Sauto - dokumentace k importnímu rozhraní

Obsah

1 Úvod

Serverové rozhraní XML-RPC se nachází na adrese https://import.sauto.cz/RPC2 a slouží k importu vozidel na službu Sauto.cz. Pro komunikaci lze použít i adresu http://import.sauto.cz/RPC2, ta ovšem není šifrovaná a používá starší technologie. S adresou http://import.sauto.cz/RPC2 se do budoucna nepočítá a proto doporučujeme používat jen https://import.sauto.cz/RPC2.

Veškerá komunikace se serverem a zpět probíhá v kódování UTF-8. Na adrese http://xmlrpc.com/spec.md se nalézá podrobná specifikace protokolu. Pro optimální rychlost komunikace doporučujeme použít HTTP protokol ve verzi alespoň *HTTP/1.1*, nejlépe však *HTTP/2.0*. Verze HTTP protokolu *HTTP/1.0* je stále funkční, ale její podpora je omezená.

Vytváříte-li nový exportní software, obraťte se na infolinku Sauta (sauto@sauto.cz), kde vám zapůjčí testovací účet. Přes tento účet lze pouze otestovat komunikaci s importním rozhraním. Inzeráty naimportované přes testovací účty se nedostanou do výdeje.

1.1 Co je nového

Úpravy ve verzi 4.0.7:

- Přibyla nová metoda *getReplies()*, které umožňuje získat odpovědi uživatelů na inzeráty.

Úpravy ve verzi 4.0.6:

- Položka karoserie "Valník a sklápěč" v kategoriích užitkové (ID 66) a nákladní (ID 17) byla rozdělena, položka Valník zůstává pod původními ID, nová položka Sklápeč má ID 107 (užitkové) a ID 108 (Nákladní)

Úpravy ve verzi 4.0.5:

- Fotografie - hodnoty řazení *main* větší jak 100 se uloží, ale při práci na importu se s ní bude pracovat jako s neseřazenou fotografii (main=0)

Úpravy ve verzi 4.0.4:

- Nahrazení hodnoty číselníku pohonu osobnních aut z 1="4x2" na 9="Pohon předních kol" nebo 10="Pohon zadních kol"

Úpravy ve verzi 4.0.3:

- *addEditCar()* - zvětšení velikosti položky *note* z 500 na 1000 znaků

Úpravy ve verzi 4.0.2:

- Aktualizace ukázkového příkladu. Ukázkové příklady v jazyce PHP na správu inzerce

Úpravy ve verzi 4.0.1:

- *addEditPhoto()* - při překročení limitu počtu fotografií vrací metoda status=418
- *listOfPhotos()* - do výpisu přidány položky *vin* a *deactivation_reason*
- *getCat()* - do detailu přidán položka *deactivation_reason*
- *addEditCar()* - zrušení recyklace inzerátu podle *custom_id*

Úpravy ve verzi 4.0.0 (změny nasazeny k 14.6.2021):

- Šifrované rozhraní na adrese https://import.sauto.cz/RPC2.
- Přidání parametru *address* do addEditCar(). Inzerátu lze nastavit přesnou adresu, kde je vozidlo k dispozici.
- Nová metoda pro topování inzerátů topCars().

- Nově nelze u vložených inzerátu změnit VIN. Zpětně lze doplnit VIN pouze u nových vozidel, které v okamžiku zaslání neměli VIN v atributech inzerátu nastaven.
- Kategorie *Náhradní díly* byla zrušena s nasazením nového sauta.
- Nová položka pohon - *drive*

Úpravy ve verzi 3.2.9:

- Hodnota stavu vozidla (*condition*) se kontroluje - musí odpovídat určitému druhu vozidla. Tabulka uvedena v popisu funkce addEditCar().

Úpravy ve verzi 3.2.8:

- Nová cesta k uloženému obrázku - attribut filename v metodě listOfPhotos()

Úpravy ve verzi 3.2.7:

- Nové parametry inzerátu pro produktové inzeráty pro Sklik: *client_url*, *custom_label*, *custom_label2*, *custom_label3*, *custom_label4*.
- Zpřesnění dokumentace týkající se pořadí nahrávaných fotografií

Úpravy ve verzi 3.2.6:

- Do parametru *iframe_url* lze vložit šablonovatelnou část.

Úpravy ve verzi 3.2.5:

- Doplnění dokumentace funkce addEditCar(). Úpravy v povinných položkách.
- Oprava kontroly parametrů ve funkci addEditCar()

Úpravy ve verzi 3.2.4:

- Oprava kontroly parametrů ve funkci addEditCar(). Rozšíření používání strukturovaných chyb (*error_items*) a varování (*warning_items*)
- Přepracování českých chybových zpráv

Úpravy ve verzi 3.2.3:

- Automatická korekce karoserií na základě administrátorem definovaných pravidel.
- Ve funkci addEditCar() přibyl fragment *warning_items*, který obsahuje varování.
- Optimalizace nahrávání a mazání fotografií v rámci jednoho inzerátu.

Úpravy ve verzi 3.2.2:

- České chybové zprávy (do budoucna budeme ještě zlepšovat)

Úpravy ve verzi 3.2.1:

- Opravy v dokumentaci (atributy inzerátů, revize status kodu XML-RPC metod, atd.)
- Oprava defaultních návratových hodnot u funkce getCar() - položky datového typu bool, int, codebook a float nejsou navráceny jako prázdný řetězec (""), ale jako prázdná hodnota pro daný typ(0 pro bool, int, codebook a 0.0 pro float).
- Oprava ve funkci getCar() - pri chybe 406 (problém s nevalidním výrobcem, modely a karosérií) se vracel slovník. Nově chodí chybová zpráva

- Ve funkci addEditCar() zlepšena konverze atributů datového typu date
- České chybové zprávy (do budoucna budeme ještě zlepšovat)
- Zrušení anglické verze dokumentace
- Při reimportech inzerátů funkcí addEditCar() se shodnými atributy jaké jsou v databázi se nebude provádět jejich přeukládání do databáze.
- Snížení životnosti *session_id* z 24 hodin na 8 hodiny
- Zlepšení kontroly nahrávání příliš velkých souborů

1.2 Typy dat

Typ	Prázdná hodnota	Popis
int	0	celočíselný datový typ
bool	0	datový typ reprezentující 0 (false) nebo 1 (true)
float	0.0	číslo s plovoucí řádovou čárkou
string	""	řetězec tisknutelných znaků
date	""	datum ve formátu 'rrrr-mm-dd', 'rrrr-mm' nebo 'rrrr' (rrrr = rok, mm = měsíc, dd= den)
codebook	0	odpovídá typu int, jen má pojmenované hodnoty
base64	N/A	base64 encoded data, například obrázek

1.3 Autorizace

Pro přihlášení se nejprve musí zavolat metoda getHash(), která vygeneruje *session_id* a *hash_key*. Hodnota *session_id* se pak používá pro autorizaci všech importních metod, avšak před jejím použitím ji musíme aktivovat pomocí metody login() jež se předají parametry *session_id*, zahashované heslo a softwarový klíč. Hash hesla se vygeneruje následujícím výpočtem:

```
MD5(MD5('heslo')+hash_key)
```

Kde *hash_key* je získán z předchozího volání metody getHash(). Pokud metoda login() vrátí status 200, je *session_id* aktivní a lze ho používat pro autorizované metody. Pro přehlednost postup přihlášení:

```
login = "login"
passwd = "tajneheslo"
swKey = "swklic"

result = xmlRpcClient.getHash(self.login)

sessionId = result["output"]["session_id"]
hashKey = result["output"]["hash_key"]
hash = md5(md5(passwd) + hashKey)

xmlRpcClient.login(sessionId, hash, swKey)
```

Poznámka: *Platnost* session_id *vyprší po 8 hodinách nebo ji lze ihned zrušit zavoláním metody logout().*

2 Metody

2.1 Autorizační metody

2.1.1 struct getHash (string login)

Metoda slouží k získání atributů pro přihlášení. Po zavolání se správným loginem je vrácen *status*=200 a v návratové hodnotě output je *session_id* a *hash_hey*. Parametr *session_id* je použit jako první parametr u všech metod vyžadující autorizaci. Hashovací klíč *hash_hey* je použit pro přihlášení a tím i aktivaci získaného *session_id*. V kapitole Autorizace 1.3 najdete bližší informace k tomu jak funguje proces autorizace.

Parametry:

```
string login                login klienta, který se chce přihlásit
```

Návratová hodnota:

```
struct {
    int status                   status (200=OK,
                                         401=Neexistující klient,
                                         452=Nevalidní parametry,
                                         500=Interní chyba serveru)
    string status_message       slovní popis statusu
    [struct output] {
        [string session_id]     identifikace spojení
        [string hash_key]       hashovací klíč
        [string error]           textový popis chyby. nemusí být vrácen vždy
    }
}
```

2.1.2 struct login (string session_id, string password, string software_key)

Metodě login se předávají parametry *session_id* (z metody getHash()), hashované heslo (viz. kapitola Autorizace 1.3) a softwarový klíč. Hashované heslo se vytvoří pomocí vzorce MD5(MD5('heslo')+*hash_key*), kde *hash_key* se rovněž získá z metody getHash(). Softwarový klíč *software_key* je přidělen administrátorem Sauta zvlášť pro každého klienta. Pokud je vrácen status 200, je *session_id* spojení autorizováno a lze provádět správu inzerátů.

Parametry:

```
string session_id       id session získané z metody getHash()
string password          hashované heslo; MD5(MD5('heslo')+hash_key);
                         hash_key se získá z getHash();
                         heslo je heslo klienta, který se chce přihlásit
string software_key     klíč přidělený klientovi administrátorem Sauta
```

Návratová hodnota:

```
struct {
    int status               status (200=OK,
                                     402=Neexistující klient nebo špatné heslo,
                                     403=Neplatný softwarový klíč,
                                     404=Neplatné session_id,
                                     452=Nevalidní parametry,
                                     500=Interní chyba serveru)
    string status_message   slovní popis statusu.
    [struct output] {
        [string error]       textový popis chyby. nemusí být vrácen vždy
    }
}
```

2.1.3 struct logout (string session_id)

Metoda slouží k odhlášení. Po zavolání této metody a navrácení statusu 210 dojde k okamžitému zneplatnění *session_id*.

Parametry:

```
string session_id   id session získané z metody getHash()
```

Návratová hodnota:

```
struct {
    int status              status (210=odhlášení je OK,
                                    404=Neplatné session_id,
                                    452=Nevalidní parametry,
                                    500=Interní chyba serveru)
    string status_message  slovní popis statusu
    [struct output] {
        [string error]      textový popis chyby. nemusí být vrácen vždy
    }
}
```

2.1.4 struct version ( )

Vrátí verzi xml-rpc serveru.

```
Návratová hodnota:
    struct {
        int status              status (200=OK,
                                        500=Interní chyba serveru)
        string status_message   slovní popis statusu
        struct output {
            string version      řetězec s číslem verze import. rozhraní
        }
    }
```

2.2 Metody pro práci s inzeráty

2.2.1 Atributy inzerátu

Při práci s inzeráty se používají následující atributy:

Název atributu	Datový typ	Popis
address	string	adresa, kde je vozidlo k dispozici. Typicky ve tvaru "město, ulice, číslo popisné". Akceptujeme všechny formáty adresy, které lze dohledat na mapy.cz. Například address="Praha, Radlická 3294/10"
airbag	codebook	počet airbagů (1="1x airbag", 2="2x airbagy", dále viz. číselník)
aircondition	codebook	klimatizace (1="bez klimatizace", 2="manuální klimatizace", dále viz. číselník)
attractive_offer	bool	inzerát je v atraktivní nabídce (*attractive_offer* = 1). Pro zobrazení inzerátu v tomto produktu je nutno mít i aktivní objednávku pro tento produkt.
beds	codebook	počet lůžek (1="jedno lůžko", 2="dvě lůžka", dále viz. číselník)
body_id	codebook	id karosérie (viz číselník)
capacity	codebook	počet míst (1="jedno místo", 2="dvě místa", dále viz. číselník). Pro nastavení počtu sedadel u autobusů použijte atribut *seatplace*
car_id	int	id inzerátu (vozidla) podle číslování Sauta
car_status	int	inzerát je aktivní (*car_status* = 1)
certified_id	int	Ověření vozu. Jen pro klienty zapojených do programu Škoda Plus, Das WeltAuto atd.
client_url	string	klientská URL inzerátu - produktové inzertáty pro Sklik (maximálně 1024 znaků)
color	codebook	barva (1="bílá", 2="žlutá", dále viz. číselník)
color_tone	codebook	odstín barvy (1="světlá", 2="tmavá", dále viz. číselník)
color_type	codebook	typ barvy (1="základní", 2="metalíza", dále viz. číselník)
condition	codebook	stav vozidla (1="nové", 2="ojeté", dále viz číselník)
cr	string	číslo (identifikátor) Cebia reportu (maximálně 40 znaků)
crashed	bool	havarováno (*crashed* = 1)
custom_id	string	id inzerátu (vozidla) podle číslování klienta
custom_label	string	uživatelský štítek - produktové inzeráty pro Sklik (max. 50 znaků)
custom_label2	string	uživatelský štítek 2 - produktové inzeráty pro Sklik (max. 50 znaků)
custom_label3	string	uživatelský štítek 3 - produktové inzeráty pro Sklik (max. 50 znaků)
custom_label4	string	uživatelský štítek 4 - produktové inzeráty pro Sklik (max. 50 znaků)
deactivation_reason	string	důvod nevypublikování inzeratu jež může nabývat hodnot: user_deactivated (zneaktivněno uživatelem), admin_deactivated (zneaktivněno adminem), insufficient_images_count (nedostatečný počet fotek), insufficient_modules_count (nedostatečný počet modulů) a vin_duplication (duplicitni vin - muze byt aktivní jen jeden inzerat s daným VINem)
district	codebook	okres (viz. číselník). Podmínkou pro nastavení daného okresu je mít v dané okresu pobočku
disused_date	string	datum - mimo provoz od - formát "rrrr[-mm[-dd]]"
dph	bool	určuje, zda je cena vozidla s DPH (dph=1) nebo bez DPH (dph=0)
door	codebook	počet dveří (1="jedny dveře", 2="dvoje dveře", dále viz číselník)
drive	codebook	pohon (1="4x2", 2="4x4", dále viz číselník) V číselníku https://www.sauto.cz/import/carList je uvedeno, které položky patří do které kategorie
engine_power	int	výkon motoru v kilowattech
engine_volume	int	obsah motoru v ccm
environmental_tax	bool	zaplacená ekologická daň (*environmental_tax* = 1)
euro	codebook	emisní norma (1="splňuje EURO 1", 2="splňuje EURO 2", dále viz. číselník)
first_owner	codebook	zda se jedná o prvního majitele vozidla (1="ano", 2="ne", dále viz. číselník)
fuel	codebook	druh pohonných hmot (1="benzín", 2="nafta", dále viz číselník)

Název atributu	Datový typ	Popis
gas_mileage	float	průměrná spotřeba (l/100km)
gearbox	codebook	převodovka (1="manuální", 2="poloautomatická", dále viz. číselník)
gearbox_level	codebook	počet převodových stupňů (1="3 a méně stupnů", 2="4 stupňová", dále viz. číselník)
gearbox_auto_type	codebook	druh automatické převodovky (1="automatická", 2="dvojspojková", dále viz. číselník)
guarantee_date	string	datum - záruka do - formát "rrrr[-mm[-dd]]"
handicapped	bool	úprava pro zdravotně postižené (*handicapped* = 1)
iframe_url	string	URL iframe (maximálně 250 znaků). Do url lze vložit šablonu např. "...&utm_source=##TYP##...". Výraz ##TYP## bude nahrazen textem "*doporucena-nabidka*", "*vypis*" nebo "*detail*" podle typu stránky, odkud uživatel přišel na váš web. Výchozí hodnota parametru je text "*(nic)*".
iframe_height	int	výška iframe v pixelech
iframe_small_url	string	URL pro iframe pod fotkou
kind_id	codebook	id druhu vozidla (1="osobní", 3="motorky", dále viz. číselník)
load_capacity	int	nosnost v kg
manufacturer_id	codebook	id výrobce (viz. číselník)
made_date	string	datum - výroby vozidla - formát "rrrr[-mm[-dd]]"
model_id	codebook	id modelu (viz. číselník)
motohodiny	int	motohodiny
note	string	poznámka (maximálně 1000 znaků)
payment	int	pokud je vůz na leasing, je tu výše splátky v Kč
payment_count	int	pokud je vozidlo na leasing, je tu počet splátek
perex	string	krátké informace k vozidlu (maximálně 30 znaků)
price	int	cena vozidla. Pokud je vozidlo na leasing, je tu odstupné
price_leasing	int	cena na leasing
price_notice	string	poznámka k ceně (maximálně 150 znaků)
priority_ordering	int	inzerát je v přednostním řazení (*priority_ordering* = 1). Pro zobrazení inzerátu v tomto produktu je nutno mít i aktivní objednávku pro tento produkt.
run_date	string	datum - odkdy je vozidlo v provozu - formát "rrrr[-mm[-dd]]"
seatplace	codebook	počet sedadel (viz. číselník) - jen pro autobusy (kind_id=6). Pro ostatní kategorie použijte *capacity*
service_book	codebook	zda má vozidlo servisní knížku (1="ano", 2="ne", dále viz. číselník)
sign_note	string	poznámka k ceduli za okno vozu (maximálně 100 znaků)
state_id	codebook	id státu, ve kterém bylo vozidlo koupeno (viz. číselník)
stk_date	string	datum - konce STK - formát "rrrr[-mm[-dd]]"
tachometr	int	stav tachometru (počet najetých km, mil)
tachometr_unit	codebook	jednotka, ve které se udává stav tachometru (1="km", 2="mil", dále viz číselník)
total_views	int	Statistika zobrazení detailu. Attribute je jen pro čtení a jeho hodnota se generuje jednou denně (v noci).
tunning	bool	tunningová úprava (*tunning* = 1)
type_info	string	doplňující informace o modelu (maximálně 30 znaků)
url	string	URL pro vlastí kartu vozu (maximálně 150 znaků)
vat_deductable	bool	určuje, zda je možný odpočet DPH (*vat_deductable* = 1)
video_filename	string	Název souboru nahraného videa. V metodě addEditCar() je pouze pro čtení. Lze nastavit jen v addVideo()
vin	string	vin kód (17 znaků)
weight	int	hmotnost v kg

Poznámka: color, color_tone *a* color_type *lze načíst i z proměnné* color *jež je možné poslat i jako řetězec a importní server se pokusí rozpoznat číselníkové hodnoty. Tato funkcionalita je již zastaralá a do budoucna se s ní nepočítá, proto pokud to jde tuto funkcionalitu doporučujeme nepoužívat a do budoucna předělat na číselníky*

Poznámka: *Všechny atributy lze posílat jako řetězec a nechat konverzi na importním serveru Sauta. Tato funkcionalita se nemusí ve všech případech chovat podle předpokladů, proto doporučujeme posílat atributy datového typu jež je uveden v tabulce výše*

Všechny číselníky jsou uloženy na adrese https://www.sauto.cz/dokumentace-importu

2.2.2 struct addEditCar (string session_id, struct car_data)

Metoda pro vložení či editace inzerátu (vozidla). Pokud bude parametr *car_id* v struktuře *car_data* kladné (nenulové) celé číslo, tak se provede editace inzerátu *car_id*. V opačném případě dojde k v vložení nového inzerátu. Pokud vkládáme nový inzerát a nechceme nějakou položku vyplnit (tj. chceme jí nastavit hodnotu 'Neuvedeno'), tak ji buď do struktury car_data vůbec neuvedeme nebo jí přiřadíme defaultní hodnotu pro daný typ. Při editaci, když nechceme měnit určitou položku, tak ji jednoduše do struktury neuvedeme.

V případě, že se jedná o nové vozidlo (*condition* = 1), tak parametr tachometr (počet najetých km/mil) nesmí být větší než 6000. Pokud se jedná o předváděcí vozidlo (*condition* = 4), tak parametr tachometr (počet najetých km/mil) nesmí být větší než 25000. Nejsou-li tyto podmínky splněny, atribut *condition* bude upraven na ojeté vozidlo (*condition* = 2).

Pro některé datumy existují určitá omezení. Datum "Mimo provoz od" (*disused_date*) musí být datum v minulosti, STK (*stk_date*) může být maximálně 7 let v budoucnosti a datumy roku výroby (*made_date*) a "V provozu od" (*run_date*) mohou být maximálně rok v budoucnu. Pro všechny jmenované atributy platí, že by neměli být menší jak 1. 1. 1900 a větší jak 1. 1. 2100.

V následující tabulce jsou uvedeny povinné položky ze struktury *car_data*.

Název atributu	Povinnost
body_id	vždy povinný
condition	vždy povinný. Tabulka povolených hodnot stavu vozidla v závislosti na druhu vozidla: *condition* | *kind_id* 1 | 1, 3, 4, 5, 6, 7, 9, 10, 11 2 | 1, 3, 4, 5, 6, 7, 9, 10, 11 3 | 1, 3, 4, 5, 6, 7, 9, 10, 11 4 | 1, 3, 4, 7 5 | 1, 3, 4, 5, 6, 7, 9, 10, 11 7 | 12 8 | 12 9 | 12
color	Povinný pro osobní (*kind_id* = 1), užitková (*kind_id* = 4), nákladní (*kind_id* = 5) a obytná (*kind_id* = 9) vozidla, autobusy (*kind_id* = 6) a přívěsy (*kind_id* = 7)
dph	Povinný pro všechny druhy vozů (*kind_id*) vyjma náhradních dílů (*kind_id* = 12) a pracovních strojů (*kind_id* = 10)
engine_volume	Povinný pro osobní (*kind_id* = 1), užitková (*kind_id* = 4) a nákladní (*kind_id* = 5) vozidla, autobusy (*kind_id* = 6), motocykly (*kind_id* = 3) a čtyřkolky (*kind_id* = 11)
fuel	Povinný pro osobní (*kind_id* = 1), užitková (*kind_id* = 4), nákladní (*kind_id* = 5) a obytná (*kind_id* = 9) vozidla a autobusy (*kind_id* = 6)
kind_id	vždy povinný
manufacturer_id	vždy povinný
made_date	Povinný pro všechny druhy vozů (*kind_id*) vyjma náhradních dílů (*kind_id* = 12)
model_id	vždy povinný
price	vždy povinný
state_id	Povinný pro osobní (*kind_id* = 1), užitková (*kind_id* = 4), nákladní (*kind_id* = 5) a obytná (*kind_id* = 9) vozidla, autobusy (*kind_id* = 6) a přívěsy (*kind_id* = 7)
tachometr	Povinný pro osobní (*kind_id* = 1), užitková (*kind_id* = 4) a nákladní (*kind_id* = 5) vozidla a autobusy (*kind_id* = 6). Vyjímkou jsou nová vozidla (*condition* = 1). U nich tento parametr povinný není
tachometr_unit	Povinný pro osobní (*kind_id* = 1), užitková (*kind_id* = 4) a nákladní (*kind_id* = 5) vozidla a autobusy (*kind_id* = 6). Vyjímkou jsou nová vozidla (*condition* = 1). U nich tento parametr povinný není
vin	Povinný pro všechny druhy vozů (*kind_id*) vyjma náhradních dílů (*kind_id* = 12). *vin* dále není povinný pro nové (*condition* = 1) vozy a veterány (*condition* = 5). Uložený vin není možné změnit na jinou hodnotu.

Parametry:

```
string session_id          id session získané z metody getHash()
struct car_data {         data vkládaného inzerátu (jednotlivé položky a jejich hodnoty)
    int car_id                      id inzerátu (vozidla) podle číslování Sauta
    string custom_id                id inzerátu (vozidla) podle číslování klienta
    codebook kind_id                id druhu vozidla
    codebook manufacturer_id        id výrobce
    codebook model_id               id modelu
    codebook body_id                id karosérie
    string vin                       vin kód (17 znaků)
    ...
}
```

Návratová hodnota:

```
struct {
    int status                          status (200=OK,
                                                404=Neplatné session_id,
                                                405=Inzerát neexistuje,
                                                406=Chyba v položkách (inzerátu),
                                                408=Vyčerpán počet modulů,
                                                452=Nevalidní parametry,
                                                500=Interní chyba serveru)
    string status_message              slovní popis statusu
    [struct output] {
        [int car_id]                   id právě vloženého (zeditovaného) inzerátu (podle
                                        číslování Sauta).
        [string error]                 textový popis chyb
        [array error_items] {         podrobný popis chyby. Struktura jejíž klíč je index pořadí.
            "0": struct {
                string item             název položky, ve které byla chyba
                string error_message   slovní popis chyby, jaká v položce nastala
                string type             typ chybové hlášky
            }
            "1": struct {
                ...
            }
            ...
        }
        [array warning_items] {       podrobný popis varování. Struktura jejíž klíč je index pořadí.
            "0": struct {
                string item             název položky, ve které bylo varování
                string warning_message slovní popis varování, které v položce nastalo
                string type             typ varování
            }
            "1": struct {
                ...
            }
            ...
        }
    }
}
```

2.2.3 struct getCar(string session_id, int car_id)

Vrátí informace z databáze pro dané auto. Pro zadané id auta vrátí všechny parametry, které bylo možné zadat pomocí metody addEditCar().

Parametry:

```
string session_id    číslo aktuální relace
int car_id           id auta, které se má vrátit
```

Návratová hodnota:

```
struct {
    int status                      status (200=OK,
                                            404=Neplatné session_id,
                                            405=Inzerát neexistuje,
                                            452=Nevalidní parametry,
                                            500=Interní chyba serveru)
    string status_message          slovní popis statusu
    [struct output] {              vracený záznam, struktura identická jako v metodě addEditCar
        int car_id                     id inzerátu (vozidla) podle číslování Sauta
        string custom_id               id inzerátu (vozidla) podle číslování klienta
        codebook kind_id               id druhu vozidla
        codebook manufacturer_id       id výrobce
        codebook model_id              id modelu
        codebook body_id               id karosérie
        int car_status                 aktivní/neaktivní
        string vin                      vin kód (17 znaků)
        ...
        [string error]              textový popis chyby. nemusí být vrácen vždy
    }
}
```

2.2.4 struct getCarId (string session_id, string custom_id)

Na základě klientského Id inzerátu *custom_id* vrátí interní Id Sauta (*car_id*).

Parametry:

```
string session_id   id session získané z metody getHash()
string custom_id    id inzerátu (vozidla) podle číslování klienta
```

Návratová hodnota:

```
struct {
    int status              status (200=OK,
                                    404=Neplatné session_id,
                                    405=Inzerát neexistuje,
                                    452=Nevalidní parametry,
                                    500=Interní chyba serveru)
    string status_message  slovní popis statusu
    [struct output] {
        [int car_id]       id inzerátu (vozidla) podle číslování Sauta
        [string error]      textový popis chyby. nemusí být vrácen vždy
    }
}
```

2.2.5 struct delCar (string session_id, int car_id)

Podle zadaného *car_id* tato metoda označí daný inzerát jako smazaný.

Parametry:

```
string session_id   id session získané z metody getHash()
int car_id          id inzerátu (vozidla) podle číslování Sauta
```

Návratová hodnota:

```
struct {
    int status              status (200=OK,
                                    404=Neplatné session_id,
                                    405=Inzerát neexistuje,
                                    452=Nevalidní parametry,
                                    500=Interní chyba serveru)
    string status_message  slovní popis statusu
    [struct output] {
        [string error]      textový popis chyby. nemusí být vrácen vždy
    }
}
```

2.2.6 struct listOfCars(string session_id, string imported)

Tato metoda vypíše seznam nesmazaných inzerátů, které má klient na Sauto.cz nahrané. Defaultně vrátí jen inzeráty, které byly založeny přes import. Pokud je třeba vrátit i inzeráty vložené z administračního rozhraní Sauta, je nutné explicitně nastavit parametr *imported*.

Parametry:

```
string session_id  id session získané z metody getHash()
string imported     nepovinný parametr. Pro výpis všech inzerátů,
                    včetně těch zadaných ručně z adminu, je potřeba poslat hodnotu 'all'
```

Návratová hodnota:

```
struct {
    int status                                status (200=OK,
                                                      404=Neplatné session_id,
                                                      452=Nevalidní parametry,
                                                      500=Interní chyba serveru)
    string status_message                   slovní popis statusu
    [struct output] {
        [array list_of_cars] {           inzeráty (struktura jejíž klíč je index pořadí)
            "0": struct {                  záznam jednoho inzerátu (vozidla)
                int car_id                 id inz. (vozidla) podle číslování Sauta
                string custom_id           id inz. (vozidla) podle číslování klienta
                int car_status             stav inzerátu (aktivní/neaktivní)
                str deactivation]_reason   duvod nevypublikovani inzeratu
                                            mozne hodnoty popsany u metody addEditCar()
                int kind_id                id druhu vozidla
                int manufacturer_id        id výrobce vozidla
                int model_id               id modelu vozidla
                str vin                     vin kód (17 znaků)
            }
            "1": struct {
                ...
            }
            ...
        }
        [string error]                      textový popis chyby. nemusí být vrácen vždy
    }
}
```

2.2.7 struct topCars(string session_id, array int car_ids)

Metoda topuje zadané inzeráty. Všechny inzeráty předané v seznamu inzerátů *car_ids* označí jako topované a posune je ve výpisech inzerátů, řazených podle data vložení, nahoru na začátek výpisu. Pro možnost využívat topování je potřeba mít v profilu na Sauto.cz nastavenou peněženku. Po úspěšném topování inzerátů dojde ke stržení částky z peněženky podle platného ceníku.

Parametry:

```
string session_id     id session získané z metody getHash()
array int car_ids (  pole s id inzerátů
    int car_id       id inzerátu (vozidla) podle číslování Sauta
    int car_id
    ...
)
```

Návratová hodnota:

```
struct {
    int status                           status (200=OK,
                                                 404=Neplatné session_id,
                                                 416=Chyba topování - nevalidní inzeráty,
                                                 417=Chyba topování - neúspěšná platba,
                                                 452=Nevalidní parametry,
                                                 500=Interní chyba serveru)
    string status_message               slovní popis statusu
    [struct output] {
        [string error]                   textový popis chyby. nemusí být vrácen vždy
    }
}
```

2.3 Metody pro práci s fotografiemi

K určení pořadí slouží parametr *main* s nímž lze třemi způsoby seřadit fotografie v galerii daného inzerátu.

Nastavení pořadí všech fotografií:

- *main* by měla nabývat hodnot 1 až 50. *main* = 1 znamená, že jde o první (hlavní) fotografii, *main* = 2 znamená, že jde o druhou fotografii v pořadí atd. Hodnoty *main* větší jak 100 se uloží, ale při práci na importu se s ní bude pracovat jako s neseřazenou fotografii (*main* = 0)
- žádná hodnota pořadí by se neměla v galerii daného inzerátu opakovat. Pokud by tato situace nastala, tak by se dané fotografie se stejným pořadím mohly zobrazovat v nahodilém pořadí
- nelze nastavit dvě fotografie jako první
- toto určení pořadí je doporučené

Určení jen hlavní fotografie:

- *main* by měla nabývat hodnoty *main* = 1 (hlavní fotografie) nebo *main* = 0 (fotografie není hlavní)
- nelze nastavit dvě fotografie jako hlavní
- u fotografií u nichž byla nastavena hodnota *main* = 0 se nepočítá s tím, že by měly být zobrazovány v nějakém přesném pořadí. Obvykle jsou tyto fotorafie v pořadí, kdy byly nahrány, ale není to pravidlem (například pokud se fotografie smažou a znovu nahrají, může být pořadí fotografií rozdílné)
- pokud již existuje hlavní fotografie a je nahrána nová fotografie s označením jako hlavní, je označení původní hlavní fotografie zrušeno a hlavní fotografií je fotografie nově označená jako hlavní.
- toto určení pořadí je doporučené jen v případě, kdy záleží jen na určení hlavní fotografie a na pořadí ostatních fotografií nezáleží

Bez určení pořadí:

- *main* není vůbec posláno
- obvykle je první nahraná fotografie označená jako hlavní fotografie (*main* = 1) a všechny ostatní fotografie jsou označené, že nejsou hlavní (*main* = 0). Toto pravidlo nemusí platit vždy (například pokud se fotografie smažou a znovu nahrají)
- na pořadí fotografií neoznačených jako hlavní se v tomto případě nelze nijak spolehnout (obzvláště pokud se fotografie smažou a znovu nahrají)
- toto určení pořadí není doporučené v případě, kdy záleží jakkoliv na pořadí fotografií

2.3.1 struct addEditPhoto(string session_id, int car_id, struct photo_data)

Tato metoda přidá k inzerátu fotografii nebo upraví některé její vlastnosti. Při editaci nelze upravit obrázek, ale jen atributy jako je popisek atd. Pokud bude *photo_id* ve struktuře *photo_data* kladné (nenulové) celé číslo, tak se provede editace této fotografie. V opačném případě dojde k založení nové fotografie. K inzerátu je možno nahrát maximálně 50 fotografií jejichž minimální velikost fotografie 1024x768 pixelů (je nutné respektovat poměr stran - maximální možná odchylka je 1024x550). Velikost nahrávané fotografie nesmí být větší než 5MB a musí být encodovaná v base64 (RFC 4648, Wikipedia). Pro bezproblémové nahrátí fotografií doporučujeme před odesláním zmenšení jejich velikosti za pomoci úpravy počtu pixelů a snížením kvality komprese (na serveru Sauto.cz se aktuálně používá formát JPG fotografie s kvalitou 80 procent).

Pokud vkládáme novou fotografii a nechceme nějakou položku vyplnit, tak ji buď do struktury *photo_data* vůbec neuvedeme nebo jí přiřadíme defaultní hodnotu. Pokud editujeme fotografii a nechceme nějaký atribut měnit, tak ho neuvedeme. Položka b64 je povinná položka pro vkládání, ovšem při editaci je ignorovaná.

Parametry:

```
string session_id          id session získané z metody getHash()
int car_id                 id inzerátu (vozidla) podle číslování Sauta
struct photo_data {       data vkládané fotografie (jednotlivé položky a jejich hodnoty)
    int photo_id               id fotografie podle číslování Sauta
    int main                    uvádí, zda se jedná o hlavní fotku, 2-50 určuje pořadí
    string alt                  popisek k fotce
    string client_photo_id    id fotografie podle číslování klienta
    base64 b64                  fotografie ve formátu JPEG, zakódovaná pomocí base64
}
```

Návratová hodnota:

```
struct {
    int status                          status (200=OK,
                                                404=Neplatné session_id,
                                                405=Inzerát (se zadaným car_id) neexistuje,
                                                406=Chyba v položkách (fotografie))
                                                409=Editovaná fotografie neexistuje,
                                                412=Fotka je chybných rozměrů
                                                418=Byl překročen limit počtu fotografií
                                                452=Nevalidní parametry,
                                                476=Chybný formát fotografie,
                                                500=Interní chyba serveru)

    string status_message              slovní popis statusu
    [struct output] {
        [int photo_id]                 id právě vložené (zeditované) fotografie.
                                        číslování podle Sauta
        [string error]                  textový popis chyby. nemusí být vrácen vždy
        [array error_items] {         struktura jejíž klíč je index pořadí
            "0": struct {              chybová položka
                string item             název položky, ve které byla chyba
                string error_message   slovní popis chyby, jaká v položce nastala
            }
            "1": struct {
                ...
            }
            ...
        }
    }
}
```

2.3.2 struct delPhoto (string session_id, int photo_id)

Smaže fografii podle zadaného *photo_id*. Identifikátor *photo_id* je jedinečné nejen v rámci jednoho inzerátu (vozidla), ale v rámci celého serveru Sauta, proto pro smazání fotografie není potřeba *car_id*.

Parametry:

```
string session_id   id session získané z metody getHash()
int photo_id        id fotografie podle číslování Sauta
```

Návratová hodnota:

```
struct {
    int status              status (200=OK,
                                    404=Neplatné session_id,
                                    409=Fotografie (se zadaným photo_id) neexistuje,
                                    452=Nevalidní parametry,
                                    500=Interní chyba serveru)
    string status_message  slovní popis statusu
    [struct output] {
        [string error]      textový popis chyby. nemusí být vrácen vždy
    }
}
```

2.3.3 struct getPhotoId (string session_id, int car_id, string client_photo_id)

Podle zadaného *client_photo_id* (id fotografie podle klienta) a *car_id* vrátí id fotografie podle číslování serveru Sauta. *client_photo_id* je jedinečné v rámci jednoho inzerátu (vozidla).

Parametry:

```
string session_id        id session získané z metody getHash()
int car_id               id inzerátu (vozidla) podle číslování Sauta
string client_photo_id   id fotografie podle číslování klienta
```

Návratová hodnota:

```
struct {
    int status              status (200=OK,
                                    404=Neplatné session_id,
                                    405=Inzerát neexistuje,
                                    409=Fotografie (se zadaným client_photo_id) neexistuje,
                                    452=Nevalidní parametry,
                                    500=Interní chyba serveru)
    string status_message  slovní popis statusu
    [struct output] {
        [int photo_id]     id fotografie podle číslování Sauta
        [string error]      textový popis chyby. nemusí být vrácen vždy
    }
}
```

2.3.4 struct listOfPhotos(string session_id, int car_id)

Vrátí seznam všech fotografií daného inzerátu. Pokud parametr *car_id* není uveden nebo má defaultní hodnotu(0), tak vrátí seznam všech fotografií od všech inzerátů daného klienta.

Parametry:

```
string session_id   id session získané z metody getHash()
int car_id          nepovinný parametr id inzerátu (vozidla) podle číslování Sauta
```

Návratová hodnota:

```
struct {
    int status                                status (200=OK,
                                                      404=Neplatné session_id,
                                                      405=Inzerát neexistuje,
                                                      452=Nevalidní parametry,
                                                      500=Interní chyba serveru)
    string status_message                   slovní popis statusu
    [struct output] {
        [array list_of_photos] {          pole fotografií (struktura jejíž klíč je index pořadí)
            "0": struct {
                int photo_id                id fotografie podle číslování Sauta
                string alt                  popisek fotografie
                int main                    (1=fotografie je hlavní, jiná čísla pořadí fotografií)
                string client_photo_id     id fotografie podle číslování klienta
                string filename             Cesta k souboru na serveru. Povolené hodnoty parametrů:
                                            - "fl=res,124,89,3" 124x89 pixelů
                                            - "fl=res,136,88,3" 136x88 pixelů
                                            - "fl=res,150,,3" šířka 150 pixelů při zachování poměrů stran
                                            - "fl=res,210,157,3" 210x157 pixelů
                                            - "fl=res,228,,3" šířka 228 pixelů při zachování poměrů stran
                                            - "fl=res,300,,3"
                                            - "res,382,,3|wrm,/watermark/sauto.png,10,10"
                                                šířka 382 pixelů při zachování poměrů stran a s vodoznakem
                                            - "fl=res,500,,3|wrm,/watermark/sauto.png,10,10"
                                                šířka 500 pixelů při zachování poměrů stran a s vodoznakem
                                            - "fl=res,640,,3|wrm,/watermark/sauto.png,10,10"
                                                šířka 640 pixelů při zachování poměrů stran a s vodoznakem
                                            - "fl=res,1024,,3|wrm,/watermark/sauto.png,10,10"
                                                šířka 1024 pixelů při zachování poměrů stran a s vodoznakem
            }
            "1": struct {
                ...
            }
            ...
        }
        [string error]                      textový popis chyby. nemusí být vrácen vždy
    }
}
```

2.4 Metody pro práci s vybavou

2.4.1 struct addEquipment (string session_id, int car_id, array int equipment)

Tato metoda nejprve smaže všechnu výbavu u zadaného vozidla a potom mu nastaví výbavu uvedenou v poli *equipment*. Pokud chceme výbavu u vozidla jen smazat pošleme parametr *equipment* jako prázdné pole. Výbava bude uložena i v případě, že ne všechny položky pole *equipment* budou správně nastaveny podle číselníků.

Veškerou výbavu je možné najít v číselníku https://www.sauto.cz/import/equipmentList.

V číselníku https://www.sauto.cz/import/carList je uvedeno, které položky výbavy patří k jednotlivým druhům vozidel (*kind_id*).

Z důvodu lepší průchodnosti nahrávané výbavy, jsou zrušené položky *54, 55, 118 a 148* ignorovány a zrušená položka *260* je automaticky změněna na hodnotu *274*. Dále pak zrušené položky *1, 2, 9, 10, 19, 20, 48, 100, 101, 102, 109, 110, 111, 112, 113, 114, 115, 116, 117, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 185, 186, 237, 239 a 272* jsou automaticky převedeny na odpovídající atributy inzerátu. Všechny tyto položky však nedoporučujeme používat a je výhodnější používat přímo atributy inzerátu. Používání těchto zrušených položek může mít poměrně zásadní vliv na rychlost nahrávaných dat v importu.

Parametry:

```
string session_id      id session získané z metody getHash()
int car_id             id inzerátu (vozidla) podle číslování Sauta
array int equipment (   pole s id položek výbavy
    int equipment_id   id položky výbavy
    int equipment_id   id položky výbavy
    ...
)
```

Návratová hodnota:

```
struct {
    int status                          status (200=OK,
                                                404=Neplatné session_id,
                                                405=Inzerát neexistuje,
                                                410=Chyba v položkách výbavy,
                                                452=Nevalidní parametry,
                                                500=Interní chyba serveru)
    string status_message              slovní popis statusu
    [struct output] {
        [string error]                  textový popis chyby. nemusí být vrácen vždy
        [array error_equipment] {      podrobný popis chyby (struktura jejíž klíč je index pořadí)
            "0": struct {
                int equipment_id       id položky vybavení, ve které byla chyba
                int error               kód chyby ( 1=Duplicitní položka,
                                                    2=Položka s tímto ID neexistuje,
                                                    3=Položka pro zadaný druh neexistuje,)
                string error_message   slovní popis chyby jaká v položce vybavení nastala
            }
            "1": struct {
                ...
            }
            ...
        }
        [array warning_items] {        podrobný popis varování. Struktura jejíž klíč je index pořadí.
            "0": struct {
                string item             název položky, ve které bylo varování
                string warning_message slovní popis varování, které v položce nastalo
                string type             typ varování
            }
            "1": struct {
                ...
            }
            ...
        }
    }
}
```

Následující tabulka obsahuje výbavu, jež byla zrušena a převedena na atributy inzerátu:

Název výbavy	Id výbavy	Jméno atributu	Hodnota
automatická převodovka	1	gearbox	3
manuální převodovka	2	gearbox	1
poloautomatická převodovka	174	gearbox	2
variátor	48	gearbox	3
manuální klimatizace	9	aircondition	2
automatická klimatizace	10	aircondition	3
dvouzonova automatická klimatizace	239	aircondition	4
airbag	173	airbags	1
1 x airbag	19	airbag	1
2 x airbag	20	airbag	2
4 x airbag	100	airbag	3
6 x airbag	101	airbag	4
7 x airbag	185	airbag	5
8 x airbag	102	airbag	6
9 x airbag	186	airbag	7
10 x airbag	166	airbag	8
10 x airbag	237	airbag	9
12 x airbag	167	airbag	10
2 x dveře	168	door	2
3 x dveře	169	door	3
4 x dveře	170	door	4
5 x dveře	171	door	5
6 x dveře	172	door	6
plní EURO 0	112	euro	(nezadáno)
plní EURO 1	111	euro	1
plní EURO 2	110	euro	2
plní EURO 3	109	euro	3
plní EURO 4	163	euro	4
plní EURO 5	164	euro	5
plní EURO 6	165	euro	6
plní EURO I	175	euro	1
plní EURO II	176	euro	2
plní EURO III	177	euro	3
plní EURO IV	178	euro	4
plní EURO V	179	euro	5
plní EURO VI	180	euro	6
pohon 4 x 2	181	drive	1
pohon 4 x 4	113	drive	2
pohon 6 x 2	182	drive	3
pohon 6 x 4	114	drive	4
pohon 6 x 6	115	drive	5
pohon 8 x 4	116	drive	6
pohon 8 x 6	272	drive	7
pohon 8 x 8	117	drive	8

2.4.2 struct listOfEquipment (string session_id, int car_id)

Metoda pro inzerát zadaný pomocí *car_id* vrátí všechny položky jeho výbavy.

Parametry:

```
string session_id   id session získané z metody getHash()
int car_id          id inzerátu (vozidla) podle číslování Sauta
```

Návratová hodnota:

```
struct {
    int status                          status (200=OK,
                                                404=Neplatné session_id,
                                                405=Inzerát neexistuje,
                                                452=Nevalidní parametry,
                                                500=Interní chyba serveru)
    string status_message               slovní popis statusu
    [struct output] {
        [string error]                   textový popis chyby. nemusí být vrácen vždy
        [array equipment] {             pole s id položek výbavy
            "0": int equipment_id       id položky výbavy
            "1": int equipment_id       id položky výbavy
            ...
        }
    }
}
```

2.5 Metody pro práci s videem

2.5.1 struct addVideo(string session_id, int car_id, struct video_data)

Metoda pro přidání videa k inzerátu. Lze vložit pouze jedno video jehož maximální velikost nesmí přesáhnou velikost 1 GB. Video je rovněž jako fotografie encodované v base64.

Parametry:

```
string session_id      id session získané z metody getHash()
int car_id             id inzerátu (vozidla) podle číslování Sauta
struct video_data {   data vkládaného videa
    string filename     název videa včetně koncovky (např. auto2.avi)
    base64 b64          video zakódované pomocí base64
}
```

Návratová hodnota:

```
struct {
    int status                          status ( 200=OK,
                                                 404=Neplatné session_id,
                                                 405=Inzerát (se zadaným car_id) neexistuje,
                                                 413=Video u inzerátu již existuje,
                                                 414=Video se zpracovává,
                                                 452=Nevalidní parametry,
                                                 500=Interní chyba serveru)
    string status_message             slovní popis statusu
    [struct output] {
        [int car_id]                  id inzerátu, ke kterému bylo přidáno video
        [string error]                 textový popis chyby. nemusí být vrácen vždy
        [array error_items] {        podrobný popisem chyby (struktura jejíž klíč je index pořadí)
            "0": struct {
                string item            název položky, ve které byla chyba
                string error_message  slovní popis chyby, jaká v položce nastala
            }
            "1": struct {
                ...
            }
            ...
        }
    }
}
```

2.5.2 struct delVideo(string session_id, int car_id)

Smaže video u inzerátu podle *car_id*.

Parametry:

```
string session_id      id session získané z metody getHash()
int car_id             id inzerátu podle číslování Sauta
```

Návratová hodnota:

```
struct {
    int status              status (200=OK,
                                    404=Neplatné session_id,
                                    405=Inzerát (se zadaným car_id) neexistuje,
                                    414=Video se zpracovává,
                                    415=Video neexistuje,
                                    452=Nevalidní parametry,
                                    500=Interní chyba serveru)
    string status_message  slovní popis statusu
    [struct output] {
        [string error]      textový popis chyby. nemusí být vrácen vždy
    }
}
```

2.6 Metody pro práci s odpověď'mi na inzerát

2.6.1 struct getReplies (string session_id, struct filters, int offset, int limit)

Metoda vrací odpovědi uživatelů na inzeráty. Odpovědi je možné získat ke konkrétnímu inzerátu(parametr *car_id*) nebo pro celou firmu. Pomocí parametrů *offset* a *limit* je možné stránkovat přes odpovědi. Maximální hodnota parametru *limit* je *100*. Odpovědi jsou řazené od nejnovejších záznamů.

Při filtrování je nutné uvést alespoň jednu z možností, tedy id inzerátu nebo datové rozmezí. Pokud se filtrování provádí dle data, je potřeba uvést datum od i do.

Parametry:

```
string session_id      id session získané z metody getHash()
struct filters {
    int car_id          id inzerátu
    string date_from    datum od(format %Y-%m-%d, například 2021-09-01)
    string date_to      datum do(format %Y-%m-%d, například 2021-09-30))
}
int offset             offset v poli odpovedi
int limit              limit poctu vracenych odpovedi
```

Návratová hodnota:

```
struct {
    int status                  status (200=OK,
                                        404=Neplatné session_id,
                                        405=Inzerát (se zadaným car_id) neexistuje,
                                        452=Nevalidní parametry,
                                        500=Interní chyba serveru)
    string status_message             slovní popis statusu
    [struct output] {
        [array replies] {   pole odpovědí na inzerát
            "0": struct
                int car_id          id inzerátu
                date create_date    datum vytvoření odpovědi
                string message      text odpovědi na inzerat
                string seller_email email obchodníka
                string user_email   email uživatele
                string user_phone   telefon uživatele

            "1": struct {
                ...
            }
            ...
        }
    }
}
```

3 Seznam všech statusů a hlášek

status	slovní popis statusu
200	OK
210	Odhlášení je OK
401	Neexistující klient
402	Neexistující klient nebo špatné heslo
403	Neplatný klíč softwaru
404	Neplatné session_id
405	Inzerát neexistuje
406	Chyba v položkách inzerátu (při práci s inzerátem)
406	client_photo_id není unikátní (při práci s fotografiemi)
409	Fotografie neexistuje
410	Chyba v položkách výbavy
412	Fotografie chybných rozměrů
413	Video u inzerátu již existuje
414	Video se zpracovává
415	Video neexistuje
418	Byl překročen limit počtu fotografií
452	Nevalidní parametry
476	Chybný formát fotografie
500	Chyba serveru

4 Příklady

4.1 Zjištění verze importniho rozhraní - jazyk PHP

Pro komunikaci s rozhraním sauta, lze použít nativních knihoven jako jsou https://www.php.net/manual/en/ref.xmlrpc.php a https://www.php.net/manual/en/ref.curl.php

Další ukázkové příklady lze nalézt na adrese https://www.sauto.cz/documents/examples.zip

```
<?php

function call_xmlrpc($method, $args) {

    $curl_client = curl_init();

    // vytvorime xmlrpc pozadavku z PHP struktur
    $xmlrpc_request = xmlrpc_encode_request($method, $args, array("encoding" => "utf-8"));

    // nastaveni parametru pozadavku
    curl_setopt_array($curl_client, [
        // https adresa na import - zabezpe
        CURLOPT_URL             => "https://import.sauto.cz/RPC2",
        CURLOPT_HEADER          => false,
        CURLOPT_NOBODY          => false,
        CURLOPT_RETURNTRANSFER  => true,
        // verze HTTP by mela byt aslespon verze HTTP/1.1
        CURLOPT_HTTP_VERSION    => CURL_HTTP_VERSION_2_0,
        // kodovani UTF-8
        CURLOPT_ENCODING        => "UTF-8",
        // XMLRPC na zaklade POST pozadavku
        CURLOPT_CUSTOMREQUEST   => 'POST',
        // odesilame zpravy ktere jsou XML majici znacky podle standardu XMLPRC (http://xmlrpc.com/spec.md)
        CURLOPT_HTTPHEADER      => array('Content-Type: text/xml'),
        // telo pozadavku
        CURLOPT_POSTFIELDS      => $xmlrpc_request,
        // timeout na server - 5 000 ms => 5 s
        CURLOPT_TIMEOUT_MS      => 5000
    ]);

    $raw_response = curl_exec($curl_client);

    curl_close($curl_client);

    // dekodujeme z xml (xmlrpc odpovedi) polozky odpovedi do PHP struktur
    $response = xmlrpc_decode($raw_response);

    // zkontrolujeme zda xmlprc volani proselo a v odpovedi je validni odpoved od serveru
    if (is_array($response) && xmlrpc_is_fault($response)) {
        throw new Exception($response['faultString'], $response['faultCode']);
    }

    return $response;
}

$response = call_xmlrpc("version", array());

print("Version {$response['output']['version']} \n");
```