

## Správa kanceláří přes importní rozhraní

Představuje rozhraní na bázi XML-RPC sloužící pro správu realitních kanceláří na serveru viaReality.cz a dalších přidružených aplikacích. Rozhraní je zabezpečeno přihlášením a každý subjekt může spravovat ze svého SW pouze svou množinu realitních kanceláří.

## Adresa importního rozhraní a přihlašovací údaje

- Adresa pro import na **ostrý server** je na adrese:  
<http://client.viareality.cz/import/accounts.rem>
- Adresa pro import na **testovací server** je na adrese:  
<http://test.viareality.cz/import/accounts.rem>

Pro povolení přístupu k testovacímu či ostrému serveru a získání přihlašovacích údajů nás prosím kontaktujte na emailové adrese [podpora@viareality.cz](mailto:podpora@viareality.cz).

## 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 vstupují současné *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
```

## Specifikace metod

Zde je uvedena množina podporovaných metod, specifikace jednotlivých metod následuje dále v dokumentu:

- `getHash`
- `login`
- `logout`
- `addAccount`
- `listAccount`
- `delAccount`
- `addLogo`
- `version`

### **getHash**(id as int)

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

**Id** = přidělená hodnota sloužící k identifikaci spravované skupiny účtů

Metoda slouží k získání řetězce pro hashování hesla. Po zavolání s existujícím 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
    )
}
```

### **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 viaReality zvlášť pro každého správce účtů v závislosti na použitém exportním software. Nazpět je vrácen status úspěšnosti.

**Poznámka:** Bez úspěšného dokončení metody *login* je celá importní relace ve stavu nepřihlášeno a nelze ve správě registrací pokračovat.

Vrací-li metoda opakovaně 407 `Bad session` je špatně zadané heslo, importní klíč, id u klienta nebo na importním serveru. Je třeba zkontrolovat tyto údaje.

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

## logout(session\_id as string)

Odhlášení.

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

## addAccount(session\_id as string, account\_data as struct)

Přidání/editace účtu realitní kanceláře.

Metodě se předává slovník *account\_data*, který obsahuje veškeré hodnoty o vytvářeném nebo měněném účtu. Veškeré možné hodnoty jsou uvedeny níže. Pokud chybí některá povinná položka, metoda skončí s chybou.

Metoda vrací identifikaci účtu (realitní kanceláře) a hodnoty pro přihlášení – jak do webové administrace, tak údaje pro export realitních dat.

Heslo pro přístup do webové administrace je vráceno jen v případě prvního vložení kanceláře. Klient si heslo může později změnit ve své administraci. Pokud je potřeba heslo změnit kontaktujte viaReality, nebo můžete využít formulář na obnovu hesla na <http://www.viareality.cz/ucet/obnova-hesla>.

Identifikátor účtu (accountId) je nutné uchovat pro další operace s účtem (aktualizace, smazání,...). Detail realitní kanceláře na viaReality lze prohlédnout pomocí url: [http://www.viareality.cz/<ACCOUNT\\_ID>](http://www.viareality.cz/<ACCOUNT_ID>)

Pro změnu hesla pro export dat je nutné kontaktovat technickou podporu viaReality ([podpora@viareality.cz](mailto:podpora@viareality.cz)).

```

struct
{
    int status Status (200=Registrace byla úspěšně uložena
                      407=Neplatné přihlášení
                      404=Software s <ID> není v systému nalezen
                      409=Přihlašovací jméno (email) je obsazeno,
                          email má špatný formát nebo účet s tímto
                          emailem byl již smazán
                      452=Chybí některá povinná položka
                      500=Interní chyba systému (akci opakujte))
    string statusMessage Slovní popis statusu.
    array struct output
    (
        ..
        Položky stejné jako account_data
        ..
        string webName Přihlašovací jméno do webové administrace
        string webPassword Heslo do webové administrace
        string exportName Přihlašovací jméno pro export
        string exportPassword Heslo pro export
        string exportSWKey SW klíč pro export
    )
}

```

Slovník *account\_data* obsahuje následující položky (tučně jsou uvedeny povinné položky):

Typ	Název	Popis
<b>string</b>	<b>name</b>	Název kanceláře
int	accountId	Identifikace účtu, pokud je hodnota uvedena, jedná se o editaci již existujícího účtu.
Int	exportSoftwareID	Id exportního software realitní kanceláře. Pokud není uvedeno, popř. < 1, není vytvořen export pro danou kancelář. Pokud je uvedeno v rámci aktualizace a daná RK nemá propojení, je vytvořeno nové pro ID software. Hodnota 3 je hodnota pro SW Maxis, další ID podporovaných SW poskytneme na vyžádání.
<b>string</b>	<b>email</b>	Email kanceláře, email je použit i jako přihlašovací jméno do systému (webová administrace), proto musí být v rámci aplikace viaReality jedinečný.
<b>bool</b>	<b>visible</b>	True = kancelář je viditelná na server viaReality False = kancelář není viditelná na serveru viaReality, ale není ovlivněna její dostupnost v navázaných projektech (např. brandované mobilní aplikace).
<b>string</b>	<b>www</b>	Webové stránky kanceláře.
string	telephone	Telefon do realitní kancelář.
string	telephone2	Další telefon do realitní kanceláře.
string	telephone3	Další telefon do realitní kanceláře.
string	locality_city	Město, ve kterém je umístěno sídlo realitní kanceláře.
string	locality_citypart	Název městské části.
string	locality_co	Číslo orientační.
string	locality_country	Stát, kde je sídlo RK.
string	locality_cp	Číslo popisné.

int	locality_inaccuracy_level	Udává úroveň zneřesnění adresy při zobrazení ve veřejné části.
double	locality_latitude	Zeměpisná šířka.
double	locality_longitude	Zeměpisná délka.
int	locality_ruian	RUIAN kód adresy.
int	locality_ruian_level	RUIAN level adresy (stejně jako specifikace SReality)
string	locality_street	Název ulice.
int	locality_uir	UIR kód adresy. (pokud je uveden RUIAN) je tato hodnota ignorována.
int	locality_uir_level	Uir level adresy.
string	contactName	Jméno a příjmení odpovědné osoby v rámci RK. Odpovědná osoba se nikde na viaReality.cz nezobrazuje.
string	contactEmail	Email odpovědné osoby.
string	contactTelephone	Telefon odpovědné osoby.
string	dic	DIČ kanceláře.
int	ico	IČO kanceláře.

## listAccount(session\_id as string)

Výpis všech účtů přihlášeného správce.

V *output* je vráceno pole, kde každý prvek pole obsahuje strukturu (viz níže). Účty jsou seřazeny podle identifikátoru registrace (id).

```

struct
{
    int status Status (200=OK
                                407=Neplatné přihlášení)
    string statusMessage Slovní popis statusu
    array struct output
    (
        ..
        Položky stejné jako account_data
        ..
        string webName Přihlašovací jméno do webové administrace
        string webPassword Heslo do webové administrace
        string exportName Přihlašovací jméno pro export
        string exportPassword Heslo pro export
        string exportSWKey SW klíč pro export
    )
}

```

## delAccount(session\_id as string, account\_id as int)

Vymazání účtu.

Ihned po smazání účtu (tj. zavolání metody `delAccount`) nebudou žádná data z účtu dostupná na serveru viaReality ani v navázaných aplikacích (brandované mobilní aplikace). Pokud účet neexistuje, bude vrácen status *OK (200)*.

```
struct
{
    int status Status (200=OK,
                                404=RK nebyla nalezena,
                                407=Neplatné přihlášení)
    string statusMessage Slovní popis statusu
    array struct output
    (
    )
}
```

## addLogo(session\_id as string, account\_id as int, logo as binary)

Vložení loga k již uložené registraci.

Vstupními parametry jsou `session_id` a `account_id` a vlastní data obrázku (loga). Metoda provede kontrolu vstupních dat (rozměr a formát loga) a pokud nedojde k zamítnutí loga, uloží se k účtu realitní kanceláře. Pokud již realitní kancelář logo má, bude nahrazeno tímto novým logem.

Loga k účtům není potřeba nahrávat. Naše grafické oddělení každodenně kontroluje loga, a pokud nějaké chybí, vytvoří jej z volně dostupných materiálů – v souladu s grafikou kanceláře. S kancelářemi dlouhodobě spolupracujeme, a proto známe řadu logomanuálů. Výhodou tohoto řešení je, že logo je vždy vytvořeno přesně a profesionálně a vždy vypadá dobře. Tato služba není nijak zpoplatněna. Reakční doba je maximálně v řádu dnů, ale většinou se pohybuje do 24 hodin.

Pokud si přejete loga nahrávat, je vyžadován čtvercový obrázek – *jpeg* nebo *png*, minimální délka hrany je *200px* a maximální velikost *5MB*.

```
struct
{
    int status Status (200=OK
                                404=Účet nebyl na serveru nalezen
                                407=Neplatné přihlášení
                                412=Logo je příliš malé nebo není čtvercové
                                476=Obrázek není JPEG/PNG nebo je prázdný)
    string statusMessage Slovní popis statusu
    array struct output
    (
    )
}
```

## version()

Výpis verze importu. Metoda nevyžaduje přihlášení.

```
struct {
```

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

## Technická podpora viaReality

Pokud budete mít nějaké dotazy nebo problémy během implementace, v průběhu nasazení nebo při používání, neváhejte a obraťte se na technickou podporu viaReality.

Email: [podpora@viareality.cz](mailto:podpora@viareality.cz)

Tel: 604 114 475