

Sreality - dokumentace k importnímu rozhraní

Obsah

1	Úvod	1
1.1	Co je nového	1
1.2	Názvosloví	4
1.3	Typy dat	4
1.4	Struktura odpovědi serveru	4
1.5	Autorizace	4
1.6	Identifikace inzerátu, obecné vazby	5
1.7	Zadávání lokality	5
1.8	Vkládání videoprohlídek	6
2	Metody	7
2.1	Autorizační metody	7
2.2	Správa inzerátů	8
2.3	Správa fotek u inzerátů	10
2.4	Správa videoprohlídek u inzerátů	11
2.5	Správa makléřů	12
2.6	Správa developerských projektů	14
2.7	Správa fotek u developerských projektů	15
2.8	Správa videoprohlídek u developerských projektů	16
2.9	Čtení statistik	17
2.10	Metoda dostupná bez přihlášení	19
3	Datové struktury a číselníky	20
3.1	Atributy a číselníky inzerátu	20
3.2	Atributy a číselníky developerského projektu	26
3.3	Návratové kódy a chybové hlášky	29
3.4	Stavy inzerátu ve výpise	30
3.5	Podporované formáty videozáznamů	30
4	Příklady	33
4.1	PHP	33
4.2	Python	35
4.3	Příklad XML požadavku	36

1 Úvod

Serverové rozhraní XML-RPC se nachází na adrese <http://import.sreality.cz/RPC2> a slouží k exportu zakázek realitních kanceláří do systému Srealit.

Veškerá komunikace se serverem a zpět probíhá v kódování UTF-8. Na adrese <http://www.xmlrpc.com/spec> se nalézá podrobná specifikace protokolu.

Zbytek dokumentu pojednává o RPC metodách, které importní rozhraní nabízí. Všem uvedeným metodám je nutno předat všechny specifikované parametry. Pokud je uvedeno, že parametr metody je nepovinný, stačí jej vyplnit podle typu prázdnou hodnotou. Počet parametrů musí odpovídat předpisu.

Vytváříte-li nový exportní software, obraťte se na info linku srealit (info@sreality.cz), kde vám založí testovací účet. Potřebujete k tomu pouze email na Seznam.cz, pod kterým bude účet zpřístupněn. Testovací účet vydrží 1 měsíc a je zdarma. Nelze přes něj žádným způsobem inzerovat, slouží pouze k ověření vyvíjeného exportního softwaru.

1.1 Co je nového

Proti verzi 2.1.15:

- Zvýšen limit pro nahrání videa u inzerátu na hodnotu **1024MB**.

Proti verzi 2.1.14:

- Byla přidána možnost přiřadit k inzerátu odkaz na virtuální prohlídku na webu Matterport.com. URL se vkládá pod parametrem *matterport_url*. VR se zobrazí v detailu inzerátu jako další snímek galerie inzerátu. Inzerát bude opatřen štítkem "Virtuální prohlídka" ve výpisu inzerátu. Podrobnější popis naleznete v dokumentaci metody *addAdvert()*.

Proti verzi 2.1.13:

- Byla přidána podpora Produktových inzerátů verze 3.0 (PI3) naší sesterské služby Sklik. U metody *addAdvert()* tedy přibylly datové parametry *advert_custom_label*, *advert_custom_label2*, *advert_custom_label3*, *advert_custom_label4* a *advert_custom_url*. Podrobnější popis naleznete v dokumentaci metody *addAdvert()*.

Proti verzi 2.1.12:

- Nový parametr *project_url* při importu developerských DP obsahuje webovou adresu na projekt na stránkách developera.

Proti verzi 2.1.11:

- Možnost zadat ruský popis inzerátu a poznámku k ceně. (položky *description_ru* a *advert_price_text_note_ru*).
- Doplnění informace o maximálním počtu fotografií
- Oprava maximálního počtu znaků v anotaci developerských projektů (položka *annotation*).
- Nastavení energetického štítku dle vyhlášky č. 78/2013 Sb. (položka *energy_performance_certificate*).

Proti verzi 2.1.10:

- Nově vracíme položku typu *hash_id* v metodě pro výpis inzerátů (*listAdvert*) jako řetězec (string)
- Přidána chybová hláška o detekci příliš velké fotografie při jejím uploadu (metody *addPhoto* a *addProjectPhoto*, chyba číslo 410)
- Přidána chybová hláška o detekci duplicitní fotografie při jejím uploadu (metoda *addPhoto*, chyba číslo 451)
- Přidána kontrola textových položek inzerátu a projektu (metody *addAdvert*, *addProject* chyba číslo 455)
- Přidání položek do číselníku *advert_price_unit*:
 - za m²/den (ID 9),

- za m²/hodinu (ID 10).
- Úprava číselníku advert_subtype:
 - Přejmenování položky 6+kk na "6 a více", položky 6+1, 7+1 a 7+kk sloučeny na "6 a více".
 - Položka Jiný sloučena s Atypický.
 - Přejmenování položky "Pro komerční výstavbu" na Komerční.
 - Přejmenování položky "Pro bydlení" na Bydlení.
 - Přejmenování položky "Zemědělská půda" na Pole.
 - Přejmenování položky "Trvalý travní porost" na Louka.
 - Přejmenování položky "Zemědělské objekty" na Zemědělský.
 - Přejmenování položky Jiný na Ostatní.
 - Přejmenování položky "Činžovní" na "Činžovní dům".
 - Přejmenování položky Vily na Vila.
 - Odstranění položky Dřevostavby (typ dřevostavba je určen číselníkem building_type, konverze inzerátů s podkategorií Dřevostavby na podkategorii Rodinný).
 - Přesun položky Nízkoenergetické do samostatné položky advert_low_energy (konverze inzerátů s podkategorií Nízkoenergetické na podkategorii Rodinný).
 - Přidání položek Sady/vinice, Virtuální kancelář, Vinný sklep, Půdní prostor, Garážové stání, Mobilheim a Ostatní.
- Balkón povinný pouze pro Byty.
- Lodžie povinná pouze pro Byty.
- Bazén povinný pro Domy.
- Terasa povinná jen pro Byty.
- Přejmenování položky číselníku vlastnictví Jiné na Státní/obecní.
- Přejmenování položky "Parkovací stání" na Parkování.
- Přejmenování položky "Bezbariérový přístup" na Bezbariérový.
- Přejmenování položky Zařízeno na Vybavení.
- Pole auction_date_tour a auction_date_tour2 jsou nyní povinné pro Dražby.
- Do číselníku auction_kind přibyly následující položky:
 - Exekutorská dražba (ID 3),
 - Aukce (ID 4),
 - Obchodní veřejná soutěž (ID 5).
- Nově přidaná pole:
 - Velikost (advert_room_count),
 - Poznámka k ceně v Angličtině (advert_price_text_note_en),
 - Náklady na bydlení (cost_of_living),
 - Typ bytu (flat_class),
 - Nízkoenergetické (advert_low_energy).
- Odstraněná pole:
 - Stavba (estate_building_type),
 - Počet objektů (object_count),
 - Počet domů (house_count),

- Počet obchodů (shops),
 - Počet míst (seats),
 - Počet lůžek (beds),
 - Počet bytů (flats),
 - Počet volných bytů (free_flats),
 - Počet kanceláří (offices),
 - Počet volných kanceláří (free_offices),
 - Rozměry - výška (height),
 - Rozměry - délka (length),
 - Rozměry - šířka (width).
- Opravy oproti původním změnám:
 - Odstranění pole equipment.
 - Namísto sjednocení položek Jiný (ID 36) a "Historické objekty" (ID 35) na "Památka/jiné" (ID 36) došlo jen k přejmenování položky "Historické objekty" (ID 35) na "Památka/jiné" a přejmenování položky Jiné (ID 36) na "Ostatní". ID těchto položek tedy zůstávají stejné, jako ve verzi 2.1.9.

Proti verzi 2.1.9:

- Aktualizovány příklady použití importního rozhraní v PHP a Pythonu. Obě varianty nyní obsahují ukázkou přidání inzerátu.

Proti verzi 2.1.8:

- Možnost definovat lokalitu na základě RUIAN kódu (viz. 1.7). Tato změna se dotkla rpc metod addProject, addAdvert, kde byly vstupní parametry rozšířeny o dva nepovinné parametry locality_ruian resp. locality_ruian_level.
- Do návratových kódů přidán kód 415 - Company is not active.
- Přidána metoda *listAllDailyStat()* vracející statistiky všech inzerátů daného klienta za konkrétní den.

Proti verzi 2.1.7:

- Možnost zadat nové nepovinné parametry pro výtah (atribut elevator) a datum prohlídky do (atribut first_tour_date_to). Všechny tyto nové atributy jsou nepovinné a tudíž jejich absence neohrozí export dat do systému Sreality.
- Upraven parametr bezbariérový přístup (atribut easy_access) se zachováním zpětné kompatibility k předešlým stavům. Stav false a true automaticky přetypován na 0 a 1. Přibyla hodnota číselníku 2 jejíž pomocí lze explicitně říci bez bezbariérového přístupu.

Proti verzi 2.1.6:

- Možnost zadat parametry pro energetický štítek náročnosti budovy pomocí 4 nepovinných atributů inzerátu. Atributy jsou: energy_efficiency_rating (číselník třídy A-G), energy_performance_summary (celkový ukazatel náročnosti kWh/m2 za rok), energy_performance_attachment (příloha ve formátu .pdf nebo .jpg), energy_performance_certificate (číselník odkazuje na platnou normu).
- Při editaci hlavních položek (kdy nelze modifikovat povinné položky) se nevrací chybový kód 452, ale nový chybový kód 484, viz kapitola 3.3. Hlavní položky jsou: advert_function (typ inzerátu) a advert_type (kategorie inzerátu).
- Rozšíření výstupu pro metody listSellerStat, listDailyStat, listStat o atribut with_vat, který určuje zda ceny jsou s nebo bez DPH.

1.2 Názvosloví

<i>klient</i>	z pohledu importního rozhraní je klientem pobočka realitní kanceláře
<i>password, heslo na import</i>	heslo potřebné k přístupu, lze nastavit v adminwebu
<i>rkid</i>	importní id, id z realitní kanceláře
<i>seller</i>	makléř RK
<i>session, relace</i>	časově omezený přístup na importní rozhraní iniciováno metodou login, ukončeno metodou logout
<i>session_id</i>	řetězec znaků a čísel, který identifikuje konkrétní importní session
<i>software_key, SW klíč</i>	řetězec, identifikující použitý software. Lze změnit přes obchodního zástupce
<i>topování inzerátu</i>	placené zvýhodnění pozice v hledání na webu sreality

1.3 Typy dat

Typ	Prázdná hodnota	Popis
<i>int</i>	0	celočíslný datový typ
<i>double</i>	0.0	číslo s plovoucí řádovou čárkou
<i>bool</i>	N/A	true nebo false, kompatibilní s int (0,1)
<i>string</i>	""	řetězec tisknutelných znaků
<i>date</i>	00010101T00:00:00+0000	datum
<i>datetime</i>	00010101T00:00:00+0000	datum a čas
<i>codebook</i>	0	odpovídá typu int, jen má pojmenované hodnoty
<i>multiselect</i>	0 nebo ()	pole intů, každý prvek symbolizuje zaškrtnutou položku
<i>base64</i>	N/A	base64 encoded data, například obrázek

Prázdná hodnota je při vkládání nového objektu ekvivalentní zcela chybějící položce. Při editaci pomáhá zrušit vybranou položku, aby se neukazovala v detailu zakázky.

Poznámka: Typy *date* a *datetime* se přes XML-RPC protokol přenáší stejným typem *datetime*, pouze serverová strana ukládá u typu *date* jen část informace.

1.4 Struktura odpovědi serveru

Všechny metody vrací strukturovaný výstup (asociativní pole), který tvoří položky *output* (výstupní datová struktura), *status* (kód chyby), *statusMessage* (chyba jako textový řetězec).

```
struct(
  [status] => 200,
  [statusMessage] => 'OK',
  [output] => array( )
)
```

Obecně platí, že význam návratových kódů je podobný jako u HTTP protokolu, tedy 2xx (kód je v intervalu 200-299) informuje o dobře dopadnuvší operaci. Vyšší kódy signalizují chybu, jejichž seznam je uveden v sekci 3.3 na straně 29.

1.5 Autorizace

Pro přihlášení se nejprve musí zavolat metoda *getHash*, která inicializuje *session_id*. Ta se používá pro autorizaci všech importních metod.

Každá importní metoda s výjimkou metody *getHash* a *version* mají jako první parametr *session_id*, což je řetězec identifikující relaci, který se s každým autorizovaným přístupem mění!! To znamená, že předaná hodnota *session_id* je už v příštím požadavku neplatná. Pro jakýkoliv nový dotaz na importní rozhraní je tedy nutné vypočítat novou hodnotu *session_id* z naposledy platné hodnoty.

Poznámka: Relaci vyprší platnost, pokud se více jak 15 minut nepošle žádný autorizovaný požadavek.

Session_id se skládá z fixní a variabilní části. Fixní se vytvoří při volání metody *getHash* a zůstává stejná po celý čas relace. Jedná se o prvních 48 znaků *sessionId*. Variabilní část je vypočtena pomocí funkce MD5 běžně používanou na hashování dat, do které vstupuje současně *session_id* a heslo.

Postup výpočtu nového *session_id*:

```
fixedPart = session_id[0:48]
varPart = md5(session_id + md5(password) + software_key)
session_id = fixedPart + varPart
```

1.6 Identifikace inzerátu, obecné vazby

Pro identifikaci slouží dva druhy id, dají se odlišit pomocí přípony:

- **_id** - interní id v databázi Seznamu - unikátní v celém univerzu
- **_rkid** - id realitní kanceláře (RK), pod kterým jej zná klient - unikátní v rámci jednoho klienta, tedy jedné pobočky RK

Takto se identifikují makléři, inzeráty, dev. projekty a fotografie.

Použití *advert_rkid* u metody *addAdvert* říká, že edituji inzerát, který odpovídá danému id. Pokud uvádím *advert_rkid* u metody *addPhoto*, říkám, že fotografii chci svázat s příslušným inzerátem.

Rozdíl v chování *_id* proti *_rkid* je patrný hlavně při přidávání. Příklady dvou druhů adresací:

- *advert_rkid* předávám vždy naplněné. Pokud pod takovým id inzerát již existuje, jedná se o editaci, pokud ne, vytvoří se nový a přidělí se mu předané id.
- *advert_id* posílám prázdné, pokud chci inzerát přidat a naplněné interním id inzerátu v případě editace.

1.7 Zadávaní lokality

Klasické zadání adresy znamená, co nejpřesněji uvést všechny položky týkající se adresy (*locality_city* - město, *locality_street* - ulice, *locality_co* - číslo orientační, *locality_cp* - číslo popisné). Adresa je ověřena proti databázi adres serveru mapy.cz a pokud není jednoznačně zadána, vrací se chyba. Díky tomuto ověření je adresa inzerátu validována a lze podle zadaných údajů spolehlivěji hledat, protože atributy jako kraj a okres, případně jiné nevyplněné atributy, jsou automaticky doplněny na základě informací z map.

Zadání přes UIR-ADR nově i přes RUIAN-ADR celou věc zjednodušuje na pouhé získání detailů o zadaném RUIAN(UIR), které se vepíše do příslušných kolonek. Více informací je uvedeno níže v kapitole 1.7.

Důležité: Oba tyto způsoby vyžadují co nejpřesnější zadání adresy. Ale ne vždy je přesná šipka na mapě žádoucí. Proto lze zadat ještě atribut *locality_inaccuracy_level*, který říká jak moc je dobré znepřesnit adresu pro zobrazení na mapě. Tento parametr má 3 možné hodnoty:

1. adresa je zobrazena přesně dle zadání;
2. adresa je o 1 stupeň znepřesněna, místo přesné adresy se ukazuje ulice, případně obec, podle toho, který nadřazený prvek je logicky nejbližší;
3. adresa je o 2 stupně znepřesněna, místo přesné adresy se ukazuje na mapách část města.

GPS souřadnice Posledním druhem je zadání adresy přes GPS souřadnice (*locality_longitude*, *locality_latitude*). Používá se v situacích, kdy není ještě oblast zaměřena v mapách. Zadává se jako zeměpisná šířka a výška ve stupních. Minuty a vteřiny jsou vyjádřeny desetinnou částí. Předpokládá se vždy šířka severní a délka východní. Nevýhoda tohoto zadání spočívá v tom, že inzerát je svázán pouze s okresem a krajem, ale nikoliv s městem či ulicí, která je nejbližší. Inzerát je spolehlivě dohledatelný pouze pomocí funkce hledat v okolí.

Zahraniční nemovitosti Importovat lze i zahraniční nemovitosti. V takovém případě je vhodné posílat s městem nejlépe i zemi, ve které se nachází. Jelikož se lokalita validuje proti databázi mapy.cz, jsme omezeni jen na Evropu.

Adresa UIR-ADR a RUIAN-ADR Adresy je nutno specifikovat na město (případně městskou část), byty až na číslo popisné nebo číslo orientační.

Ministerstvo práce a sociálních věcí ve spolupráci s obecními úřady udržuje registr adres všech stavebních objektů, které mají číslo domovní. Česká pošta poskytuje pro adresy platná poštovní směrovací čísla. Ministerstvo práce a sociálních věcí dává tento registr k dispozici veřejnosti. Kromě zpřístupnění dat registru na www stránkách MPSV (<http://forms.mpsv.cz/uir/>)

je možno získat zdarma CD-ROM s daty a programy pro prohlížení a periodickou aktualizaci těchto dat. Informace o RUIAN kódu je možné získat na [www](http://www.ruian.cz) adrese (<http://www.ruian.cz>). Při použití RUIAN(UIR) kódu není nutno vyplňovat adresu, kromě případu, kdy RUIAN(UIR) kód není dostatečně podrobný, pak je možno adresu upřesnit textově. (Příklad: K bytu máme k dispozici RUIAN(UIR) kód na úrovni 5, je tedy potřeba doplnit položky *locality_street*, *locality_cp*. V případě, že bychom doplnili i *locality_city*, nebude bráno v potaz a naopak bude přepsáno hodnotou z RUIAN(UIR)).

Popis UIR se předává pomocí parametrů *locality_uir* (vlastní uir) a *locality_uir_level* (přesnost, typ). Popis RUIAN se předává pomocí parametrů *locality_ruian* (vlastní ruian) a *locality_ruian_level* (přesnost, typ). Odpovídající hodnoty parametrů *locality_ruian_level* resp. *locality_uir_level* naleznete v tabulce níže.

Název (<i>locality_uir_level</i> resp. <i>locality_ruian_level</i>)	Popis
Okres (1)	Číselník obsahuje veškeré okresy.
Obec (3)	Číselník obsahuje všechny obce. Z obce lze zjistit k jakému okresu patří.
Část obce (5)	Číselník obsahuje všechny části obce. Některé obce jsou jen částí hlavní obce. Zpětně lze dohledat obci a okres.
Ulice (7)	Číselník obsahuje všechny ulice. Zpětně lze dohledat obci a okres.
Objekt (9)	V číselníku je uvedena část obce, číslo popisné a případná městská část. Zpětně lze dohledat část obce a okres.
Adresa (11)	Z RUIAN-ADR(UIR-ADR) adresy lze kompletně dohledat celou adresu, tj. okres, obec část obce, městskou část, číslo popisné, číslo orientační a PSČ.
Městská část (17)	Obsahuje všechny městské části. Zpětně lze dohledat obec a okres.

Zadání lokality pomocí RUIAN parametrů má vyšší prioritu než-li zadání lokality pomocí UIR parametrů. V případě neúspěchu při získávání lokality pomocí RUIAN parametrů se server pokouší získat lokalitu pomocí UIR parametrů. Naopak v případě úspěchu pro RUIAN vstupní parametry server ignoruje UIR parametry.

1.8 Vkládání videoprohlídek

Sreality.cz podporují vkládání videozáznamů do inzerátů a developerských projektů ve spolupráci se službou Stream.cz. Vždy maximálně jedno video na inzerát/projekt. Každé uploadované video se musí vždy nejdříve zpracovat, což znamená převést do různých kvalit a připravit tak výdej pro přehrávač. Doba zpracování každého videa se mění od tří do deseti minut v závislosti na délce videozáznamu a také na denní době. Než se video překóduje do příslušných kvalit pro naše přehrávače, tak video nelze vidět ani ho modifikovat. Pokud překódování trvá déle jak hodinu, dostává video příznak Timeout a nadále se s ním už na Srealitych nepočítá.

Uploadované video tedy není v detailu inzerátu či developerského projektu vidět ihned po zveřejnění, ačkoliv inzerát už zveřejněný je a všechny fotografie jsou již přístupné. Pro exportní softwary proměnlivá délka zpracování znamená, že se nedozví výsledek operace. 200 OK u *addVideo* znamená, že se povedlo vložit video do fronty požadavků ke zpracování. Nepodporovaný formát případně jinou chybu je možné získat až s časovým odstupem, což není šťastné pro automatizované nástroje.

Doporučený postup je stejný jako při práci s fotografiemi. V rámci aktualizace nabídky nejdříve zjistit již přítomné videoprohlídky (*listVideo*) a následně přidat, resp. odebrat videozáznam podle zjištěných rozdílů (*addVideo*, resp. *delVideo*).

Maximální velikost video souboru je 150MB, podporované formáty lze dohledat na straně 30. Zjednodušeně se dá říct, že se podporují všechny formáty knihovny ffmpeg.

2 Metody

2.1 Autorizační metody

2.1.1 getHash(client_id as int)

Získání řetězce zvaného *sessionId*, který je použit pro hashování hesla.

Metoda slouží k získání řetězce pro hashování hesla. Po zavolání s existujícím *client_id* je vrácen status 200 a v output je *sessionId* a hashovací klíč. Volá se vždy na začátku relace.

```
struct
{
    int status                Status (200=OK
                                402=Neexistující klient
                                )
    string statusMessage      Slovní popis statusu.
    array struct output
    (
        string sessionId      Identifikace spojení
    )
}
```

2.1.2 login(session_id as string)

Potvrzení přihlášení.

Metoda login potvrdí session jako autorizovanou. V kladném případě se ověří, že výpočet nového *session_id* odpovídá očekávání na straně serveru.

Metodě login se předávají parametry *session_id* z metody getHash. Pozor, *session_id* už musí být přepočítáno pomocí vzorečku. Softwarový klíč je přidělen administrátorem sreality zvlášť pro každého klienta (RK) v závislosti na použitém exportním software. Nazpět je vrácen status úspěšnosti.

Poznámka: Bez úspěšně dokončené metody login je celá importní relace ve stavu nepřihlášen a tedy nelze v exportu zakázek pokračovat.

Vrací-li metoda opakovaně 407 Bad session je špatně zadané heslo, importní klíč a nebo *client_id* u klienta nebo na importním serveru. Je třeba zkontrolovat tyto údaje. Heslo na import si nastavuje klient sám, importní klíč lze měnit přes obchodního zástupce, *clientid* bylo firmě přiděleno.

```
struct
{
    int status                Status (200=OK
                                402=Neexistující klient
                                405=Neplatný klíč softwaru
                                407=Neplatné přihlášení
                                )
    string statusMessage      Slovní popis statusu.
    array struct output
    (
    )
}
```

2.1.3 logout(session_id as string)

Odhlášení.

```
struct
{
    int status                Status (200=Odhlášení je OK)
    string statusMessage      Slovní popis statusu.
    array struct output
    (
    )
}
```


2.2 Správa inzerátů

2.2.1 addAdvert(session_id as string, advert_data as struct)

Přidání / editace inzerátu.

Metodě se předává slovník `advert_data`, který v sobě nese veškeré číselníkové i nečíselníkové údaje o inzerátu. Možné atributy inzerátu jsou vypsány v tabulce 1 na straně 20. Pokud chybí povinná položka, `addAdvert` končí s chybou. Pozor, některé položky jsou povinné v závislosti na typu (kategorii) inzerátu.

Pozor, datová struktura `advert_data` nemusí obsahovat všechny předepsané atributy, jsou vyžadovány pouze povinné atributy. Pokud atribut není ve slovníku obsažen, server odpovídající položku nechává nedotčenou. Tedy nemění samovolně její hodnotu.

Metoda vrací status a `advert_id`, které je vhodné si uchovat pro další operace s uloženým inzerátem. Pokud je vyplněn parametr `advert_rkid`, musí být v rámci inzerce konkrétního klienta unikátní. Tento parametr obsahuje vlastní identifikátor inzerátu a tento identifikátor lze v budoucnu použít pro práci s inzerátem přes importní rozhraní, kde plně zastupuje `advert_id`. Editace inzerátu se zajistí vyplněním `advert_id` již uloženého inzerátu, nebo je možno zadat existující `advert_rkid` (viz kapitola 1.6). Adresu inzerátu lze vyplnit více způsoby: klasické zadání, RUIAN(UIR-ADR), GPS souřadnice, viz kapitola 1.7.

Byla přidána podpora Produktových inzerátů verze 3.0 (PI3) naší sesterské služby Sklik. Přibyly tedy nové datové parametry `advert_custom_label`, `advert_custom_label2`, `advert_custom_label3`, `advert_custom_label4` a `advert_custom_url` popsané v tabulce 1 na straně 20. Jedná se o textové uživatelské popisky inzerátu do maximální délky 50 znaků ve formátu lowercase, které slouží k filtrování položek kampaní v Skliku. Položka `advert_custom_url` pak slouží jako klientská URL vedoucí na klientem specifikovanou adresu a její délka je omezena na 1024 znaků. Pozor, všechny řetězce položek typu `advert_custom_` je třeba zadávat malými písmeny, jinak budou stejně na tyto převedeny automaticky.

Poznámka: Pokud zadáte cenu 0 nebo 1, bude místo ceny zobrazeno Info o ceně u RK.

```
struct
{
    int status                Status (200=OK
                                404=Inzerat pro tuto RK neexistuje
                                407=Neplatne prihlaseni
                                452=Nekompletni data k inzeratu
                                455=Nevalidni textová položka inzerátu
                                461=Makler neexistuje
                                484=Hlavni polozky jiz neni mozne modifikovat
                                491=Projekt neexistuje
                                )
    string statusMessage      Slovní popis statusu.
    array struct output
    (
        int advert_id        Cislo inzeratu
    )
}
```

Metoda `addAdvert` při zpracování vstupních parametrů kontroluje maximální počet znaků a detekuje nevhodná slova u některých textových položek. V případě neúspěšné kontroly vrací metoda `addAdvert` status kód 455. V tabulce uvedené níže je výčet textových položek u kterých probíhá kontrola během zpracování inzerátu:

Název textové položky	Max. počet znaků	Detekce nevhodných slov
<code>description</code>	3000	ano
<code>description_en</code>	3000	ano
<code>description_ru</code>	3000	ano

2.2.2 delAdvert(session_id as string, advert_id as int, advert_rkid as string)

Vymazání inzerátu.

Pomocí `advert_id` nebo `advert_rkid` se identifikuje inzerát (viz kapitola 1.6). Uložené fotografie u inzerátu a statistiky budou smazány. Pokud inzerát neexistuje bude vrácen status OK (200).

Poznámka: Inzeráty nejsou smazány ihned po zavolání této metody, aktuálně se drží ještě po dobu 21 dní než se opravdu smažou.

```
struct
{
    int status                Status (200=OK
```

```

                                407=Neplatne prihlaseni
                                452=Nejsou vyplneny vsechny povinne polozky
                                    nebo jsou spatneho typu
                                )
    string statusMessage      Slovní popis statusu.
    array struct output
    (
    )
}

```

2.2.3 listAdvert(session_id as string)

Výpis všech inzerátů přihlášené realitní kanceláře.

V output je vráceno pole, kde každý prvek pole obsahuje strukturu (viz níže). Inzeráty jsou seřazeny podle pořadí vložení.

```

struct
{
    int status                Status (200=OK
                                407=Neplatne prihlaseni
                                )
    string statusMessage      Slovní popis statusu.
    array struct output
    (
        int advert_id         číslo inzerátu
        string advert_rkid     interní číslo realitky
        string advert_url      odkaz na detail inzerátu na server sreality.cz
        int advert_type        druh inzerátu (byty, pozemky...) podle dokumentace
        string hash_id         veřejné číslo inzerátu
        string modified         datum poslední editace. (format datumu: YYYY-MM-DD)
        int published          1=inzerát zveřejněn, 0=inzerát nezveřejněn
        int published_status    status kód - důvod nezveřejnění inzerátu
                                (viz Status kódy inzerátu)
        int top                 inzerát je(1)/není(0) dnes zvýhodněn [0,1]
    )
}

```

2.2.4 topAdvert(session_id as string, advert_id as int, advert_rkid as string)

Nastavit zvýhodnění inzerátu pro aktuální den. Pomocí advert_id nebo advert_rkid se identifikuje inzerát (viz kapitola 1.6). Zvýhodnění inzerátu (tzv. topování) je možnost inzerát umístit do horních pozic ve výpise inzerátů. Topování je zpoplatněno dle aktuálního ceníku Sreality.cz. Inzerát lze "topovat" jen jednou denně nebo dle aktuálních podmínek. Nelze topovat v den vložení inzerátu, ani pokud není inzerát zveřejněn. Také inzerát označený jako duplicitní s nesmazaným originálem nelze zvýhodnit. Nelze zvýhodnit (topovat) nezveřejněný nebo neschválený inzerát."

```

struct
{
    int status                Status (200=OK
                                404=Inzerat nenalezen
                                407=Neplatne prihlaseni
                                452=Nejsou vyplneny vsechny povinne polozky
                                    nebo jsou spatneho typu
                                477=Nelze topovat v den vlozeni
                                478=Nelze zvýhodnit (topovat) inzerát označený jako duplicitní s nesmazaným
                                479=Nelze topovat nezveřejněný inzerát
                                482=Dnes již byl inzerát zvýhodněn maximalní počet krát
                                )
    string statusMessage      Slovní popis statusu.
    array struct output
    (
    )
}

```

2.3 Správa fotek u inzerátů

2.3.1 addPhoto(session_id as string, advert_id as int, advert_rkid as string, data as struct)

Vložení fotografie k již uloženému inzerátu.

Vstupními parametry jsou advert_id nebo advert_rkid a struktura data:

```
struct data
{
    binary data          vlastní obrázek
    int main             1=hlavní fotka, 0=ostatní
    int order            nepovinné pořadí v rámci vedlejších fotek
    string alt           nepovinný údaj, popis obrázku
    int photo_id         nepovinný údaj, interní id fotky
    string photo_rkid    nepovinný údaj, rkid fotky
}
```

Pomocí advert_id nebo advert_rkid se identifikuje inzerát (viz kapitola 1.6). Výstupem je photo_id, které je vhodné si uložit pro mazání fotky. Pokud je vkládána vedlejší fotografie, a přitom není u inzerátu žádná, stává se tato automaticky hlavní fotografií. Pokud je vkládána hlavní fotografie, a přitom u inzerátu již jedna je, stane se vložená fotka fotkou hlavní. Minimální rozlišení fotografie je 480x360 a maximální velikost souboru 5 MB. Pro publikování inzerátu je nutné vložit minimálně 3 fotografie. Maximální počet fotografií je 30. Po překročení této velikosti server vrací chybu a fotografii nepracuje.

Je možné obrázek pouze editovat, což znamená, že se nepřekládá vlastní obrázek, pouze se editují jeho parametry (hlavní, pořadí, popis).

Pro snadnější orientaci se dá vložit photo_rkid, pomocí kterého se později fotka adresuje.

```
struct
{
    int status              Status (200=OK
                           404=Inzerat nebo fotografie nenalezen
                           407=Neplatne prihlaseni
                           410=Obrazek je prilis velky
                           412=Sirka nebo vyska obrazku je prilis mala
                           450=Fotka patri k jinemu inzeratu
                           451=Fotku nelze pridat, jelikoz je duplicitni
                           452=Nejsou vyplneny vsechny povinne polozky
                           nebo jsou spatneho typu
                           476=Neni to JPEG/GIF/PNG obrazek
                           477=Problem s predctenim metadat pro zjistení formátu obrazku
    )
    string statusMessage    Slovní popis statusu.
    array struct output     (
        int photo_id        číslo fotografie
    )
}
```

2.3.2 delPhoto(session_id as string, photo_id as int, photo_rkid as string)

Vymazání fotografie.

Pomocí photo_id nebo photo_rkid se identifikuje fotografie (viz kapitola 1.6). Pokud je mazána hlavní fotografie, hlavní se automaticky stane první vedlejší. Pokud fotografie neexistuje bude vrácen status OK (200).

```
struct
{
    int status              Status (200=OK
                           407=Neplatne prihlaseni
                           452=Nejsou vyplneny vsechny povinne polozky
                           nebo jsou spatneho typu
    )
    string statusMessage    Slovní popis statusu.
    array struct output     (
    )
}
```

2.3.3 listPhoto(session_id as string, advert_id as int, advert_rkid as string)

Výpis fotografie existujícího inzerátu.

Pomocí advert_id nebo advert_rkid se identifikuje makléř (viz kapitola 1.6). V output je vráceno pole, kde každý prvek pole obsahuje strukturu (viz níže). Fotografie jsou seřazeny podle pořadí (atribut order) a v tomto pořadí se ukazují i na webu.

```
struct
{
    int status                Status (200=OK
                                404=Inzerat nenalezen
                                407=Neautorizovany pristup
                                452=Nejsou vyplneny vsechny povinne polozky
                                nebo jsou spatneho typu
                                )
    string statusMessage      Slovní popis statusu.
    array struct output
    (
        int photo_id          interni id fotografie
        string photo_rkid      id fotografie realitky
        int main               hlavni fotografie (1=ano, 0=ne)
        int order              poradí (0=na konci, 1=hlavni, 2..50=poradí)
    )
}
```

2.4 Správa videoprohlídek u inzerátů

2.4.1 addVideo(session_id as string, advert_id as int, advert_rkid as string, data as struct)

Vložení videoprohlídky k již uloženému inzerátu. Opětovné vložení nahrazuje původní obsah.

```
struct data
{
    string video_name          nepovinný údaj, orientační pojmenování videa
    binary video_data          povinná data, vlastní datový záznam
}
```

Vstupními parametry jsou advert_id nebo advert_rkid a struktura video_data obsahuje vlastní video.

Pomocí advert_id nebo advert_rkid se identifikuje inzerát (viz kapitola 1.6). Nepovinný údaj video_name se používá pouze k internímu označení videa. Objevuje se pouze při volání metody *listVideo*, není zveřejněn.

Maximální velikost datového záznamu je 150MB. Po překročení této velikosti server vrací chybu a video nezpracuje. Zpracování je blíže popsáno v kapitole 1.8, podporované formáty lze nalézt na straně 30.

```
struct
{
    int status                Status (200=OK
                                404=Inzerat nenalezen
                                407=Neplatne prihlaseni
                                413=Prilis velky video soubor
                                420=Aktualni video se stale zpracovava
                                452=Nejsou vyplneny vsechny povinne polozky
                                nebo jsou spatneho typu
                                )
    string statusMessage      Slovní popis statusu.
    array struct output       Prázdný výstup
    (
    )
}
```

2.4.2 delVideo(session_id as string, advert_id as int, advert_rkid as string)

Vymazání videoprohlídky.

Pomocí advert_id nebo advert_rkid se identifikuje inzerát (viz kapitola 1.6). Pokud video neexistuje bude vrácen status OK (200).

```

struct
{
    int status                Status (200=OK
                                404=Inzerat nenalezen
                                407=Neplatne prihlaseni
                                420=Video se stale zpracovava
                                452=Nejsou vyplneny vsechny povinne polozky
                                nebo jsou spatneho typu
                                )
    string statusMessage      Slovní popis statusu.
    array struct output
    (
    )
}

```

2.4.3 listVideo(session_id as string, advert_id as int, advert_rkid as string)

Získání informací o videoprohlídkách existujícího inzerátu.

Pomocí advert_id nebo advert_rkid se identifikuje inzerát (viz kapitola 1.6). V output je vráceno pole, které má 0-1 prvků. Prázdné pole se vrací v případě, že inzerát nemá u sebe žádné video.

```

struct
{
    int status                Status (200=OK
                                404=Inzerat nenalezen
                                407=Neautorizovany pristup
                                452=Nejsou vyplneny vsechny povinne polozky
                                nebo jsou spatneho typu
                                )
    string statusMessage      Slovní popis statusu.
    array struct output
    (
        string video_name      symbolické jméno vložené v addVideo
        int video_state         stav zpracování videa
        int video_length        délka zpracovaného videa v sekundách
    )
}

```

Následující tabulka ukazuje stavy atributu video_state:

video_state	Popis
0	Video se zpracovává
1	Video je v pořádku, připraveno k přehrávání v detailu inzerátu na webu
2	Nepodporovaný formát videa, více v tabulce 3 na straně 30
3	Selhalo zpracování videa

2.5 Správa makléřů

2.5.1 addSeller(session_id as string, seller_id as int, seller_rkid as string, client_data as struct)

Přidání nového makléře.

Aby makléř byl vložen, je potřeba vyplnit povinné položky v rámci struktury client_data.

```

struct client_data
{
    string client_login        emailový účet, kterým se dá přihlásit na seznam.cz
    string client_domain       nepovinný údaj, doména pro login,
                                lze ji uvést přímo do client_login
    string client_name          jméno makléře
    string contact_gsm          mobil makléře
    string contact_gsm_code     nepovinný údaj, mezinárodní předvolba makléře
    string contact_phone        nepovinný údaj, telefon
    string contact_phone_code   nepovinný údaj, mezinárodní předvolba
    string contact_email        nepovinný údaj, kontaktní email se může lišit
    string makler_note          nepovinný údaj, poznámka
    string broker_specialization nepovinný údaj, specializace makléře
}

```

```

    binary photo                nepovinný údaj, obrázek makléře
}

```

Pomocí `seller_id` nebo `seller_rkid` se identifikuje makléř (viz kapitola 1.6). Při vkládání inzerátu, pokud má vlastního makléře, je tedy na výběr mezi `seller_rkid` a `seller_id`. Ostatní položky ze struktury `client_data` jako `contact_phone` (telefon), `makler_note` (poznámka makléře), `broker_specialization` (specializace makléře) a `photo` (fotografie makléře) jsou nepovinné. U korektně vloženého makléře je získáno `seller_id` číslo.

```

struct
{
    int status                Status (200=OK
                                407=Neplatne prihlaseni
                                452=Nejsou vyplneny vsechny povinne polozky
                                nebo jsou spatneho typu
                                461=Makler neexistuje
                                462=Login maklere je jiz pouzit
                                476=Neni to JPEG/GIF/PNG obrazek
                                )
    string statusMessage      Slovní popis statusu.
    array struct output
    (
        int seller_id        číslo maklere
    )
}

```

2.5.2 delSeller(session_id as string, seller_id as int, seller_rkid as string)

Odebrání existujícího makléře.

Odstranění proběhne při vyplněním `seller_id` nebo `seller_rkid`. Pomocí `seller_id` nebo `seller_rkid` se identifikuje makléř (viz kapitola 1.6).

```

struct
{
    int status                Status (200=OK
                                407=Neplatne prihlaseni
                                461=Makler nenalezen
                                )
    string statusMessage      Slovní popis statusu.
    array struct output
    (
    )
}

```

2.5.3 listSeller(session_id as string)

Výpis makléřů.

V output je vráceno pole, kde každý prvek pole obsahuje strukturu (viz níže). Makléři jsou seřazeni podle pořadí vložení.

```

struct
{
    int status                Status (200=OK
                                407=Neplatne prihlaseni
                                )
    string statusMessage      Slovní popis statusu.
    array struct output
    (
        int seller_id        číslo maklere
        string seller_rkid    interni číslo rk maklere
        string client_name    jméno a příjmení
        string client_login    login (email) maklere
        int photo             fotografie 1-ano, 0-ne
    )
}

```

2.6 Správa developerských projektů

2.6.1 addProject(session_id as string, project_data as struct)

Přidání nového developerského projektu.

Metodě se předává slovník `project_data`, který v sobě nese veškeré číselníkové i nečíselníkové údaje o projektu. Tyto atributy naleznete v tabulce 2 na straně 26. Pokud chybí povinná položka, `addProject` končí s chybou.

Metoda vrací status a `project_id`, které je dobré si uchovat pro další operace s uloženým projektem. Pokud je vyplněn parametr `project_rkid`, musí být v rámci inzerce konkrétního klienta unikátní. Tento parametr obsahuje vlastní identifikátor projektu a tento identifikátor lze v budoucnu použít pro práci s projektem přes importní rozhraní, kde plně zastupuje `project_id`. Editace projektu se zajistí vyplněním `project_id` již uloženého projektu, nebo je možno zadat existující `project_rkid` (viz kapitola 1.6). Adresu projektu lze vyplnit více způsoby: klasické zadání, RUIAN(UIR-ADR), GPS souřadnice (viz kapitola 1.7).

Poznámka: Atribut `developer_ic` (IČ developerské společnosti), který je povinný a slouží ke svázání projektu s developerem, se kontroluje, zda IČ developera bylo evidováno v systému *sreality*. Registrace developera do systému není přes importní rozhraní možná. Pouze přes obchodního zástupce.

Metoda `addProject` při zpracování vstupních parametrů kontroluje maximální počet znaků a detekuje nevhodná slova u některých textových položek. V případě neúspěšné kontroly vrací metoda `addProject` status kód 455. V tabulce uvedené níže je výčet textových položek u kterých probíhá kontrola během zpracování projektu:

Název textové položky	Max. počet znaků	Detekce nevhodných slov
<code>project_name</code>	200	ano
<code>annotation</code>	250	ano
<code>description</code>	2800	ano
<code>locality_description</code>	1400	ano

```
struct
{
    int status                Status (200=OK
                                404=Projekt nenalezen
                                407=Neplatne prihlaseni
                                452=Nejsou vyplneny vsechny povinne polozky
                                nebo jsou spatneho typu
                                455=Nevalidní textová položka inzerátu
                                471=IC developera nenalezeno
                                )
    string statusMessage      Slovní popis statusu.
    array struct output
    (
        int project_id        Cislo projektu
    )
}
```

2.6.2 delProject(session_id as string, project_id as int, project_rkid as string)

Odebrání existujícího developerského projektu.

Pomocí `project_id` nebo `project_rkid` se identifikuje projekt (viz kapitola 1.6), který má být smazán. Uložené fotografie u projektu budou smazány.

```
struct
{
    int status                Status (200=OK
                                404=Projekt nenalezen
                                407=Neplatne prihlaseni
                                452=Nejsou vyplneny vsechny povinne polozky
                                nebo jsou spatneho typu
                                )
    string statusMessage      Slovní popis statusu.
    array struct output
    (
    )
}
```

2.6.3 listProject(session_id as string)

Výpis developerských projektů.

V output je vráceno pole, kde každý prvek pole obsahuje strukturu (viz níže). Projekty jsou seřazeny podle pořadí vložení.

```
struct
{
    int status                Status (200=OK
                                407=Neplatne prihlaseni
                                )
    string statusMessage      Slovní popis statusu.
    array struct output
    (
        int project_id        číslo projektu
        string project_rkid    interní číslo reality
        string project_name    název projektu
    )
}
```

2.7 Správa fotek u developerských projektů

2.7.1 addProjectPhoto(session_id as string, project_id as int, project_rkid as string, data as struct)

Vložení fotografie k již uloženému developerskému projektu.

Vstupními parametry jsou project_id nebo project_rkid a struktura data.

```
struct data {
    binary data        vlastní obrázek
    int main            1=hlavní fotka, 0=ostatní
    int order           nepovinné pořadí v rámci vedlejších fotek
    string alt          nepovinný údaj, popis obrázku
    int photo_id        nepovinný údaj, interní id fotky
    string photo_rkid   nepovinný údaj, id fotky reality
}
```

Pomocí project_id nebo project_rkid se identifikuje projekt (viz kapitola 1.6). Výstupem je photo_id, které je výhodné si uložit pro mazání fotky. Pokud je vkládána vedlejší fotografie, a přitom není u projektu žádná, stává se tato automaticky hlavní fotografií. Pokud je vkládána hlavní fotografie, a přitom u projektu již jedna je, stane se vložená fotka fotkou hlavní. Minimální rozlišení fotografie je 480x360 a maximální velikost souboru 5 MB. Po překročení této velikosti server vrací chybu a fotografii nezpracuje.

Je možné obrázek pouze editovat, což znamená, že se nepřekládá vlastní obrázek, pouze se editují jeho parametry (hlavní, pořadí, popis).

Pro snadnější orientaci se dá vložit photo_rkid, pomocí kterého se později fotka adresuje.

```
struct
{
    int status                Status (200=OK
                                404=Projekt nebo fotografie neexistuje
                                407=Neplatne prihlaseni
                                410=Obrázek je příliš velký
                                412=Sirka nebo výška obrázku je příliš malá
                                450=Fotka patří k jinému projektu
                                452=Nejsou vyplněny všechny povinné položky
                                nebo jsou špatného typu
                                476=Není to JPEG/GIF/PNG obrázek
                                )
    string statusMessage      Slovní popis statusu.
    array struct output
    (
        int photo_id          číslo fotografie
    )
}
```


2.7.2 delProjectPhoto(session_id as string, photo_id as int, photo_rkid as string)

Vymazání fotografie.

Pomocí photo_id nebo photo_rkid se identifikuje fotografie (viz kapitola 1.6). Pokud je mazána hlavní fotografie, hlavní se automaticky stane první vedlejší. Pokud fotografie neexistuje bude vrácen status OK (200).

```
struct
{
    int status                Status (200=OK
                                407=Neplatne prihlaseni
                                452=Nejsou vyplneny vsechny povinne polozky
                                nebo jsou spatneho typu
                                )
    string statusMessage      Slovní popis statusu.
    array struct output
    (
    )
}
```

2.7.3 listProjectPhoto(session_id as string, project_id as int, project_rkid as string)

Výpis fotografie existujícího developerského projektu.

Pomocí project_id nebo project_rkid se identifikuje projekt (viz kapitola 1.6). V output je vráceno pole, kde každý prvek pole obsahuje strukturu (viz níže). Fotografie jsou seřazeny podle pořadí (atribut order) a v tomto pořadí se ukazují i na webu.

```
struct
{
    int status                Status (200=OK
                                404=Projekt nenalezen
                                407=Neautorizovany pristup
                                452=Nejsou vyplneny vsechny povinne polozky
                                nebo jsou spatneho typu
                                )
    string statusMessage      Slovní popis statusu.
    array struct output
    (
        int photo_id          interni id fotografie
        string photo_rkid     id fotografie reality
        int main               hlavni fotografie (1=ano, 0=ne)
        int order              poradí (0=na konci, 1=hlavni, 2..50=poradí)
    )
}
```

2.8 Správa videoprohlídek u developerských projektů

2.8.1 addProjectVideo(session_id as string, project_id as int, project_rkid as string, data as struct)

Vložení videoprohlídky k již uloženému developerskému projektu. Opětovné vložení nahrazuje původní obsah.

```
struct data
{
    string video_name          nepovinný údaj, orientační pojmenování videa
    binary video_data          povinná data, vlastní datový záznam
}
```

Vstupními parametry jsou project_id nebo project_rkid a struktura video_data obsahuje vlastní video.

Pomocí advert_id nebo advert_rkid se identifikuje projekt (viz kapitola 1.6). Nepovinný údaj video_name se používá pouze k internímu označení videa. Objevuje se pouze při volání metody *listProjectVideo*, není zveřejněn.

Maximální velikost datového záznamu je 150MB. Po překročení této velikosti server vrací chybu a video nezpracuje. Zpracování je blíže popsáno v kapitole 1.8, podporované formáty lze nalézt na straně 30.

```
struct
{
    int status                Status (200=OK
```

```

                                404=Developersky projekt nenalezen
                                407=Neplatne prihlaseni
                                420=Aktualni video se stale zpracovava
                                413=Prilis velky video soubor
                                452=Nejsou vyplneny vsechny povinne polozky
                                    nebo jsou spatneho typu
                                )
    string statusMessage        Slovní popis statusu.
    array struct output         Prazdny vystup
    (
    )
}

```

2.8.2 delProjectVideo(session_id as string, project_id as int, project_rkid as string)

Vymazání videoprohlídky.

Pomocí project_id nebo project_rkid se identifikuje dev. projekt (viz kapitola 1.6). Pokud video neexistuje bude vrácen status OK (200).

```

struct
{
    int status                Status (200=OK
                                404=Developersky projekt nenalezen
                                407=Neplatne prihlaseni
                                420=Video se stale zpracovava
                                452=Nejsou vyplneny vsechny povinne polozky
                                    nebo jsou spatneho typu
                                )
    string statusMessage      Slovní popis statusu.
    array struct output
    (
    )
}

```

2.8.3 listProjectVideo(session_id as string, project_id as int, project_rkid as string)

Získání informací o videoprohlídkách existujícího developerského projektu.

Pomocí project_id nebo project_rkid se identifikuje dev. projekt (viz kapitola 1.6). V output je vráceno pole, které má 0-1 prvků. Prázdné pole se vrací v případě, že projekt nemá u sebe žádné video.

```

struct
{
    int status                Status (200=OK
                                404=Developersky projekt nenalezen
                                407=Neautorizovany pristup
                                452=Nejsou vyplneny vsechny povinne polozky
                                    nebo jsou spatneho typu
                                )
    string statusMessage      Slovní popis statusu.
    array struct output
    (
        string video_name     symbolické jméno vložené v addProjectVideo
        int video_state        stav zpracování videa
        int video_length       délka zpracovaného videa v sekundách
    )
}

```

Stavy atributu video_state lze najít na straně 12.

2.9 Čtení statistik

2.9.1 listStat(session_id as string, advert_id as array, advert_rkid as array)

Výpis statistiky inzerátu / inzerátů.

Výpis je závislý na vyplněných vstupních parametrech. Pro výpis statistiky všech inzerátů, je advert_id i advert_rkid

odesláno jako prázdné pole. Pokud je vyplněno pole `advert_id`, vypíše se jen uvedené inzeráty. Stejně je tomu i u `advert_rkid`. Oba parametry se dají kombinovaně použít, vždy musí oba obsahovat stejný počet prvků. Při vyplnění `advert_id` se ignorují hodnoty v `advert_rkid`, `advert_id` je preferováno (viz kapitola 1.6).

```
struct
{
    int status                Status (200=OK
                                407=Neplatne prihlaseni
                                )
    string statusMessage      Slovní popis statusu.
    array struct output
    (
        int advert_id         číslo inzerátu
        string rkid            číslo inzerátu realitní kanceláře
        int total_views        celková návštěvnost
        double total_price     celkové náklady
        string advert_code     ID zakázky
        double topped_price     strženo za zvýhodnění inzerátu
        double advert_price     strženo za beznou inzerci
        int top                inzerát byl(1)/nebyl(0) zvýhodněn [0,1]
        int with_vat           ceny jsou s DPH (1) nebo bez DPH (0)
    )
}
```

2.9.2 listDailyStat(session_id as string, advert_id as int, advert_rkid as string)

Výpis denní statistiky inzerátu.

Pomocí `advert_id` nebo `advert_rkid` se identifikuje inzerát (viz kapitola 1.6).

```
struct
{
    int status                Status (200=OK
                                407=Neplatne prihlaseni
                                )
    string statusMessage      Slovní popis statusu.
    array struct output
    (
        string date            datum dne
        int views              počet shlednutí
        double price            stržená cena
        double topped_price     strženo za zvýhodnění inzerátu
        int with_vat           cena je s DPH (1) nebo bez DPH (0)
    )
}
```

2.9.3 listAllDailyStat(session_id as string, date as datetime)

Výpis statistiky všech inzerátů daného klienta za konkrétní den určený parametrem *date*.

```
struct
{
    int status                Status (200=OK
                                407=Neplatne prihlaseni
                                )
    string statusMessage      Slovní popis statusu.
    array struct output
    (
        int advert_id         číslo inzerátu
        string rkid            číslo inzerátu realitní kanceláře
        int views              návštěvnost
        double advert_price     strženo za beznou inzerci
        double topped_price     strženo za zvýhodnění inzerátu
        double total_price     celkové náklady
        int with_vat           ceny jsou s DPH (1) nebo bez DPH (0)
    )
}
```

2.9.4 listSellerStat(session_id as string, seller_id as int, seller_rkid as string, from as datetime, till as datetime)

Výpis denních statistik inzerátů jednoho makléře. Čas je zadán uzavřeným intervalem <from, till>.

Pomocí seller_id nebo seller_rkid se identifikuje inzerát (viz kapitola 1.6).

```
struct
{
    int status                Status (200=OK
                             407=Neplatne prihlaseni
                             )
    string statusMessage      Slovní popis statusu.
    array struct output
    (
        string date           den, ke kterému jsou vztazena nasledujici cisla
        int advert_count      pocet inzeratu maklere pro aktualni den
        int views              pocet shlednuti inzeratu maklere
        double advert_price    platba za zverejneni inzeratu
        double topped_price    platba za topovaci operace
        double total_price     soucet za inzerci a topovani pro aktualni den
        int with_vat           ceny jsou s DPH (1) nebo bez DPH (0)
    )
}
```

2.10 Metoda dostupná bez přihlášení

2.10.1 version ()

Výpis verze importu.

```
struct {
    int status                Status (200=OK)
    string statusMessage      Slovní popis statusu.
    array output
    (
        string version
    )
}
```

3 Datové struktury a číselníky

3.1 Atributy a číselníky inzerátu

V následující tabulce jsou jednotlivé atributy řazeny tak, že nahoře jsou vyjmenované povinné položky a dole nepovinné.

Tabulka 1: Seznam atributů inzerátu (metoda addAdvert)

Název	Typ	Popis	Povinný
advert_function	codebook 1 2 3	Typ Prodej Pronájem Dražby	vždy
advert_lifetime	codebook 1 2 3 8 4 6 7	Datum do 7 dní 14 dní 30 dní 45 dní 90 dní 180 dní 360 dní	vždy
advert_price	double	Cena	vždy
advert_price_currency	codebook 1 2 3	Měna Kč USD EUR	vždy
advert_price_unit	codebook 1 2 3 4 5 6 7 8 9 10	Jednotka za nemovitost za měsíc za m ² za m ² /měsíc za m ² /rok za rok za den za hodinu za m ² /den za m ² /hodinu	vždy
advert_type	codebook 1 2 3 4 5	Kategorie Byty Domy Pozemky Komerční Ostatní	vždy
description	string	Popis	vždy
locality_city	string	Město	vždy
locality_inaccuracy_level	int	Úroveň zneřesnění adresy	vždy
advert_id	int	ID interní	pokud zadáno, musí být advert_rkid prázdné
advert_rkid	string	ID pro import	pokud zadáno, musí být advert_id prázdné
advert_room_count	codebook 1 2 3 4 5 6	Velikost 1 pokoj 2 pokoje 3 pokoje 4 pokoje 5 a více pokojů Atypický	povinné pro Domy

Pokračuje na další stránce

Název	Typ	Popis	Povinný
advert_subtype	codebook	Podkategorie	povinné, platí však vždy jen pro jednu kategorii
	2	1+kk	Byty
	3	1+1	Byty
	4	2+kk	Byty
	5	2+1	Byty
	6	3+kk	Byty
	7	3+1	Byty
	8	4+kk	Byty
	9	4+1	Byty
	10	5+kk	Byty
	11	5+1	Byty
	12	6 a více	Byty
	16	Atypický	Byty
	18	Komerční	Pozemky
	19	Bydlení	Pozemky
	20	Pole	Pozemky
	21	Lesy	Pozemky
	22	Louky	Pozemky
	23	Zahrady	Pozemky
	24	Ostatní	Pozemky
	25	Kanceláře	Komerční
	26	Sklady	Komerční
	27	Výroba	Komerční
	28	Obchodní prostory	Komerční
	29	Ubytování	Komerční
	30	Restaurace	Komerční
	31	Zemědělský	Komerční
	32	Ostatní	Komerční
	33	Chata	Domy
	34	Garáž	Ostatní
	35	Památka/jiné	Domy
	36	Ostatní	Ostatní
	37	Rodinný	Domy
	38	Činžovní dům	Komerční
	39	Vila	Domy
	40	Na klíč	Domy
	43	Chalupa	Domy
	44	Zemědělská usedlost	Domy
	46	Rybníky	Pozemky
	47	Pokoje	Byty
	48	Sady/vinice	Pozemky
	49	Virtuální kancelář	Komerční
	50	Vinný sklep	Ostatní
	51	Půdní prostor	Ostatní
	52	Garážové stání	Ostatní
	53	Mobilheim	Ostatní
balcony	bool	Balkón	povinné pro Byty
basin	bool	Bazén	povinné pro Domy
building_condition	codebook	Stav objektu	povinné pro Byty, Domy, Ostatní, Komerční
	1	Velmi dobrý	
	2	Dobrý	
	3	Špatný	
	4	Ve výstavbě	
	5	Projekt	

Pokračuje na další stránce

Název	Typ	Popis	Povinný
	6	Novostavba	
	7	K demolici	
	8	Před rekonstrukcí	
	9	Po rekonstrukci	
building_type	codebook	Stavba	povinné pro Byty, Domy, Ostatní, Komerční
	1	Dřevěná	
	2	Cihlová	
	3	Kamenná	
	4	Montovaná	
	5	Panelová	
	6	Skeletová	
	7	Smíšená	
cellar	bool	Sklep	povinné pro Byty, Domy
estate_area	int	Plocha pozemku	povinné pro Pozemky, Domy
floor_number	int	Patro	povinné pro Byty
garage	bool	Garáž	povinné pro Byty, Domy, Komerční
locality_latitude	double	Zeměpisná šířka	validní pouze s locality_longitude
locality_longitude	double	Zeměpisná délka	validní pouze s locality_latitude
locality_ruian	int	RUIAN lokality	validní pouze s locality_ruian_level
locality_ruian_level	int	RUIAN-level lokality	validní pouze s locality_ruian
locality_uir	int	UIR lokality	validní pouze s locality_uir_level
locality_uir_level	int	UIR-level lokality	validní pouze s locality_uir
loggia	bool	Lodžie	povinné pro Byty
object_type	codebook	Typ domu	povinné pro Domy, Komerční
	1	Přízemní	
	2	Patrový	
ownership	codebook	Vlastnictví	povinné pro Byty
	1	Osobní	
	2	Družstevní	
	3	Státní/obecní	
parking_lots	bool	Parkování	povinné pro Byty, Domy, Komerční
project_id	int	ID projektu u sreality	pokud zadáno, project_rkid je prázdné
project_rkid	string	ID projektu u realitky	pokud zadáno, project_id je prázdné
seller_id	int	ID maklere u sreality	pokud zadáno, seller_rkid je prázdné
seller_rkid	string	ID maklere u realitky	pokud zadáno, seller_id je prázdné
terrace	bool	Terasa	povinné pro Byty
usable_area	int	Užitná plocha	povinné pro Byty, Domy, Ostatní, Komerční
acceptance_year	int	Rok kolaudace	
advert_code	string	ID zakázky	
advert_custom_label	string	Uživatelský štítek/popisek inzerátu	
advert_custom_label2	string	Uživatelský štítek/popisek inzerátu 2	
advert_custom_label3	string	Uživatelský štítek/popisek inzerátu 3	
advert_custom_label4	string	Uživatelský štítek/popisek inzerátu 4	
advert_custom_url	string	Uživatelská URL u inzerátu vedoucí na webovou adresu zadanou klientem	
advert_low_energy	bool	Nízkoenergetický	
advert_price_charge	codebook	poplatky	
	1	včetně poplatků	
	2	bez poplatků	
advert_price_commission	codebook	Provize	
	1	včetně provize	

Pokračuje na další stránce

Název	Typ	Popis	Povinný
	2	+ provize RK	
advert_price_legal_services	codebook	Právní servis 1 včetně právního servisu 2 bez právního servisu	
advert_price_negotiation	bool	Cena k jednání	
advert_price_text_note	string	Poznámka k ceně	
advert_price_text_note_en	string	Poznámka k ceně v Angličtině	
advert_price_text_note_ru	string	Poznámka k ceně v Ruštině	
advert_price_vat	codebook	DPH 1 včetně DPH 2 bez DPH	
annuity	int	Anuita	
auction_advertisement_pdf	base64	Dražební vyhláška v PDF	povinné pro Dražby
auction_date	datetime	Datum konání dražby	povinné pro Dražby
auction_date_tour	datetime	Termín 1. prohlídky	povinné pro Dražby
auction_date_tour2	datetime	Termín 2. prohlídky	povinné pro Dražby
auction_kind	codebook	Druh dražby 1 Nedobrovolná 2 Dobrovolná 3 Exekutorská dražba 4 Aukce 5 Obchodní veřejná soutěž	povinné pro Dražby
auction_place	string	Místo konání dražby	povinné pro Dražby
auction_review_pdf	base64	Posudek znalce v PDF	povinné pro Dražby
balcony_area	int	Plocha balkónu	
basin_area	int	Plocha bazénu	
beginning_date	date	Datum zahájení výstavby	
building_area	int	Plocha zastavěná	
ceiling_height	double	Výška stropu	
cellar_area	int	Plocha sklepu	
cost_of_living	string	Náklady na bydlení	
description_en	string	Anglický popis	
description_ru	string	Ruský popis	
easy_access	codebook	Bezbariérový 1 Ano 2 Ne	
electricity	multiselect	Elektřina 1 120V 2 230V 4 400V	
elevator	codebook	Výtah 1 Ano 2 Ne	
energy_efficiency_rating	codebook	Energetická náročnost budovy 1 A - Mimořádně úsporná 2 B - Velmi úsporná 3 C - Úsporná 4 D - Méně úsporná 5 E - Nehospodárná 6 F - Velmi nehospodárná 7 G - Mimořádně nehospodárná	
energy_performance_attachment	base64	Energetický průkaz v PDF/JPG	
energy_performance_certificate	codebook	podle vyhlášky 1 č. 148/2007 Sb. 2 č. 78/2013 Sb.	

Pokračuje na další stránce

Název	Typ	Popis	Povinný
energy_performance_summary	double	Ukazatel energetické náročnosti budovy	
extra_info	codebook 1 2	Stav Rezervováno Prodáno	
finish_date	date	Datum ukončení výstavby	
first_tour_date	datetime	Datum prohlídky	
first_tour_date_to	datetime	Datum prohlídky do	
flat_class	codebook 1 2 3	Typ bytu Mezonet Loft Podkrovní	
floor_area	int	Plocha podlahová	
floors	int	Podlaží počet	
furnished	codebook 1 2 3	Vybavení Ano Ne Částečně	
garage_count	int	Počet garáží	
garden_area	int	Plocha zahrady	
garret	bool	Půdní vestavba	
gas	multiselect 1 2	Plyn Individuální Plynovod	
gully	multiselect 1 2 3 4	Odpad Veřejná kanalizace ČOV pro celý objekt Septik Jímka	
heating	multiselect 1 2 3 4 5 6 7 8	Topení Lokální plynové Lokální tuhá paliva Lokální elektrické Ústřední plynové Ústřední tuhá paliva Ústřední elektrické Ústřední dálkové Jiné	
locality_citypart	string	Městská část	
locality_co	string	Číslo orientační	
locality_cp	string	Číslo popisné	
locality_street	string	Ulice	
loggia_area	int	Plocha lodžie	
matterport_url	string	Odkaz na VR prohlídku na webu Matterport	
mortgage	bool	Hypotéka	
mortgage_percent	double	Hypotéka - procenta	
nolive_total_area	int	Plocha nebytových prostor	
object_age	int	Rok výstavby	
object_kind	codebook 1 2 3 4	Poloha domu Řadový Rohový V bloku Samostatný	
object_location	codebook	Umístění objektu	

Pokračuje na další stránce

Název	Typ	Popis	Povinný
	1 2 3 4 5 6 7	Centrum obce Klidná část obce Rušná část obce Okraj obce Sídliště Polosamota Samota	
offices_area	int	Plocha kanceláří	
parking	int	Míst k parkování	
personal	codebook 1 2	Převod do OV Ano Ne	
price_auction_principal	double	Aukční jistina	povinné pro Dražby
price_expert_report	double	Znalecký posudek	povinné pro Dražby
price_minimum_bid	double	Minimální příhoz	povinné pro Dražby
production_area	int	Plocha výrobní	
protection	codebook 1 2 3	Ochrana Ochranné pásmo Národní park CHKO	
ready_date	date	Datum nastěhování	
reconstruction_year	int	Rok rekonstrukce	
road_type	multiselect 1 2 3 4	Komunikace Betonová Dlážděná Asfaltová Neupravená	
sale_date	date	Datum zahájení prodeje	
shop_area	int	Obchodní plocha	
spor_percent	double	Stavební spoření - procenta	
steps	string	Etapa	
store_area	int	Plocha skladů	
surroundings_type	codebook 1 2 3 4 5 6 7 8	Zástavba Obytná Obchodní a obytná Obchodní Komerční Průmyslová Venkovská Rekreační Rekreačně nevyužitá	
telecommunication	multiselect 1 2 3 4 5 6	Telekomunikace Telefon Internet Satelit Kabelová televize Kabelové rozvody Ostatní	
terrace_area	int	Plocha terasy	
transport	multiselect 1 2 3 4 5	Doprava Vlak Dálnice Silnice MHD Autobus	
underground_floors	int	Podlaží podzemní	

Pokračuje na další stránce

Název	Typ	Popis	Povinný
usable_area_ground	int	Užitná plocha přízemí	
user_status	bool	Aktivní	
water	multiselect	Voda 1 Místní zdroj 2 Dálkový vodovod	
workshop_area	int	Plocha dílen	

3.2 Atributy a číselníky developerského projektu

V následném seznamu atributů developerského projektu jsou nejdříve vypsané povinné položky.

Tabulka 2: Seznam atributů developerského projektu (metoda addProject)

Název	Typ	Popis	Povinný
annotation	string	Anotace	vždy
description	string	Popis	vždy
locality_city	string	Město	vždy
locality_description	string	Popis lokality	vždy
project_lifetime	codebook	Platnost 1 1 měsíc 2 3 měsíce 3 12 měsíců	vždy
project_name	string	Jméno	vždy
ceilings	codebook	Stropy 1 keramické 2 žebet. montované 3 žebet. monolitické 4 dřevěné trámové	
date_construct_completion	date	Dokončení výstavby	
date_move	date	Nastěhování	
date_sale	date	Zahájení prodeje	
developer_ic	int	IČ developera	
doors	codebook	Dveře 1 foliované 2 termofoliované 3 lakované 4 dýhované 5 masivní 6 laminátované 7 plastové	
facade_coats	codebook	Fasádní omítky 1 akrylátové 2 silikátové 3 silikonové	
floors	codebook	Podlahy 1 koberec 2 plovoucí laminátové 3 plovoucí dřevěné 4 dřevěné parkety 5 marmoleum 6 vinyl 7 linoleum 8 korek	
foundations	string	Základy	
interior_plasters	codebook	Vnitřní omítky	

Pokračuje na další stránce

Název	Typ	Popis	Povinný
	1	sádrové	
	2	štukové	
interior_staircase	codebook	Interiérové schodiště	
	1	celodřevěné	
	2	v kombinaci dřevo a nerez	
	3	celokovové	
	4	jiné	
interior_wall_lining	string	Vnitřní obklady	
kitchen_cabinets	codebook	Kuchyňská linka	
	1	ano	
	2	ne	
locality_citypart	string	Městská část	
locality_co	string	Číslo orientační	
locality_cp	string	Číslo popisné	
locality_latitude	double	Zeměpisná šířka	validní pouze s locality_longitude
locality_longitude	double	Zeměpisná délka	validní pouze s locality_latitude
locality_ruian	int	RUIAN lokality	validní pouze s locality_ruian_level
locality_ruian_level	int	RUIAN-level lokality	validní pouze s locality_ruian
locality_street	string	Ulice	
locality_uir	int	UIR lokality	validní pouze s locality_uir_level
locality_uir_level	int	UIR-level lokality	validní pouze s locality_uir
project_active	bool	Aktivní	
project_id	int	ID interní	
project_rkid	string	ID pro import	
project_url	string	Url projektu	
reinforced_concrete_staircase	codebook	Železobetonové schodiště	
	1	s kobercovou krytinou	
	2	s obložením laminátovou plovoucí podlahou	
	3	s obložením dřevěnou plovoucí podlahou	
	4	s obložením z masivu	
	5	s obložením z marmolea	
	6	s obložením z vinylu	
	7	s obložením z linolea	
	8	s obložením z korku	
roof	codebook	Střecha	
	1	sedlová	
	2	valbová	
	3	pultová	
	4	mansardová	
	5	plochá	
	6	plochá pochozí	
roofing	codebook	Krytina	
	1	pálená	
	2	betonová	
	3	plechová	
	4	břidlice	
	5	umělá břidlice	
	6	živičná	
	7	plastová fólie	
seller_id	int	ID maklere u srealit	pokud zadáno, seller_rkid je prázdné
seller_rkid	string	ID maklere u realitky	pokud zadáno, seller_id je prázdné
sheetmetal_structure	codebook	Klempířská konstrukce	
	1	pozink	
	2	měď	
	3	titanzinek	

Pokračuje na další stránce

Název	Typ	Popis	Povinný
	4	jiný	
siding	codebook	Vnější obklady	
	1	keramické	
	2	dřevěné	
	3	lícovými pásky nebo cihlami	
	4	kámen, mramor	
	5	cembonit	
windows	codebook	Okna	
	1	plastová	
	2	Eurookna	
	3	plastohliníková (Aluclip)	
	4	hliníková	

3.3 Návratové kódy a chybové hlášky

Status	Status message	Popis
200	OK	Vše je v pořádku
202	Accepted, RUIAN or UIR code not found	Inzerát/projekt přijat s tím, že předaný RUIAN nebo UIR kód nebyl rozpoznán
203	Accepted, address is ambiguous	Inzerát/projekt přijat s tím, že textová reprezentace lokality nabídla více jak jednu možnost
204	<i>Accepted, cannot modify required parameters</i>	<i>Inzerát přijat s tím, že modifikované povinné položky nebyly uloženy</i> (protože uplynula doba 3 hodiny od prvního importu)
404	Not found	Firma, makléř, inzerát, projekt či fotografie nebyla nalezena
405	Software key not active	Použitý SW klíč není aktivní
407	Session is bad	Přihlášení se nezdařilo, zkontrolujte heslo a SW klíč
410	Uploaded image is too large	Obrázek je příliš velký
412	Result image too small	Fotografie nemá dostatečné rozměry
413	Uploaded video is too large	Videozáznam je příliš velký
414	Photo limit exceeded	Počet nahraných fotografií překročil limit
415	Company is not active	Firma není aktivní
420	Cannot modify video during encoding	Během překódování videozáznamu s ním nelze pracovat (po 1 hodině stav encoding vyprší)
450	Photo already exists	Fotografie je již vložena, toto id je svázáno s jiným inzerátem
451	Photograph is duplicat.	Fotku nelze přidat, jelikož je duplicitní
452	Invalid parameters	Nejsou vyplněny všechny povinné položky nebo jsou špatného typu (v závorce je pak vidět, které položky se hláška týká)
453	Address not found	Selhala validace lokality
454	RUIAN or UIR code not resolved	Selhala nejen rozpoznání RUIAN nebo UIR kódu, ale i textová reprezentace
455	Invalid text item [error description]	Text obsahuje nevhodná slova nebo je příliš krátký (dlouhý)
461	Seller not found	ID makléře neexistuje
462	Seller login already exists	Login makléře je již použit
463	Seller login not found	Login makléře nelze dohledat v databázi Seznamu (neregistrován)
476	Invalid image format	Neznámý typ obrázku, používejte obrázky typu JPG
477	Advert was inserted this day	Nelze zvýhodnit (topovat) inzerát v den vložení
478	Cannot top duplicity advert	Nelze zvýhodnit (topovat) inzerát označený jako duplicitní
479	Advert must be published	Nelze zvýhodnit (topovat) nezveřejněný nebo neschválený inzerát
482	Advert top limit per day reached	Inzerát za aktuální den již byl zvýhodněn maximální počet krát
483	Cannot modify required parameters	Povinné položky již není možné modifikovat protože uplynula doba 3 hodiny od prvního importu. Tento návratový kód byl zrušen a nahrazen kódem 204
484	Cannot modify main parameters	Hlavní položky již není možné modifikovat protože uplynula doba 3 hodiny od prvního importu. Hlavní položky jsou: advert_function (typ inzerátu) a advert_type (kategorie inzerátu)
485	Ambiguous RKID	Zadané RKID není jednoznačné
500	Internal server error	Interní chyba systému (nahodilý výskyt=timeout, pravidelný=bug)

3.4 Stavby inzerátu ve výpise

Hodnota publish_status	Popis
0	Probíhá zpracování ...
1	Zveřejněný
2	Firma nemá kredit
3	Dlužná částka je vyšší než kredit
4	Firma nemá zaplacený produkt 'registrace'
5	Firma je smazaná
6	Dlužná částka uživatele je vyšší než kredit
7	Uživatel nemá kredit
8	Makléř není aktivní
9	Není přidělen obchodní zástupce
10	Inzerát nemá žádné fotografie
11	Inzerát je zakázaný
12	Inzerát je smazaný
13	Inzerát není aktivní
14	Inzerát není schválen
15	Inzerát je duplicitní
16	Inzerát čeká na vyhodnocení duplicity
17	Firma není aktivní
18	Inzerát je expirovaný
22	Překročení počtu inzerátů pro soukromou inzerci
24	Uživatel není aktivní
29	Inzerát nemá dostatek fotek (minimum 3)

3.5 Podporované formáty videozáznamů

Tabulka 3: *Výčet podporovaných video-formátů*

Zkratka	Název
4xm	4X Technologies format
IFF	IFF format
ISS	Funcom ISS format
MTV	MTV format
RoQ	raw id RoQ format
aac	raw ADTS AAC
ac3	raw AC-3
aea	MD STUDIO audio
aiff	Audio IFF
alaw	PCM A-law format
alsa	ALSA audio output
amr	3GPP AMR file format
anm	Deluxe Paint Animation
apc	CRYO APC format
ape	Monkey's Audio
asf	ASF format
ass	SSA/ASS format
au	SUN AU format
avi	AVI format
avs	AVS format
bethsoftvid	Bethesda Softworks VID format
bfi	Brute Force & Ignorance
bink	Bink
c93	Interplay C93
caf	Apple Core Audio Format
cavsvideo	raw Chinese AVS video

Pokračuje na další stránce

Zkratka	Název
cdg	CD Graphics Format
daud	D-Cinema audio format
dirac	raw Dirac
dnxhd	raw DNxHD (SMPTE VC-3)
dsicin	Delphine Software International CIN format
dtb	raw DTS
dv	DV video format
dv1394	DV1394 A/V grab
dxa	DXA
ea	Electronic Arts Multimedia Format
ea_cdata	Electronic Arts cdata
eac3	raw E-AC-3
f32be	PCM 32 bit floating-point big-endian format
f32le	PCM 32 bit floating-point little-endian format
f64be	PCM 64 bit floating-point big-endian format
f64le	PCM 64 bit floating-point little-endian format
ffm	FFM (FFserver live feed) format
film_cpk	Sega FILM/CPK format
filmstrip	Adobe Filmstrip
flac	raw FLAC
flic	FLI/FLC/FLX animation format
flv	FLV format
gsm	raw GSM
gxf	GXF format
h261	raw H.261
h263	raw H.263
h264	raw H.264 video format
idcin	id Cinematic format
image2	image2 sequence
image2pipe	pipelined image2 sequence
ingenient	raw Ingenient MJPEG
ipmovie	Interplay MVE format
iv8	A format generated by IndigoVision 8000 video server
lmlm4	lmlm4 raw format
m4v	raw MPEG-4 video format
matroska	Matroska file format
mjpeg	raw MJPEG video
mlp	raw MLP
mm	American Laser Games MM format
mmf	Yamaha SMAF
mov,mp4,m4a,3gp,3g2,mj2	QuickTime/MPEG-4/Motion JPEG 2000 format
mp3	MPEG audio layer 3
mpc	Musepack
mpc8	Musepack SV8
mpeg	MPEG-1 System format
mpegts	MPEG-2 transport stream format
mpegtsraw	MPEG-2 raw transport stream format
mpegvideo	raw MPEG video
msnwtcp	MSN TCP Webcam stream
mulaw	PCM mu-law format
mvi	Motion Pixels MVI format
mxp	Material eXchange Format
nc	NC camera feed format
nsv	Nullsoft Streaming Video
nut	NUT format
nuv	NuppelVideo format
ogg	Ogg

Pokračuje na další stránce

Zkratka	Název
oma	Sony OpenMG audio
oss	Open Sound System playback
psxstr	Sony Playstation STR format
pva	TechnoTrend PVA file and stream format
qcp	QCP format
r3d	REDCODE R3D format
rawvideo	raw video format
rl2	RL2 format
rm	RealMedia format
rpl	RPL/ARMovie format
rtsp	RTSP output format
s16be	PCM signed 16 bit big-endian format
s16le	PCM signed 16 bit little-endian format
s24be	PCM signed 24 bit big-endian format
s24le	PCM signed 24 bit little-endian format
s32be	PCM signed 32 bit big-endian format
s32le	PCM signed 32 bit little-endian format
s8	PCM signed 8 bit format
sdp	SDP
shn	raw Shorten
siff	Beam Software SIFF
smk	Smacker video
sol	Sierra SOL format
sox	SoX native format
swf	Flash format
thp	THP
tiertexseq	Tiertex Limited SEQ format
tmv	8088flex TMV
truehd	raw TrueHD
tta	True Audio
txd	Renderware TeXture Dictionary
u16be	PCM unsigned 16 bit big-endian format
u16le	PCM unsigned 16 bit little-endian format
u24be	PCM unsigned 24 bit big-endian format
u24le	PCM unsigned 24 bit little-endian format
u32be	PCM unsigned 32 bit big-endian format
u32le	PCM unsigned 32 bit little-endian format
u8	PCM unsigned 8 bit format
vc1	raw VC-1
vc1test	VC-1 test bitstream format
video4linux	Video4Linux device grab
video4linux2	Video4Linux2 device grab
vmd	Sierra VMD format
voc	Creative Voice file format
vqf	Nippon Telegraph and Telephone Corporation (NTT) TwinVQ
w64	Sony Wave64 format
wav	WAV format
wc3movie	Wing Commander III movie format
wsaud	Westwood Studios audio format
wsvqa	Westwood Studios VQA format
wv	WavPack
xa	Maxis XA File Format
yop	Psygnosis YOP Format
yuv4mpegpipe	YUV4MPEG pipe format

4 Příklady

4.1 PHP

```
<?php
// Definovani zakladnich pristupovych udaju
$clientId = 1; // ID klienta
$password = "vlozte-md5-hash-hesla"; // heslo zasifrovane pomoci md5
$key = "vlozte-importni-klic"; // importni klic

/**
 * Vypocte nove sessionId
 *
 * @param string $sessionId Aktualni sessionId
 * @param string $password Heslo k importnimu rozhrani
 * @param string $key Klic k importnimu rozhrani
 * @return string Nove sessionId
 */
function computeSessionId($sessionId, $password, $key) {
    $newVarPart = md5($sessionId . $password . $key);
    return substr($sessionId, 0, 48) . $newVarPart;
}

// Nacteni tridy pro praci s XMLRPC, napriklad z http://phpxmlrpc.sourceforge.net
require_once 'xmlrpc/lib/xmlrpc.inc';

// Nastaveni interniho kodovani na utf-8 pro spravny prenos diakritiky
PhpXmlRpc\XmlRpc::xmlrpc_internalencoding = 'UTF-8';

/* Pripojeni k importserveru */

$client = new xmlrpc_client('/RPC2', 'import.sreality.cz', 80);

// vytvoreni dotazu - ID klienta
// parametr - ID klienta
$params = array(new xmlrpcval($clientId, 'int'));

// vytvoreni zpravy pro ziskani sessionId
$msg = new xmlrpcmsg('getHash', $params);

// poslani dotazu na server
$response = $client->send($msg);

// nacteni vysledku
$getHash = php_xmlrpc_decode($response->value());

// je dotaz OK ?
if ($getHash['status'] != 200) {
    die("Chyba pri volani getHash [{$getHash['status']}: {$getHash['statusMessage']}");
}

// sessionId se vypocte z obdrzeneho (vystup getHash), hesla a SW klice
$sessionId = computeSessionId($getHash['output'][0]['sessionId'], $password, $key);

/* Prihlaseni na importserver */

// vytvoreni dotazu pro login a poslani na server
$params = array(new xmlrpcval($sessionId));
$msg = new xmlrpcmsg("login", $params);
$response = $client->send($msg);

// nacteni vysledku
$login = php_xmlrpc_decode($response->value());

// pokud je jiny status nez 200, nekde je chyba
if ($login['status'] != 200) {
    die("Chyba pri prihlaseni [{$login['status']}: {$login['statusMessage']}");
}
```

```
/* Ukazka pridani inzeratu */

// pred kazdym dalsim pozadavkem je treba znovu vypocitat sessionId
$sessionId = computeSessionId($sessionId, $password, $key);

$advert = array(
    "advert_function" => 1,                // prodej
    "advert_lifetime" => 1,                // 7 dni
    "advert_price" => 10000.0,
    "advert_price_currency" => 1,          // Kc
    "advert_price_unit" => 2,              // za mesic
    "advert_type" => 1,                    // byty
    "description" => "Pekny byt s vyhledem na zahradu.",
    "locality_city" => "Praha",
    "locality_inaccuracy_level" => 2,      // znepresneni adresy o 1 stupen
    "floor_number" => 1,                   // prvni patro
    "garage" => false,
    "loggia" => false,
    "balcony" => false,
    "terrace" => false,
    "ownership" => 1,                      // osobni vlastnictvi
    "parking_lots" => true,
    "advert_subtype" => 4,                  // Typ bytu 2+kk
    "usable_area" => 54,                    // Plocha bytu 54m^2
    "building_type" => 2,                   // Cihlova budova
    "building_condition" => 1,              // Stav objektu velmi dobry
    "cellar" => true,                       // Ma sklep
    "heating" => array(3, 4),               // topeni lokalni elektricke a ustredni plynove
    "telecommunication" => array(1, 2, 4), // telefon, internet, kabelova televize
    "seller_id" => 123456,
);

$msg = new xmlrpcmsg("addAdvert");
$msg->addParam(xmlrpc_encode($sessionId));
$msg->addParam(xmlrpc_encode($advert));
$response = $client->send($msg);

$result = xmlrpc_decode($response->value());
if (floor($result['status'] / 100) != 2) {
    die("Chyba pri ukladani inzeratu [{"status"}]: {"statusMessage"}");
}

/* Odhlaseni */

$sessionId = computeSessionId($sessionId, $password, $key);
$params = array(new xmlrpcval($sessionId));
$msg = new xmlrpcmsg("logout", $params);
$response = $client->send($msg);
$result = xmlrpc_decode($response->value());
if ($result['status'] != 200) {
    die("Chyba pri odhlaseni [{"status"}]: {"statusMessage"}");
}
?>
```

4.2 Python

```
#!/usr/bin/python
import xmlrpclib
import hashlib

clientId = 1 # ID klienta
password = "vlozte-md5-hash-hesla" # heslo zakryptovane pres md5
key = "vlozte-importni-klic" # importni klic

def NewSessionId(oldId, password, key):
    """
    Heslo by se melo pouzivat uz zakryptovane pres md5.
    Heslo i importni klic jsou zjistitelne v adminwebu.
    """
    varPart = hashlib.md5()
    varPart.update(oldId + password + key)
    return oldId[0:48] + varPart.hexdigest()

# pripojeni na importserver
client = xmlrpclib.ServerProxy("http://import.sreality.cz/RPC2")
# zavolame metodu getHash pro ziskani pocatecniho sessionId
getHash = client.getHash(clientId)
if getHash["status"] / 100 == 2:
    # vypocteni sessionId pro dalsi dotaz
    sessionId = NewSessionId(getHash["output"][0]["sessionId"], password, key)
    # prihlaseni na importserver
    response = client.login(sessionId)
    if response["status"] / 100 == 2:
        print "logged in."

    # vlozeni inzeratu
    sessionId = NewSessionId(sessionId, password, key)
    advert = {
        "advert_function": 1, # prodej
        "advert_lifetime": 1, # 7 dni
        "advert_price": 10000.0,
        "advert_price_currency": 1, # Kc
        "advert_price_unit": 2, # za mesic
        "advert_type": 1, # byty
        "description": "Pekny byt s vyhledem na zahradu.",
        "locality_city": "Praha",
        "locality_inaccuracy_level": 2, # znepresneni adresy o 1 stupen
        "floor_number": 1, # prvni patro
        "garage": False,
        "loggia": False,
        "balcony": False,
        "terrace": False,
        "ownership": 1, # osobni
        "parking_lots": True,
        "advert_subtype": 4, # Typ bytu 2+kk
        "usable_area": 54, # Plocha bytu 54m^2
        "building_type": 2, # Cihlova budova
        "building_condition": 1, # Stav objektu velmi dobry
        "cellar": True, # Ma sklep
        "heating": (3, 4), # topeni lokalni elektricke a ustredni plynové
        "telecommunication": (1, 2, 4), # telefon, internet, kabelova televize
        "seller_id": 123456,
    }
    response = client.addAdvert(sessionId, advert)
    if response["status"] / 100 == 2:
        print "Advert successfully added."
    else:
        print "addAdvert: %d %s" % (response["status"], response["statusMessage"])

    # odhlaseni
    sessionId = NewSessionId(sessionId, password, key)
    client.logout(sessionId)
else:
    print "login: %d %s" % (response["status"], response["statusMessage"])
else:
    print "getHash: %d %s" % (getHash["status"], getHash["statusMessage"])
```

4.3 Příklad XML požadavku

```
<?xml version='1.0'?>
<methodCall>
<methodName>addAdvert</methodName>
<params>
  <param>
    <value><string>2010031613125586000000000000000000000000000000002176fd01641c12c1f4..
    </string></value>
  </param>
  <param>
    <value><struct>
      <member>
        <name>user_status</name>
        <value><int>1</int></value>
      </member>
      <member>
        <name>advert_function</name>
        <value><int>1</int></value>
      </member>
      <member>
        <name>advert_price_unit</name>
        <value><int>1</int></value>
      </member>
      <member>
        <name>advert_lifetime</name>
        <value><int>3</int></value>
      </member>
      <member>
        <name>advert_price</name>
        <value><int>10000</int></value>
      </member>
      <member>
        <name>advert_type</name>
        <value><int>3</int></value>
      </member>
      <member>
        <name>advert_price_currency</name>
        <value><int>1</int></value>
      </member>
      <member>
        <name>advert_price_unit</name>
        <value><int>2</int></value>
      </member>
      <member>
        <name>seller_rkid</name>
        <value><string>makler01</string></value>
      </member>
      <member>
        <name>description</name>
        <value><string>Krásný slunný byt s výhledem na hřbitov.</string></value>
      </member>
      <member>
        <name>locality_street</name>
        <value><string>Radlická</string></value>
      </member>
      <member>
        <name>locality_cp</name>
        <value><string>2</string></value>
      </member>
      <member>
        <name>locality_city</name>
        <value><string>Praha</string></value>
      </member>
      <member>
        <name>locality_inaccuracy_level</name>
        <value><int>2</int></value>
      </member>
      <member>
        <name>building_type</name>
        <value><int>5</int></value>
      </member>
      <member>

```

```
        <name>building_condition</name>
        <value><int>1</int></value>
    </member>
    <member>
        <name>usable_area</name>
        <value><int>100</int></value>
    </member>
</struct></value>
</param>
</params>
</methodCall>
```