

# Sreality - dokumentace k importnímu rozhraní

## Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
1.1	Co je nového	1
1.2	Názvosloví	1
1.3	Typy dat	2
1.4	Struktura odpovědi serveru	2
1.5	Autorizace	2
1.6	Identifikace inzerátu, obecné vazby	2
1.7	Zadávání lokality	3
1.8	Vkládání videoprohlídek	4
<b>2</b>	<b>Metody</b>	<b>5</b>
2.1	Autorizační metody	5
2.2	Správa inzerátů	6
2.3	Správa fotek u inzerátů	7
2.4	Správa videoprohlídek u inzerátů	9
2.5	Správa makléřů	10
2.6	Správa developerských projektů	11
2.7	Správa fotek u developerských projektů	12
2.8	Správa videoprohlídek u developerských projektů	14
2.9	Čtení statistik	15
2.10	Metoda dostupná bez přihlášení	16
<b>3</b>	<b>Datové struktury a číselníky</b>	<b>17</b>
3.1	Atributy a číselníky inzerátu	17
3.2	Atributy a číselníky developerského projektu	22
3.3	Návratové kódy a chybové hlášky	24
3.4	Stavy inzerátu ve výpise	25
3.5	Podporované formáty videozáznamů	25
<b>4</b>	<b>Příklady</b>	<b>28</b>
4.1	PHP	28
4.2	Python	29
4.3	Příklad XML požadavku	30

# 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 sreality ([info@sreality.cz](mailto: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 2.1.3:

- Byl zaveden limit 30 obrázků na jeden inzerát nebo developerský projekt. Při překročení vrací importní rozhraní chybový kód 414, viz kapitola [3.3](#)
- Původní atribut poznámky k ceně (*advert\_price\_note*) byl zrušen. Je plně nahrazen dílčími atributy *advert\_price\_commission* a *advert\_price\_charge*
- Metoda addProject nevyžaduje IČ developera jako povinnou položku. V detailu developerského projektu na webu sreality.cz se neukazuje kontakt na developera, není-li zadán, viz tabulka [2](#)

### Proti verzi 2.1.2:

- Přidány atributy pro vkládání obou speciálních příloh u drazeb v pdf formátu. Jedná se o dražební vyhlášku a posudek znalce, tabulka [1](#)
- Metody vypisující fotografie u inzerátu a projektu vrací navíc atribut order, kapitola [2.3.3](#) a [2.7.3](#)
- Metoda listSeller (kapitola [2.5.3](#)) vrací i přihlašovací login makléře

## 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 24.

### 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 současná *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 Zadávání 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 celou věc zjednodušuje na pouhé získání detailů o zadaném 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.

**Zahraníční 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** 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 \(http://forms.mpsv.cz/uir/\)](http://forms.mpsv.cz/uir/) je možno získat zdarma CD-ROM s daty a programy pro prohlížení a periodickou aktualizaci těchto dat. Při použití UIR kódu není nutno vyplňovat adresu, kromě případu, kdy UIR kód není dostatečně podrobný, pak je možno adresu upřesnit textově. (Příklad: K bytu máme k dispozici 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 UIR).

Popis uir se předává pomocí parametrů *locality\_uir* (vlastní uir) a *locality\_uir\_level* (přesnost, typ). Odpovídající *locality\_uir\_level* naleznete v tabulce níže.

Název (locality_uir_level)	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 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.

## 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 Srealitách 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 30MB, podporované formáty lze dohledat na straně 25. 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 *session\_id*, 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 *session\_id* 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 session_id     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, *client\_id* 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 17. 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, UIR-ADR, GPS souřadnice, viz kapitola 1.7.

**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
                                461=Makler neexistuje
                                491=Projekt neexistuje
                                )
    string statusMessage      Slovní popis statusu.
    array struct output
    (
        int advert_id         Cislo inzeratu
    )
}
```

### 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 reality
    string advert_url       odkaz na detail inzerátu na server sreality.cz
    int advert_type         druh inzerátu (byty, pozemky...) podle dokumentace
    int 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 /refBindings). Zvýhodnění inzerátu (tzv. topování) je možnost inzerát umístit do horních pozic ve výpise inzerátů. Topování je zpoplatněno dle aktuálního ceníku sreality. Inzerát lze "topovat" jen jednou denně. Nelze topovat v den vložení inzerátu, ani pokud není inzerát zveřejněn. Také inzerát označený jako duplicitní nelze zvýhodnit. Počet topovacích operací za den je omezen dle aktuálních podmínek.

Nelze zvýhodnit (topovat) inzerát v den vložení. Nelze zvýhodnit (topovat) inzerát označený jako duplicitní. Nelze zvýhodnit (topovat) nezveřejněný nebo neschválený inzerát.

```

struct
{
    int status                Status (200=OK
                                404=Inzerát nenalezen
                                407=Neplatne prihlaseni
                                452=Nejsou vyplněny všechny povinné položky
                                nebo jsou špatného typu
                                477=Nelze topovat v den vložení
                                478=Nelze topovat duplicitní inzerát
                                479=Nelze topovat nezveřejněný inzerát
                                482=Dnes již byl inzerát zvýhodněn
                                )
    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
    string 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 500kB. 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=Inzerat nebo fotografie nenalezen
                                407=Neplatne prihlaseni
                                412=Sirka nebo vyska obrazku je prilis mala
                                450=Fotka patri k jinemu inzeratu
                                452=Nejsou vyplneny vsechny povinne polozky
                                nebo jsou spatneho typu
                                476=Neni to JPEG/GIF/PNG obrazek
                                )
    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 30MB. Po překročení této velikosti server vrací chybu a video nezpracuje. Zpracování je blíže popsáno v kapitole 1.8, podporované formáty lze nalézt na straně 25.

```

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 25
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
    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) 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     interní 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 22. 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, 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.

```
struct
{
    int status                Status (200=OK
                              404=Projekt nenalezen
                              407=Neplatne prihlaseni
                              452=Nejsou vyplneny vsechny povinne polozky
                              nebo jsou spatneho typu
                              471=IC developera nenalezeno
                              )
}
```

```

    )
    string statusMessage      Slovní popis statusu.
    array struct output
    (
        int project_id        Číslo 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=Neplatné přihlášení
                                452=Nejsou vyplněny všechny povinné položky
                                nebo jsou špatného 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=Neplatné přihlášení
    )
    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
    string 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 500kB. 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
                                412=Sirka nebo vyska obrazku je prilis mala
                                450=Fotka patri k jinemu projektu
                                452=Nejsou vyplneny vsechny povinne polozky
                                nebo jsou spatneho typu
                                476=Neni to JPEG/GIF/PNG obrazek
                                )
    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 realitky
        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 30MB. Po překročení této velikosti server vrací chybu a video nezpracuje. Zpracování je blíže popsáno v kapitole 1.8, podporované formáty lze nalézt na straně 25.

```

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 10.

## 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=Neplatné přihlášení
                               )
    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
        double topped_price     strženo za zvýhodnění inzerátu
        double advert_price    strženo za beznou inzerci
        top                   inzerát byl(1)/nebyl(0) zvýhodněn [0,1]
    )
}

```

### 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=Neplatné přihlášení
                               )
    string statusMessage      Slovní popis statusu.
    array struct output
    (
        string date           datum dne
        int views              počet shlédnutí
        double price           stržená cena
    )
}

```



### 2.9.3 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
    )
}
```

## 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	Datum do 7 dní 14 dní 30 dní 45 dní 90 dní	vždy
advert_price	double	Cena	vždy
advert_price_currency	codebook 1 2 3	Měna CZK USD EUR	vždy
advert_price_unit	codebook 1 2 3 4 5 6 7 8	Jednotka za nemovitost za měsíc za m <sup>2</sup> za m <sup>2</sup> /měsíc za m <sup>2</sup> /rok za rok za den za 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_subtype	codebook 1 2 3 4 5 6 7 8 9 10 11 12	Podkategorie Garsonka 1+kk 1+1 2+kk 2+1 3+kk 3+1 4+kk 4+1 5+kk 5+1 6+kk	povinné, platí však vždy jen pro jednu kategorii Byty Byty Byty Byty Byty Byty Byty Byty Byty Byty Byty Byty

Pokračuje na další stránce

Název	Typ	Popis	Povinný
	13	6+1	Byty
	14	7+kk	Byty
	15	7+1	Byty
	16	Atypický	Byty
	17	Jiný	Byty
	18	Pro komerční výstavbu	Pozemky
	19	Pro bydlení	Pozemky
	20	Zemědělská půda	Pozemky
	21	Les	Pozemky
	22	Trvalý travní porost	Pozemky
	23	Zahrada	Pozemky
	24	Ostatní	Pozemky
	25	Kanceláře	Komerční
	26	Sklady	Komerční
	27	Výroba	Komerční
	28	Obchodní prostory	Komerční
	29	Ubytování	Komerční
	30	Restaurace	Komerční
	31	Zemědělské objekty	Komerční
	32	Jiný	Komerční
	33	Chaty	Ostatní
	34	Garáže	Ostatní
	35	Historické objekty	Ostatní
	36	Jiný	Ostatní
	37	Rodinný	Domy
	38	Činžovní	Domy
	39	Vily	Domy
	40	Na klíč	Domy
	41	Dřevostavby	Domy
	42	Nízkoenergetické	Domy
	43	Chalupy	Ostatní
	44	Zemědělské usedlosti	Ostatní
	45	Objekty obč. vybavenosti	Ostatní
	46	Rybářky	Ostatní
balcony	bool	Balkón	povinné pro Byty, Domy
building_condition	codebook	Stav objektu	povinné pro Byty, Domy, Ostatní, Komerční
	1	Velmi dobrý	
	2	Dobrý	
	3	Špatný	
	4	Ve výstavbě	
	5	Projekt	
	6	Novostavba	
	7	K demolici	
	8	Před rekonstrukcí	
	9	Po rekonstrukci	
building_type	codebook	Budova	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	Podlaží umístění	povinné pro Byty

Pokračuje na další stránce

Název	Typ	Popis	Povinný
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_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, Domy
object_type	codebook 1 2	Typ domu Přízemní Patrový	povinné pro Domy, Komerční
ownership	codebook 1 2 3	Vlastnictví Osobní Družstevní Jiné	povinné pro Byty
parking_lots	bool	Parkovací stání	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, Domy
usable_area	int	Plocha užitná	povinné pro Byty, Domy, Ostatní, Komerční
acceptance_year	int	Rok kolaudace	
advert_code	string	ID zakázky	
advert_price_charge	codebook 1 2	poplatky včetně poplatků bez poplatků	
advert_price_commission	codebook 1 2	Provize včetně provize + provize RK	
advert_price_legal_services	codebook 1 2	Právní servis včetně právního servisu bez právního servisu	
advert_price_negotiation	bool	Cena k jednání	
advert_price_text_note	string	Poznámka k ceně	
advert_price_vat	codebook 1 2	DPH včetně DPH 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 1 2	Druh dražby nedobrovolná dobrovolná	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	bool	Bazén	
basin_area	int	Plocha bazénu	
beds	int	Počet lůžek	
beginning_date	date	Datum zahájení výstavby	
building_area	int	Plocha zastavěná	
ceiling_height	double	Výška stropu	
cellar_area	int	Plocha sklepu	
description_en	string	Anglický popis	

Pokračuje na další stránce

Název	Typ	Popis	Povinný
easy_access	bool	Bezbariérový přístup	
electricity	multiselect 1 2 4	Elektřina 120V 230V 400V	
equipment	bool	Vybavení	
estate_building_type	codebook 1 2 3 4 5 6 7 8	Stavba Budova, hala Inženýrská, pozemní stavba Rodinný dům Rekreační chata Garáž Vedlejší stavba Studna Ostatní	
extra_info	codebook 1 2	Stav Rezervováno Prodáno	
finish_date	date	Datum ukončení výstavby	
first_tour_date	datetime	Datum prohlídky	
flats	int	Počet bytů	
floor_area	int	Plocha podlahová	
floors	int	Podlaží počet	
free_flats	int	Počet volných bytů	
free_offices	int	Počet volných kanceláří	
furnished	codebook 1 2 3	Zařízeno 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	
height	int	Rozměry - výška	
house_count	int	Počet domů	
length	int	Rozměry - délka	
locality_citypart	string	Městská část	
locality_co	string	Číslo orientační	
locality_cp	string	Číslo popisné	

Pokračuje na další stránce

Název	Typ	Popis	Povinný
locality_street	string	Ulice	
loggia_area	int	Plocha lodžie	
mortgage	bool	Hypotéka	
mortgage_percent	double	Hypotéka - procenta	
nolive_total_area	int	Plocha nebytových prostor	
object_age	int	Rok výstavby	
object_count	int	Počet objektů	
object_kind	codebook	Poloha domu 1 Řadový 2 Rohový 3 V bloku 4 Samostatný	
object_location	codebook	Umístění objektu 1 Centrum obce 2 Klidná část obce 3 Rušná část obce 4 Okraj obce 5 Sídliště 6 Polosamota 7 Samota	
offices	int	Počet kanceláří	
offices_area	int	Plocha kanceláří	
parking	int	Míst k parkování	
personal	codebook	Převod do OV 1 Ano 2 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	Ochrana 1 Ochranné pásmo 2 Národní park 3 CHKO	
ready_date	date	Datum nastěhování	
reconstruction_year	int	Rok rekonstrukce	
road_type	multiselect	Komunikace 1 Betonová 2 Dlážděná 3 Asfaltová 4 Neupravená	
sale_date	date	Datum zahájení prodeje	
seats	int	Počet míst	
shop_area	int	Obchodní plocha	
shops	int	Počet obchodů	
spor_percent	double	Stavební spoření - procenta	
steps	string	Etapa	
store_area	int	Plocha skladů	
surroundings_type	codebook	Zástavba 1 Obytná 2 Obchodní a obytná 3 Obchodní 4 Komerční 5 Průmyslová 6 Venkovská 7 Rekreační	

Pokračuje na další stránce

Název	Typ	Popis	Povinný
	8	Rekreačně nevyužitá	
telecommunication	multiselect	Telekomunikace	
	1	Telefon	
	2	Internet	
	3	Satelit	
	4	Kabelová televize	
	5	Kabelové rozvody	
	6	Ostatní	
terrace_area	int	Plocha terasy	
transport	multiselect	Doprava	
	1	Vlak	
	2	Dálnice	
	3	Silnice	
	4	MHD	
	5	Autobus	
underground_floors	int	Podlaží podzemní	
usable_area_ground	int	Užitná plocha přízemí	
user_status	bool	Aktivní	
water	multiselect	Voda	
	1	Místní zdroj	
	2	Dálkový vodovod	
width	int	Rozměry - šířka	
workshop_area	int	Plocha dílen	

### 3.2 Atributy a číselníky developerského projektu

V následném seznamu atributů developerského projektu jsou nejdříve vypsány 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	vždy
	1	1 měsíc	
	2	3 měsíce	
	3	12 měsíců	
project_name	string	Jméno	vždy
ceilings	codebook	Stropy	
	1	keramické	
	2	želbet. montované	
	3	želbet. 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	

Pokračuje na další stránce

Název	Typ	Popis	Povinný
	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	
	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_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_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	
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	

Pokračuje na další stránce



Název	Typ	Popis	Povinný
	2 3 4 5 6 7	betonová plechová břidlice umělá břidlice živičná plastová fólie	
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
sheetmetal_structure	codebook 1 2 3 4	Klempířská konstrukce pozink měď titanzinek jiný	
siding	codebook 1 2 3 4 5	Vnější obklady keramické dřevěné lícovými pásky nebo cihlami kámen, mramor cembonit	
windows	codebook 1 2 3 4	Okna plastová Eurookna plastohliníková (Aluclip) hliníková	

### 3.3 Návrátové kódy a chybové hlášky

Status	Status message	Popis
200	OK	Vše je v pořádku
202	Accepted, UIR code not found	Inzerát/projekt přijat s tím, že předaný UIR kód nebyl rozpoznán
203	Accepted, address is ambiguous	Inzerát/projekt přijat s tím, že textová reprezentace lokality nabídla více jak jednu možnost
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
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
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
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	UIR code not resolved	Selhala nejen rozpoznání UIR kódu, ale i textová reprezentace
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 already topped	Inzerát za aktuální den již byl zvýhodněn
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 RUS je vyšší než kredit
7	RUS nemá kredit
8	Klient 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

### 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
cdg	CD Graphics Format
daud	D-Cinema audio format

Pokračuje na další stránce

Zkratka	Název
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
mpegs	MPEG-2 transport stream format
mpegsraw	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
oma	Sony OpenMG audio
oss	Open Sound System playback

Pokračuje na další stránce

Zkratka	Název
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

```
<?
/*****
Prihlaseni
*****/

// nacteni tridy z PEARu
require_once 'XML/RPC.php';

// connect na server
$client = new XML_RPC_Client('/RPC2', 'http://import.sreality.cz', 80);

// vytvoreni dotazu - ID klienta
// parametr - ID klienta
$params = array(new XML_RPC_Value(123456, 'int'));
$msg = new XML_RPC_Message('getHash', $params);

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

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

// je dotaz je OK ?
if ($getHash['status'] == 200) {

    // prekopirovani outputu do zvlast promenne
    $output = $getHash['output'];

    // vytvoreni dotazu pro login a poslani na server
    // sessionId se vypocte z obdrzeneho (vystup getHash), hesla a SW klice
    $newVarPart = md5($output[0]['sessionId'] . md5('heslo') . 'klic');
    $newSessionId = substr($output[0]['sessionId'], 0, 48) . $newVarPart;
    $params = array(new XML_RPC_Value($newSessionId));
    $msg = new XML_RPC_Message("login", $params);
    $response = $client->send($msg);

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

    // pokud je status 200, je to OK
    if ($login['status'] != 200) {
        echo "Chyba [$login[status]]: $login[statusMessage]\n";
    }
} else {
    echo "Chyba [$getHash[status]]: $getHash[statusMessage]\n";
}
?>
```

## 4.2 Python

```
#!/usr/bin/python

import xmlrpclib
import hashlib

def NewSessionId(oldId):
    # heslo by se melo pouzivat uz zamanglovane
    # heslo i softwarovy klic jsou zjistitelne v adminwebu
    varPart = hashlib.md5()
    varPart.update(oldId + 'heslo v md5' + 'software_key')
    return oldId[0:48] + varPart.hexdigest()

client = xmlrpclib.ServerProxy("http://import.sreality.cz/RPC2")

# id klienta je jediny parametr
getHash = proxy.getHash( 123 )

if getHash["status"] / 100 == 2:
    sessionId = NewSessionId(getHash["output"][0]["sessionId"])

    response = client.login(sessionId)

    if response["status"] / 100 == 2:
        print "logged in."
    else:
        print "login: %d %s" % (response["status"], response["statusMessage"])

else:
    print "getHash: %d %s" % (getHash["status"], getHash["statusMessage"])
```



```
        <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>
```