ostatní nechte prázdné, tj. použijte prázdný řetězec). V případě, že chcete zadat informace manuálně, vyplňte všechna relevantní pole, minimálně však pole `street` (pole `code` se v takovém případě ignoruje).

ImportDetailsType

Struktura obsahující definici podrobností k objednávce na příjem nebo výdej.

Jméno	Typ	Význam	Platné hodnoty
order	***string***	Objednávka	Jedinečný identifikátor v rámci ukladatele. Maximální délka je 64 znaků
invoice	string	Faktura	`null` nebo string o maximální délce 32 znaků
note	string	Poznámka	`null` nebo string o maximální délce 255 znaků
notePrint	string	Tisknutelná poznámka	`null` nebo string o maximální délce 64 znaků
scannerCheck	bool	Nastaveno na `true`, pokud se má přijímané / vydávané zboží kontrolovat čtečkou	`null` nebo `true`/`false`
carrier	string	Dopravce	Jedna z hodnot viz Dopravce. Lze zvolit pouze dopravce viz Načtení aktuálního seznamu dopravců
carrierName	string	Jméno dopravce	`null` nebo string o maximální délce 64 znaků

ImportDeliveryType

Struktura obsahující podrobnosti k dopravě u objednávek na výdej.

Jméno	Typ	Význam	Platné hodnoty
dispatch	***dateTime***	Datum expedice	Musí být stejné nebo vyšší než datum zaslání dat.
delivery	dateTime	Datum dodání u zákazníka	Může být `null`. Při implementaci vlastního serializéru / deserializéru je potřeba dodržet specifikace dateTime formátu definovaného XSD
leaveCustomsArea	bool	Nastaveno na `true`, pokud zboží opouští celní prostor	`null` nebo `true`/`false`
currency	string	Měna použitá pro insurancePrice a price	Může být `null`. V případě dobírky se jedná o povinný parametr. Formát dle ISO 4217
insurancePrice	float	Cena balíku (pro potřeby pojištění)	Může být `null`, 0, nebo libovolné kladné číslo. Při implementaci vlastního serializéru / deserializéru je potřeba dodržet specifikace decimal formátu definovaného XSD
paymantMethod	string	Platební metoda	Může být `null`. V případě dobírky se jedná o povinný parametr. Platné hodnoty viz definice níže …
price	float	Cena k zaplacení požadovaná při předání dle platební metody	Může být `null`. V případě dobírky se jedná o povinný parametr. Při implementaci vlastního serializéru / deserializéru je potřeba dodržet specifikace decimal formátu definovaného XSD
variableSymbol	string	Variabilní symbol	Může být `null`. V případě dobírky se jedná o povinný parametr. Maximální délka je 16 znaků

Možné platební metody jsou:

`DOB` Platba dobírkou při převzetí

V případě zvolení `DOB` jsou pole `price` a `variableSymbol` povinné.

ImportItemType

Struktura obsahující definici zboží pro objednávky na příjem nebo výdej.

Jméno	Typ	Význam	Platné hodnoty
id	***ImportItemIdType***	Definice zboží	
idDetails	***ImportItemIdDetailsType***	Informace upřesňující zboží	
activities	ImportActivityType[]	VAS prováděné nad zbožím	`null` nebo ≥ 0 položek
count	***float***	Počet kusů (MJ) zboží	Hodnota > 0. Při implementaci vlastního serializéru / deserializéru je potřeba dodržet specifikace decimal formátu definovaného XSD, platnost je na 2 desetinná místa

ImportReceiptType

TOP struktura obsahující informace o jedné objednávce na příjem.

Jméno	Typ	Význam	Platné hodnoty
partner	***ImportPartnerType***	Definice ukladatele	
details	***ImportDetailsType***	Podrobnosti k objednávce	
items	***ImportItemType[]***	Zboží objednávky	Hodnota > 0
activities	ImportActivityType[]	VAS prováděné nad celou objednávkou	`null` nebo ≥ 0 položek

ImportIssueType

TOP struktura obsahující informace o jedné objednávce na výdej.

Jméno	Typ	Význam	Platné hodnoty
partner	***ImportPartnerType***	Definice odběratele	
details	***ImportDetailsType***	Podrobnosti k objednávce	
delivery	***ImportDeliveryType***	Podrobnosti k dopravě	
items	***ImportItemType[]***	Zboží objednávky	Hodnota > 0
activities	ImportActivityType[]	VAS prováděné nad celou objednávkou	`null` nebo ≥ 0 položek

Při použití dopravce `ZAS` se očekává v parametru `note` struktury ImportPartnerType jméno e-shopu (bez www) a to bez ohledu na způsob doručení (na adresu nebo do výdejního místa).

ImportIssueWithFilesType

Struktura agregující informace jedné objednávky na výdej.

Jméno	Typ	Význam	Platné hodnoty
issue	**ImportIssueType**	Definice objednávky	
files	ImportFileType[]	Připojené soubory objednávky	`null` nebo ≥ 0 položek

Doručení do výdejního místa

Od verze API V6.5 lze alternativně pro zadání výdejního místa použít i parametr `point` ve struktuře ImportPartnerType. Tento způsob je preferovaný a universální přes všechny dopravce, u kterých je možné výdejní místo nastavit. Původní způsob zadání je zachovaný.

Dopravce	**Kde získat DATA**	**< API V6.5** Informace očekávané ve struktuře ImportPartnerType	**≥ API V6.5** Informace očekávané ve struktuře ImportPartnerPointType
ZAS	http://www.zasilkovna.cz/api/v4/41494564a70d6de6/branch.xml	`street` = DATA `id` `city` = jméno e-shopu (bez www)	`id` = DATA `id`

CP	http://napostu.cpost.cz/vystupy/balikovny.xml	`street` = „BALÍKOVNA“
		`city` = DATA `NAZEV`
		`zipCode` = DATA `PSC`
			`id` = DATA `PSC`

GLS	https://datarequester.gls-hungary.com/glsconnect/getDropoffPoints.php?ctrcode=CZ	`street` = DATA `Parcel Shop ID`	`id` = DATA `Parcel Shop ID`
DPD	https://pickup.dpd.cz/api/getAll?country=203	NENÍ PODPOROVANÉ	`id` = DATA `id`

ImportFileType

TOP struktura s definicí jednoho souboru.

Jméno	**Typ**	**Význam**	**Platné hodnoty**
type	***string***	Typ souboru	Platné hodnoty viz definice níže …
name	***string***	Jméno souboru včetně přípony	
encodedContent	***string***	Obsah souboru	Kódovaný dle Base64 standardu RFC 4648

Tabulka platných typů souborů a jejich význam:

Typ	**Použití**	**Očekávaný formát souboru**
`Expedition`	Pojí se s objednávkou na výdej. Může obsahovat dokumenty přikládané ke zboží (faktura, apod.)	PDF
`Transport_PDF`	Vlastní přepravní štítek	PDF
`Product_MainImage`	Hlavní obrázek sortimentu	JPG: maximální rozlišení 1440 x 1080 px doporučené rozlišení **480 x 360 px** (velikost kolem 300 kB, tj. u JPG komprese 50%)
`Product_Image`	Obrázek sortimentu	JPG: maximální rozlišení 1440 x 1080 px doporučené rozlišení **160 x 120 px** (velikost kolem 100 kB, tj. u JPG komprese 50%)
`VAS_Receipt`	Soubor VAS služby objednávky a/nebo položky objednávky na příjem	PDF
`VAS_Issue`	Soubor VAS služby objednávky a/nebo položky objednávky na výdej	PDF

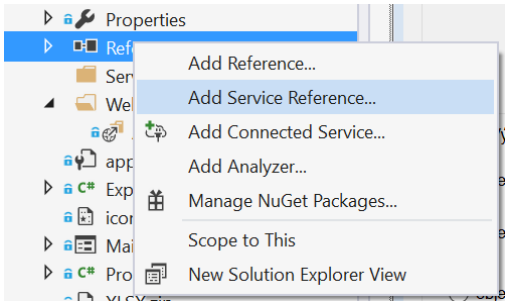
Použití

C#

Přístup k webovým službám popsaných WSDL protokolem SOAP pomocí Microsoft Visual Studio IDE a aplikací psaných v C# je velice snadný. Následující popis je pouze rychlým návodem, jak v Microsoft Visual Studio IDE 2015 k těmto službám přistoupit. Pro více informací je možné čerpat přímo z dokumentace na stránkách Microsoftu:

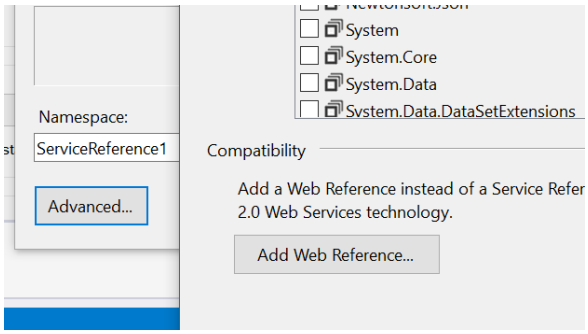
- https://docs.microsoft.com/en-us/dotnet/api/system.web.services.description
- https://docs.microsoft.com/cs-cz/visualstudio/data-tools/windows-communication-foundation-services-and-wcf-data-services-in-visual-studio

1.



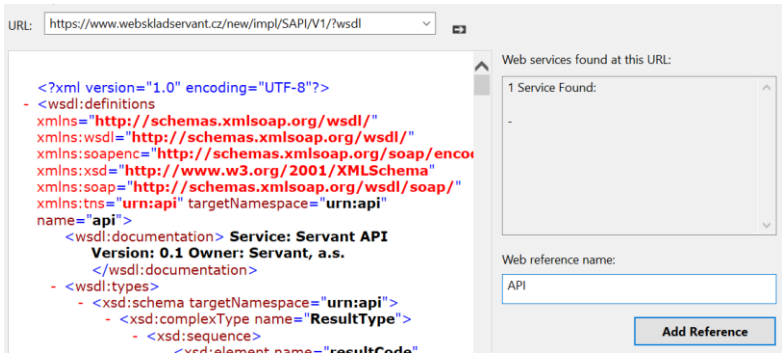
- V projektu aplikace otevřete kontextovou nabídku nad položkou „Reference“ (klikněte pravým tlačítkem)
- Z nabídky vyberte „Add Service Reference“ (nebo totožnou volbu s lokalizovaným textem)

2.



- V levém dolním rohu dialogu, co se zobrazí, vyberte „Advanced…“
- Otevře se další dialog, v jehož opět levém dolním rohu klikněte na „Add Web Reference…“

3.



- Do pole URL zadejte URL webové služby a výběr potvrďte tlačítkem enter
- Pro přístup k webové službě z kódu aplikace zvolte její název (na obrázku zvoleno „API“)
- Přidejte službu stiskem tlačítka „Add Reference“

4.

```
ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12;

API.CanAccessParams CanAccessParams = new API.CanAccessParams();
CanAccessParams.credentials = new API.CredentialsType();
CanAccessParams.credentials.email = "schematic@chematic.cz";
CanAccessParams.credentials.password = "password";
try
{
  API.CanAccessResult CanAccessResult = _API.CanAccess(CanAccessParams);
  _bConnected = CanAccessResult.result.resultCode == 0;
}
catch
{
  _bConnected = false;
}
```

- Před prvním voláním webové služby, v kódu aplikace, nastavte výchozí bezpečnostní protokol na TSL1.2
- Příklad volání služby CanAccess je zobrazen na obrázku

PHP

Přístup k webovým službám popsaných WSDL protokolem SOAP pomocí aplikací psaných v PHP je velice snadný. Následující popis je pouze rychlým návodem, jak v PHP k těmto službám přistoupit. Pro více informací je možné čerpat přímo z dokumentace:

- https://www.php.net/manual/en/book.soap.php
- https://www.php.net/manual/en/class.soapclient.php
- https://www.php.net/manual/en/soapclient.soapclient.php

Vzorová implementace volání služby CanAccess:

```php
<?php

    ini_set('display_errors', 1);
    ini_set('display_startup_errors', 1);
    error_reporting(E_ALL);

    header("Content-type: text/plain");

    try
    {
      $url = 'https://www.webskladservant.cz/impl/SAPI/V6/?wsdl';
      $soap = new SoapClient($url, Array(
          'soap_version' => SOAP_1_2,
          'stream_context' => stream_context_create(Array(
              'ssl' => Array(
                  'crypto_method' => STREAM_CRYPTO_METHOD_TLSv1_2_CLIENT
              ),
          )),
      ));
      print_r($soap->__getFunctions());

      $canAccessParams = Array(
          'credentials' => Array(
              'email' => 'schematic@schematic.cz',
              'password' => 'password'
          )
      );
      $canAccessResponse = $soap->CanAccess($canAccessParams);
      print_r($canAccessResponse);
    }
    catch (SoapFault $exception)
    {
      print_r($exception);
    }

?>
```

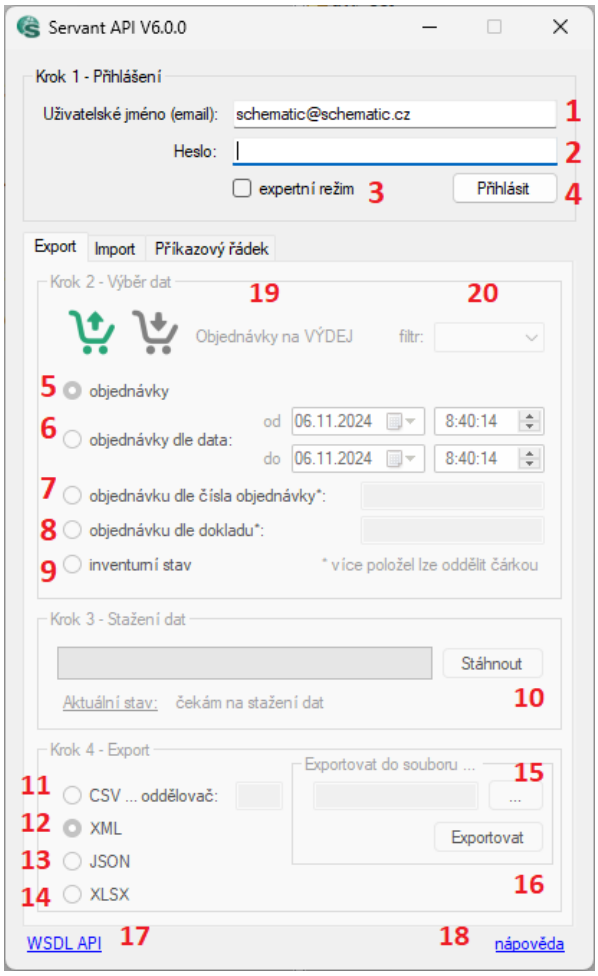
SAPI Aplikace

SAPI je jméno desktopové aplikace určené pro Microsoft Windows. Aplikace slouží zákazníkům společnosti Servant a.s., kteří mají sjednaný přístup do WSS, k ověření přístupu, ke stažení a exportu objednávek a inventurní stavu a k načtení a importu objednávek a zboží. SAPI přistupuje k datům přes webové služby definované rozhraním WSDL na adrese:

https://www.webskladservant.cz/new/impl/SAPI/V6/?wsdl.

Export z WSS

Aplikace umožňuje stažená data exportovat ve formátu XML a JSON.



1. E-mail registrovaný na WSS
2. Hesla přidružené k e-mailu
3. Zapne expertní režim přístupu (pouze pro role Admin a Operator)
4. Akce Přihlásit/Odhlásit
5. Stáhne posledních 1000 objednávek daného ukladatele (seřazeno dle požadované expedice)
6. Stáhne všechny objednávky daného ukladatele, které odpovídají vybranému časovému intervalu (maximálně 1000)
7. Stáhne konkrétní objednávku definovanou číslem objednávky (více objednávek lze oddělit čárkou)
8. Stáhne konkrétní objednávku definovanou dokladem (více objednávek lze oddělit čárkou)
9. Stáhne položky inventurního stavu daného ukladatele (seřazeno dle posledního pohybu)
10. Provede stažení dat/Zruší stažení dat (umožní výběr jiných dat)
11. N/A
12. Exportuje data ve formátu XML (více viz Detaily XML exportu)
13. Exportuje data ve formátu JSON (více viz Detaily JSON exportu)
14. N/A
15. Vybere umístění pro export
16. Provede export (průběh je signalizován procenty)
17. Otevře WSDL rozhraní používané aplikací
18. Otevře tuto nápovědu
19. Přepínač mezi objednávkami na PŘÍJEM a objednávkami na VÝDEJ
20. Nastavení filtru dotazovaných dat

Detaily JSON exportu

JSON formát odpovídá přesně (strukturou, typovostí i pojmenováním jednotlivých datových položek) definici ReceiptOrderType[] pro objednávky na PŘÍJEM, IssueOrderType[] pro objednávky na VÝDEJ a InventoryItemType[] pro položky inventury.

Všechny položky typu „datum“ nebo „datum a čas“ mohou být nastavené nebo nenastavené. Pokud je položka nenastavená, znamená to, že k události ještě nedošlo. Nenastavenou položku lze identifikovat tak, že je hodnota `< 1.1.2000 00:00:00`.

Kódování je UTF-8.

Detaily XML exportu

Pokud zvolíte datový formát XML, jsou data uložena dle struktury zobrazené na následujících stránkách.

Pro objednávky jsou XML elementy:

- `<order>` v rámci nadřazeného elementu `<root>`,
- `<orderItem>` v rámci nadřazeného elementu `<orderItems>`,
- `<attribute>` v rámci nadřazeného elementu `<attributes>`,
- `<activity>` v rámci nadřazeného elementu `<activities>`,
- `<transportItem>` v rámci nadřazeného elementu `<transportItems>`

… v exportu X krát podle počtu objednávek, počtu položek objednávky, počtu atributů položky objednávky (počet atributů je pevně nastaven na 3), počtu aktivit (označováno také jako VAS) objednávky, resp. položky objednávky a počtu položek dopravy objednávky (pouze u objednávek na VÝDEJ).

Pro položky inventury jsou XML elementy:

- `<item>` v rámci nadřazeného elementu `<root>`,
- `<attribute>` v rámci nadřazeného elementu `<attributes>`

… v exportu X krát podle počtu položek inventury a počtu atributů položky inventury (počet atributů je pevně nastaven na 3).

U XML formátu jsou:

- desetinná čísla ukládána ve formátu `0.00`
- položky datumu ukládány ve formátu `yyyy-MM-dd`
- položky datumu a času ukládány ve formátu `yyyy-MM-dd hh:mm:ss`

Kódování je UTF-8.

Struktura exportovaného souboru viz následující příklady:

Položky inventury

```
<?xml version="1.0" encoding="UTF-8" ?>
<root>
  <item>
    <code></code>
    <name></name>
    <attributtes>
      <attributte></attributte>
    </attributtes>
    <expiration></expiration>
    <batch></batch>
    <count></count>
    <requestedCount></requestedCount>
    <unit></unit>
    <group></group>
    <eans>
      <ean>
        <ean></ean>
        <count></count>
        <weight></weight>
        <dimension>
        </dimension>
        <velocity></velocity>
        <primary></primary>
      </ean>
    </eans>
    <checkExpiration></checkExpiration>
    <checkBatch></checkBatch>
    <checkSerialNumbers></checkSerialNumbers>
    <printLabel></printLabel>
    <movement></movement>
  </item>
</root>
```

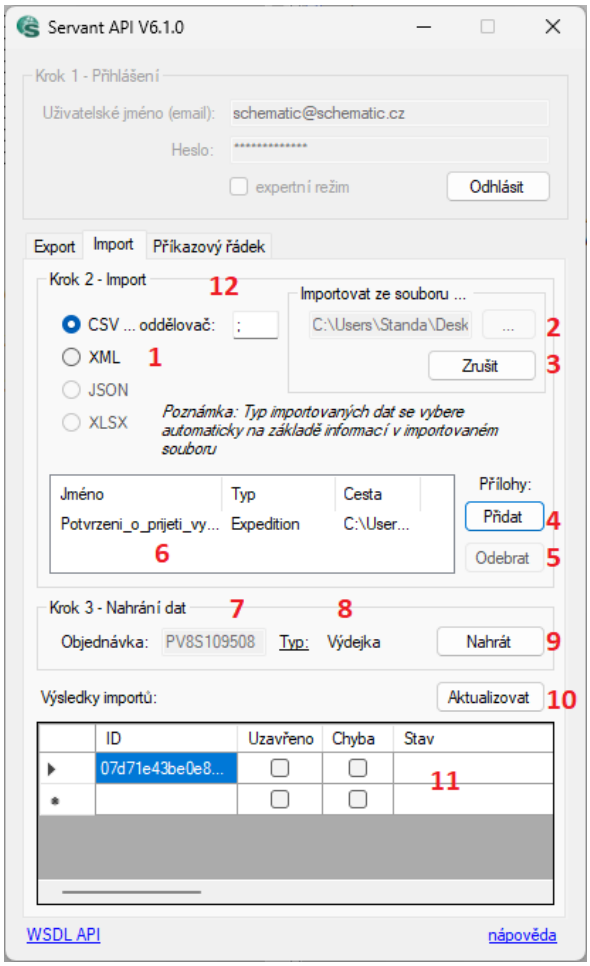
Objednávky na PŘÍJEM

```
<?xml version="1.0" encoding="UTF-8" ?>
<root>
  <order>
    <order></order>
    <document></document>
    <invoice></invoice>
    <activities>
      <activity>
        <id></id>
        <name></name>
        <value></value>
        <count></count>
        <servantNumber></servantNumber>
        <transportNumber></transportNumber>
      </activity>
    </activities>
    <orderItems>
      <orderItem>
        <code></code>
        <name></name>
        <attributtes>
          <attributte></attributte>
        </attributtes>
        <expiration></expiration>
        <batch></batch>
        <requiredCount></requiredCount>
        <count></count>
        <unit></unit>
        <activities>
          <activity>
            <id></id>
            <name></name>
            <value></value>
            <count></count>
            <servantNumber></servantNumber>
            <transportNumber></transportNumber>
          </activity>
        </activities>
      </orderItem>
    </orderItems>
    <partnerName></partnerName>
    <partnerCode></partnerCode>
    <loaded></loaded>
    <received></received>
    <checkState></checkState>
    <finished></finished>
    <error></error>
  </order>
</root>
```

Objednávky na VÝDEJ

```
<?xml version="1.0" encoding="UTF-8" ?>
<root>
  <order>
    <order></order>
    <document></document>
    <invoice></invoice>
    <activities>
      <activity>
        <id></id>
        <name></name>
        <value></value>
        <count></count>
        <servantNumber></servantNumber>
        <transportNumber></transportNumber>
      </activity>
    </activities>
    <orderItems>
      <orderItem>
        <code></code>
        <name></name>
        <attributtes>
          <attributte></attributte>
        </attributtes>
        <expiration></expiration>
        <batch></batch>
        <requiredCount></requiredCount>
        <count></count>
        <unit></unit>
        <activities>
          <activity>
            <id></id>
            <name></name>
            <value></value>
            <count></count>
            <servantNumber></servantNumber>
            <transportNumber></transportNumber>
          </activity>
        </activities>
      </orderItem>
    </orderItems>
    <transportItems>
      <transportItem>
        <unit></unit>
        <servantNumber></servantNumber>
        <transportNumber></transportNumber>
        <statusUrl></statusUrl>
        <packageType></packageType>
        <weight></weight>
        <dimension>
          <x></x>
          <y></y>
          <z></z>
        </dimension>
      </transportItem>
    </transportItems>
    <partnerName></partnerName>
    <partnerCode></partnerCode>
    <recipientAddress></recipientAddress>
    <note></note>
    <state></state>
    <warning></warning>
    <requiredDispatch></requiredDispatch>
    <dispatch></dispatch>
    <loaded></loaded>
    <entered></entered>
    <packaged></packaged>
    <transportCarrier></transportCarrier>
    <transportEntered></transportEntered>
    <transportPicked></transportPicked
    <transportDelivered></transportDelivered>
  </order>
</root>
```

Import na WSS



1. Importuje data ve formátu XML. Vzorové XML pro SAPI aplikaci pro import partnerů, import sortimentu, objednávky na příjem a objednávky na výdej.
2. Tlačítko umožňující výběr dat k importu
3. Tlačítko pro načtení vybraných dat k importu. Pokud jsou již data načtena, je možné načtení zrušit opětovných stiskem tlačítka. Po zrušení načtení je možné vybrat jiný soubor.
4. Přidání souboru odesílaného spolu s importem. V případě, že nejsou načtena žádná data importu, i tak je možné odeslat soubor. V takovém případě je jen potřeba manuálně nastavit objednávku, ke které se soubor přiřadí.
5. Odebrání souboru vybraného v seznamu připojených souborů z výčtu souborů odesílaných spolu s importem.
6. Seznam připojených souborů
7. Zobrazení buď automaticky načtené objednávky, nebo umožnění objednávku zadat manuálně
8. Typ importu, který může nabývat hodnot: **partneři** (proběhne import partnerů), **sortiment** (proběhne import sortimentu), **příjemka** (proběhne import objednávky na příjem), **výdejka** (proběhne import objednávky na výdej) nebo **soubory** (proběhne pouze upload souborů)
9. Provede import na server Servant. Pokud dojde v průběhu k chybě, informace o chybě jsou zobrazené formou dialogu. Pokud vše proběhne správně, je ID transakce importu doplněno za začátek seznamu s výsledky importů.
10. Aktualizuje tabulku s výsledky importů. **POZOR:** Maximální počet výsledků importů získaných v rámci aktualizace může být 100. Pokud bude v tabulce více řádků, dojde k aktualizaci pouze prvních 100 řádků.
11. Seznam importů, u nichž chceme získat jejich výsledky. Záznamy lze ručně editovat (upravovat, mazat staré a přidávat nové)
12. Importuje data ve formátu CSV. Vzorové CSV pro SAPI aplikaci pro objednávky na příjem a objednávky na výdej

Řízení přes příkazovou řádku

Aplikaci je možné spustit bez UI v režimu, ve kterém se veškeré vstupní informace zadají formou argumentu.

Argument	Možné hodnoty a popis	Vyžadováno pro
-modeX	X = UI, Console UI = spuštění aplikace s grafickým rozhraním (výchozí hodnota) Console = spuštění aplikace bez grafického rozhraní	
-actionX	X = CheckCredentials, Export, Import CheckCredentials = kontrola přístupu do WSS Export = export informací z WSS do PC Import = import informací z PC na WSS	-modeConsole
-emailX	X = přihlašovací jméno (email)	-modeConsole
-passwordX	X = přihlašovací heslo	-modeConsole
-userIdX	X = ID uživatele, za kterého se chci přihlásit	
-exportTypeX	X = AllOrders, OrdersByDate, Order, OrderByDocument, Inventory AllOrders = stažení všech objednávek OrdersByDate = stažení všech objednávek v daném časovém intervalu Order = stažení konkrétní objednávky OrderByDocument = stažení konkrétní objednávky specifikované číslem dokladu Inventory = stažení inventury (aktuálního stavu skladu)	-modeConsole **-actionExport**
-filterX	X = MostRelevant, NonZero MostRelevant = viz OrderFilterType NonZero = viz InventoryFilterType	
-ordersTypeX	X = Issue, Receipt Issue = objednávky na VÝDEJ Receipt = objednávky na PŘÍJEM	-modeConsole **-actionExport** -exportTypeAllOrders -exportTypeOrdersByDate -exportTypeOrder -exportTypeOrderByDocument
-ordersFromX	X = ISO 8601 datum a čas pro objednávky „od“	-modeConsole **-actionExport** -exportTypeOrdersByDate
-ordersToX	X = ISO 8601 datum a čas pro objednávky „do“	-modeConsole **-actionExport** -exportTypeOrdersByDate
-exportOrderX	X = číslo objednávky	-modeConsole **-actionExport** -exportTypeOrder
-documentX	X = číslo dokladu objednávky	-modeConsole **-actionExport** -exportTypeOrderByDocument
-exportFormatX	X = XML, JSON Zvolený formát výstupních dat	-modeConsole **-actionExport**
-exportPathX	X = celá cesta k souboru, kam se mají data uložit	-modeConsole **-actionExport**
-importTypeX	X = Import, Status Import = požadavek na import Status = požadavek na získání stavu importu	-modeConsole **-actionImport**

Argument	Možné hodnoty a popis	Vyžadováno pro

`-importFormatX`	X = XML, CSV
	Zvolený formát vstupních dat
		`-modeConsole`
		`-actionImport`
		`-importTypeImport`

`-csvSeparatorX`	X = oddělovač sloupců CSV souboru s daty	`-modeConsole`
		`-actionImport`
		`-importTypeImport`
		`-importFormatCSV`

`-importPathX`	X = celá cesta k souboru k importu [1)]	`-modeConsole`
		`-actionImport`
		`-importTypeImport`

`-attPathX`	X = celá cesta k příloze importu [1) 2) 3)]	`-modeConsole`
		`-actionImport`
		`-importTypeImport`

`-importOrderX`	X = objednávka [2)]	`-modeConsole`
		`-actionImport`
		`-importTypeImport`

`-statusX`	X = ID importu nebo seznam ID importů pro zjištění statusu (jednotlivé ID jsou v případě seznamu od sebe odděleny pomocí lomítka)	`-modeConsole`
		`-actionImport`
		`-importTypeStatus`

1) Pro `-importTypeImport` je vyžadován buď argument `-importPathX` v případě importu, nebo `-attPathX` v případě, že se nahrává pouze příloha k již zadanému importu. Argumenty `-importPathX` a `-attPathX` mohou být rovněž zadány zároveň a to pokud k zadávanému importu chceme rovnou nahrát i přílohu.

2) Pro `-importTypeImport` je vyžadována objednávka ve chvíli, kdy se nahrává pouze příloha (bez vlastního importu) k již zadanému importu. Tedy je nastaven pouze argument `-attPathX`.

3) Připojený soubor je vždy typu `EXPEDITION` viz Tabulka platných typů souborů.

Návratové kódy

Aplikace může skončit s dvěma návratovými kódy:

- 0 = operace skončila úspěšně (pokud má smysl, výstup je v souboru **result.txt***)
- 1 = operace skončila s chybou, která je blíže specifikovaná v **error.txt** souboru

* Některé operace mohou poskytovat datové výstupy, které jsou následně snadno programově zpracovatelné (typicky požadavek na získání stavu importu). V takovém případě se kromě souboru result.txt vytvoří také jeden nebo i více souborů **data_file_X_Y.json** a **data_file_X_Y.xml**, kde X je inkrementální číslo pro identifikátor operace Y.

Příklad: Pokud skončí import `8cb75b0c7a88cb131f39d0d8c4cfa6e2137bef2aab27d5d22947d71099f1f886` s chybou, budou při požadavku na získání stavu tohoto importu vytvořeny soubory:

- *`data_file_0_8cb75b0c7a88cb131f39d0d8c4cfa6e2137bef2aab27d5d22947d71099f1f886.json`,*
- *`data_file_0_8cb75b0c7a88cb131f39d0d8c4cfa6e2137bef2aab27d5d22947d71099f1f886.xml`,*

obsahující programově zpracovatelné informace o chybě.

Příklady

Zkontroluje bez UI, zdali má uživatel muj@email.cz s heslem mojeheslo přístup k serverům Servant a.s.; vrací 0, pokud ano.

```
-modeConsole -actionCheckCredentials -emailmuj@email.cz -passwordmojeheslo
```

Stáhne bez UI kompletní stav inventury uživatele muj@email.cz a uloží ho do souboru C:\inventura.json ve formátu JSON; pokud operace selže, vrací 1 a popis chyby je v souboru error.txt.

```
-modeConsole -actionExport -emailmuj@email.cz –passwordmojeheslo
   -exportTypeInventory -exportFormatJSON -exportPathC:\inventura.json
```

Nahraje bez UI zboží / objednávku uživatele muj@email.cz ze souboru C:\import.xml; pokud operace selže, vrací 1 a popis chyby je v souboru error.txt; v opačném případě se vrací 0 a ID transakce je v souboru result.txt.

```
-modeConsole -actionImport -emailmuj@email.cz –passwordmojeheslo
   -importTypeImport -importPathC:\import.xml
```

Načte bez UI stav importu 8cb75b0c7a88cb131f39d0d8c4cfa6e2137bef2aab27d5d22947d71099f1f886 a f18ada56243048cf44f804a825396e734b6447776cf354fff6d4853920d89920; pokud operace selže, vrací 1 a popis chyby je v souboru error.txt; v opačném případě se vrací 0 a stavy pro jednotlivý ID transakce jsou v souboru result.txt.

```
-modeConsole -actionImport -emailmuj@email.cz –passwordmojeheslo
   -importTypeStatus -status8cb75b0c7a88cb131f39d0d8c4cfa6e2137bef2aab27d5d22947d71
099f1f886/f18ada56243048cf44f804a825396e734b6447776cf354fff6d4853920d89920
```

Externí nástroje a aplikace

Webové služby definované pomocí WSDL a protokol SOAP je standardem a tedy je možné pro přístup využít také nástroje třetích stran.

Z těch webových je možné použít například:

- http://www.soapclient.com/soaptest.html
- https://extendsclass.com/soap-client-online.html

Pro přímé volání z konzole (bash) je možné se inspirovat například:

- https://stackoverflow.com/questions/12222607/how-to-do-a-soap-wsdl-web-services-call-from-the-command-line

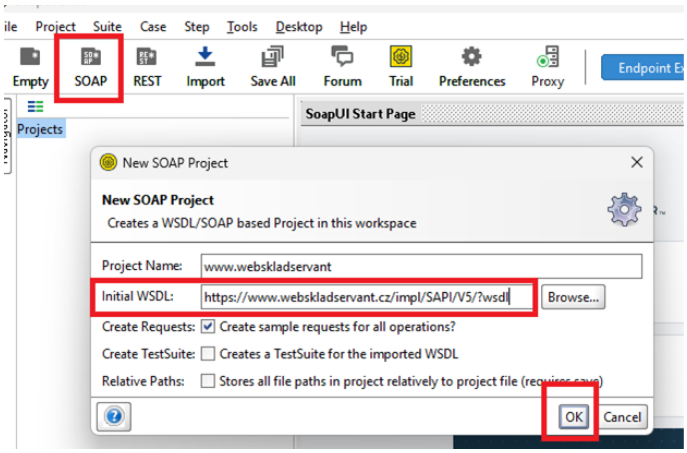
Ověření XML dotazu proti WSDL předpisu

Při sestavování XML dotazu je potřeba akceptovat, že se neřídíme pouze pravidly pro validní XML, ale také pravidly platnými pro SOAP a popis rozhraní webové služby WSDL. Tedy například i změna pořadí atributů v XML, absence nějakého atributu nebo porušení typovosti proti WSDL předpisu, to vše může vést k neplatnosti XML dotazu při validaci proti WSDL předpisu. Různé implementace SOUP/WSDL/XML pak dovolují určité deviace, nicméně dotaz jako takový nemusí být validní.

Ověření je možné provést mnoho nástroji. Zde popsaný příklad využívá open-source nástroje SoupUI. Postup je následující:

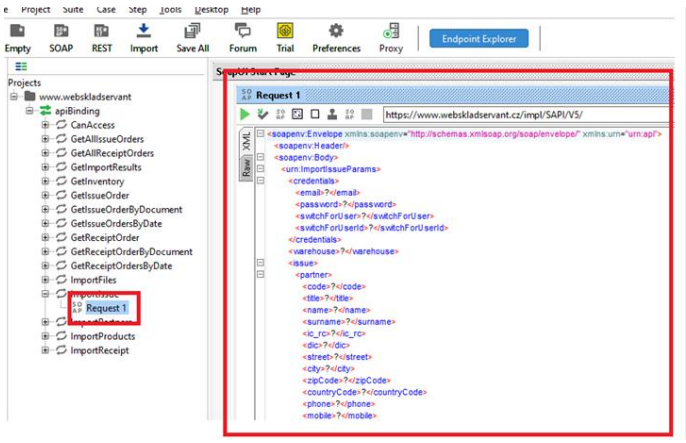
1)

Načíst WSDL API



2)

Otevřít funkci, kterou chceme kontrolovat (nástroj po otevření přímo ukáže předpis, jak má dotaz vypadat)



3)

Nakopírovat testovaný dotaz (ideálně jen obsah mezi elementy <urn:*Params> a </urn:* Params>, pravým tlačítkem vyvolat kontextovu nabídku a dát validovat

