

Specifikace exportního rozhraní z aplikace MultiEstate

Obsah

1. Úvod.....	2
1.1 Přístup k rozhraní.....	2
2. Konfigurace rozhraní.....	2
2.1 Popis struktury konfigurace.....	2
version.....	3
offer.....	3
codebook.....	4
brokers.....	4
photos.....	4
2.2 DTD specifikace.....	5
3. Metody.....	5
3.1 Datové typy.....	6
3.2 Struktura návratové hodnoty.....	6
3.3 Povinné metody.....	7
check.....	7
getConfig.....	8
sendOffer.....	8
deleteOffer.....	9
reportError.....	9
3.4 Nepovinné metody.....	10
checkBroker.....	10
sendBroker.....	10
sendFreeItems.....	10
checkPhotos.....	12
sendPhoto.....	12
deletePhoto.....	13

1. Úvod

Tento dokument popisuje princip klientské části exportního rozhraní, které umožňuje odeslat vybraná data z aplikace MultiEstate na jiný server pro jejich veřejné publikování či jiné zpracování. Pro přenos se využívá technologie XML-RPC, kdy pomocí HTTP protokolu je přenášen XML soubor obsahující požadovaná data ve specifické struktuře definované XML-RPC standardem. Pro základní funkčnost tedy postačí webový server a libovolná metoda zpracování přijatých dat (PHP, ASP.NET, Java...). Veškerá komunikace s rozhraním probíhá v kódování UTF-8.

1.1 Přístup k rozhraní

Každý server požadující export dat z aplikace musí mít ve zdrojové aplikaci zřízen profil, který je identifikován unikátním autorizačním klíčem. Pro zajištění bezpečnosti přenosu je tento klíč přenášen mezi vstupními parametry většiny metod. Tento údaj lze získat od provozovatele konkrétní instalace MultiEstate.

2. Konfigurace rozhraní

Rozhraní je postaveno s cílem být maximálně univerzální a znovu použitelné pro libovolný počet cílových serverů. Z tohoto důvodu a kvůli minimalizaci přenášených dat může být každý profil nakonfigurován tak, aby přenášel jen požadovaná data.

Základní údaje o nabídce jako např. název, popis či cena (podrobné přehledy v tabulkách dále) jsou odesílány vždy a nelze je konfigurací změnit. Pomocí konfigurace můžete určit, zda Váš server potřebuje data o makléřích, fotografie a další rozšiřující položky nabídky. U fotografií lze nastavit jejich velikost, maximální počet a vodotisk, který je automaticky přidán do těchto fotografií pro odlehčení Vašemu serveru.

2.1 Popis struktury konfigurace

Kompletní konfigurace je zapsána formou validního XML verze 1.0, které musí splňovat DTD specifikaci uvedenou na konci tohoto dokumentu. Tato konfigurace musí být neustále k dispozici na serveru, nejlépe zapsaná v souboru pro snadné sledování změn.

Následuje vychozí konfigurace, která bude použita v případě, že nedodáte vlastní konfiguraci (výstupní hodnota ExportTime metody check bude 0).

Vychozí konfigurace exportu

```
<?xml version="1.0" encoding="utf-8"?>
<econfig>
  <version>1</version>
  <offer>
    <public_url />
    <freeitems enabled="true" />
  </offer>
  <codebook enabled="true">
    <class>1</class>
    <trans>1</class>
    <stage>1</stage>
    <priceunit>1</priceunit>
    <currency>1</currency>
    <contract>1</contract>
    <freeitem>1</freeitem>
    <foptions>1</foptions>
  </codebook>
  <brokers enabled="true">
    <photos enabled="true" size="original" />
  </brokers>
  <pictures enabled="true" maxcount="0">
    <watermark>...binární data watermarku zakodovaná Base64...</watermark>
    <picture size="640x480" type="jpeg" usewatermark="false" />
  </pictures>
</econfig>
```

version

V této větvi můžete určit, kterou verzi rozhraní chcete používat. V případě, že bude vydána nová verze rozhraní, budete o tom informováni pomocí metody `check` (viz. níže). Následně můžete provést aktualizaci Vašeho rozhraní podle nové specifikace a pomocí tohoto konfiguračního nastavení oznámit, že se má použít tato nová verze pro Váš server. Neuvádějte v konfiguraci číslo verze, na které není Váš server připravený. Pokud neuvedete žádné nebo neexistující číslo verze, bude automaticky použita poslední dostupná verze.

offer

Nejkomplexnější větev **offer** představuje nastavení dalších dat, která chcete z aplikace získat navíc a případná další nastavení.

Prvním nastavením je položka **public_url**, která může obsahovat veřejnou URL adresu exportované nabídky. Tato adresa je následně dostupná pro makléře v aplikaci (např. pro manuální ověření exportu). V této hodnotě musí být obsažen řetězec „#OfferId#“, který bude následně nahrazen číselným Id získaným z metody `sendOffer`.

Dalším nastavením atributu *enabled* větve **freeitems** povolíte použití metody pro zaslání volných položek (`sendFreeItems`). V budoucích verzích rozhraní pravděpodobně dojde k podrobnějšímu rozšíření pro určení, které volné položky se mají exportovat. Do té doby budou zasílány veškeré volné položky dostupné u dané nabídky.

codebook

Tato větev ovlivňuje použití tzv. CodeBook. Jsou to jednoduché tabulky, kde určité slovní vyjádření je nahrazeno unikátním označením pro snazší modifikovatelnost aplikace (někdy se těmto seznamům také říká číselníky). Identifikátory ve většině případů budou číselné, ale mohou být i řetězcové.

Díky tomuto mechanismu můžete danému identifikátoru přiřadit vlastní slovní vyjádření a/nebo usnadnit si případné vyhledávání. Pokud je použití CodeBooku tímto nastavením zakázáno (atribut *enabled*), budou místo identifikátoru posílána rovnou textová vyjádření z těchto tabulek.

V této větvi také můžete určit, které konkrétní tabulky chcete použít přidáním další podvětvě s názvem dané tabulky a opět atributem *enabled*. Ve výchozím stavu jsou všechny tabulky povoleny.

Tyto tabulky nejsou součástí tohoto dokumentu, protože se mohou lišit a měnit pro konkrétní instalaci aplikace. Kompletní tabulky (neboli CodeBook) lze vyžádat pomocí metody „check“. Není doporučeno vyžadovat zasílání tabulek při každém importu, neboť by docházelo ke zbytečné zátěži přenosu. Není vyloučeno, že v některé z následujících verzí se toto chování změní a CodeBook bude zasílán automaticky jen při změně.

brokers

Větev **brokers** obsahuje *enabled*, který určuje, zda budou volány metody pro práci se seznamem makléřů (*checkBroker*, *sendBroker*). Pokud je povoleno, tak atribut *enabled* ve větvi **photos** určuje, zda budou zasílány také fotografie makléřů. U těchto fotografií lze určit, zda se mají ponechat v originální podobě (poměr stran jako u pasové fotografie) nebo můžete zadat velikost vlastní ve formátu šířka x výška (např. 20x50).

photos

Následuje větev **photos**, pomocí které povolíte (opět atribut *enabled*) zasílání fotografií (metody *checkPhotos*, *sendPhoto*, *deletePhoto*), pokud u nabídky nějaké existují. Atributem *maxcount* můžete omezit maximální množství zaslaných fotografií, které budou respektovat makléřem určené pořadí a zbytek bude ignorován.

Uvnitř větve naleznete prostor pro zaslání obrázku, který bude přidán do fotografií jako **watermark**. Pro průhledné pozadí použijte barvu #F0F0F0 (RGB: 127,127,127). Velikost obrázku nesmí přesáhnout rozměr 640x480. Obrázek bude umístěn uprostřed v dolní části (toto chování bude v některé z dalších verzí nastavitelné). Při nezadání watermarku se pro požadované fotografie (viz. další odstavec) použije výchozí logo aplikace MultiEstate.

Dále pomocí jednotlivých větví **picture** určujete různé formáty od každého

obrázku, které se mají odesílat. Maximum jsou tři odlišné formáty.

V součastnosti můžete ovlivnit velikost (atribut *size*), která nesmí přesáhnout rozměr 640x480, použití watermarku (atribut *usewatermark*) a grafický formát výsledného obrázku (atribut *type*). Povolené jsou hodnoty *jpeg*, *gif*, *png*. Výchozí a doporučený je JPEG formát.

2.2 DTD specifikace

```
<!DOCTYPE econfig [
<!ELEMENT version (#PCDATA)>
<!ELEMENT public_url (#PCDATA)>
<!ELEMENT freeitems EMPTY>
<!ATTLIST freeitems enabled CDATA #REQUIRED>
<!ELEMENT offer (public_url?,freeitems?)>
<!ELEMENT class (#PCDATA)>
<!ELEMENT trans (#PCDATA)>
<!ELEMENT stage (#PCDATA)>
<!ELEMENT priceunit (#PCDATA)>
<!ELEMENT currency (#PCDATA)>
<!ELEMENT contract (#PCDATA)>
<!ELEMENT freeitem (#PCDATA)>
<!ELEMENT foptions (#PCDATA)>
<!ELEMENT codebook
(class?,trans?,stage?,priceunit?,currency?,contract?,freeitem?,foptions?)>
<!ATTLIST codebook enabled CDATA #REQUIRED>
<!ELEMENT photos EMPTY>
<!ATTLIST photos
  enabled (true|false) #REQUIRED
  size CDATA #REQUIRED>
<!ELEMENT brokers (photos)>
<!ATTLIST brokers enabled CDATA #REQUIRED>
<!ELEMENT watermark (#PCDATA)>
<!ELEMENT picture EMPTY>
<!ATTLIST picture
  size CDATA #REQUIRED
  type (jpeg|gif|png) #REQUIRED
  usewatermark (true|false) #REQUIRED>
<!ELEMENT pictures (watermark,picture+)>
<!ATTLIST pictures
  enabled (true|false) #REQUIRED
  maxcount CDATA #REQUIRED>
<!ELEMENT econfig (version,offer,codebook?,brokers?,pictures?)>
]>
```

3. Metody

Metody slouží pro samotné zajištění komunikace a výměnu dat. Export je inicializován na požadavek uživatele aplikace MultiEstate. Aplikace následně volá metody serveru, které vrací výsledek dané operace a tím lze do jisté míry ovlivnit volání dalších metod. Větší část způsobu volání metod je však dána konfigurací rozhraní (viz výše).

Všechny metody, které mezi vstupními parametry mají obsažen autentizační klíč v proměnné *AuthKey*, by měli zajistit ověření tohoto klíče přednostně před

jakoukoli operací, aby se zabránilo zneužití rozhraní neoprávněnou osobou.

3.1 Datové typy

string – Řetězcová hodnota, v hranaté závorce za tímto typem je uvedena maximální délka tohoto parametru. Pokud je tento typ následován znakem dvojtečky, tak slovo za dvojtečkou označuje tabulku z CodeBooku (viz. výše), která bude použita (pokud je povolena).

integer – Celočíselná absolutní hodnota (max. 2^{32}).

boolean – Logická hodnota true/false.

struct – Strukturovaná hodnota nebo také asociativní pole, kde klíče tohoto pole představují název parametru a hodnotou je pak konkrétní hodnota tohoto parametru určitého datového typu (může být také struct).

array – Podobně jako struct (ale pro klíče) je použita zpravidla číselná řada začínající od 0. Tato čísla nemají žádný význam, jde pouze o kolekci podobných dat.

base64 – Většinou binární nebo i jiná data, u kterých je nutné zajistit přenos beze ztrát, zakódovaná pomocí kodování Base64.

3.2 Struktura návratové hodnoty

Každá metoda musí vrátet strukturovaný výstup (typ *struct*) obsahující vždy alespoň jeden klíč s názvem *StatusCode*. Ten obsahuje číselné označení výsledného stavu operace. Pro standardní operace jsou rezervovány kódy od 0 do 500 a jejich význam najdete v tabulce „Přehled návratových kódů“ níže. Druhým společným klíčem v odpovědi může být *StatusMessage*, obsahující textové vyjádření k výslednému kódu, který je následně zobrazen uživateli aplikace. Pro standardní kódy z tabulky není nutné tento výstupní parametr používat, ale je možné takto pozměnit text oznámení.

Pokud chcete z Vašeho importního rozhraní oznámit jiný druh události, můžete zvolit vlastní kód mimo rezervovaný rozsah. Pak je vhodné použít klíč *StatusMessage* s vysvětlením problému.

Další klíče návratové hodnoty se liší podle volané metody a jsou uvedeny u každé zvláště v popisech níže. Je třeba dodržet přesně jejich pojmenování včetně velikosti písmen a datový typ.

Příklad výsledku metody „check“

```
struct(  
  [StatusCode] = 200,  
  [ConfigTime] = 0,  
  [SendCodeBook] = false  
)
```

Přehled návratových kódů	
Code	Message
200	Operace byla úspěšná
300	Nesouhlasí autentizační klíč

3.3 Povinné metody

Následuje seznam a popis metod, které musí být na importním serveru implementovány. Při neexistenci libovolné z nich, dojde k vyvolání fatální chyby a export nebude dokončen.

check

Základní metoda pro ověření dostupnosti spojení se serverem a aktuálnosti konfigurace. Je volána na začátku každého exportu a pokud nevrátí platný výsledek do určitého časového limitu, je server považován za nedostupný a export skončí s chybou.

Vstupním parametrem *InterfaceVersion* je číslo poslední verze exportního rozhraní. Tento parametr má informativní charakter. V případě, že se liší Vaše používaná verze a hodnota tohoto parametru, můžete si u poskytovatele aplikace MultiEstate vyžádat aktualizovaný popis rozhraní.

Pokud výstupní hodnota *ConfigTime* je rovna 0, bude pro export použita výchozí konfigurace (viz. výše) a nebude volána metoda *getConfig* pro získání konfiguračního souboru.

Druhá návratová hodnota *SendCodeBook* informuje aplikaci, aby zavolala metodu *sendCodeBook*, která na server odešle aktuální seznamy všech povolených číselníků (více informací viz. konfigurační větěv *codebook*).

VSTUPNÍ PARAMETRY:

InterfaceVersion *string[10]* – Označení verze exportního rozhraní

VÝSTUPNÍ HODNOTY:

ConfigTime *integer* – UNIX timestamp času poslední změny konfiguračního XML

SendCodeBook *boolean* – vyžádat si od aplikace seznam aktuálních dostupných číselníků

getConfig

Metoda pro přenos konfiguračního XML souboru do MultiEstate. Po přijetí je konfigurace zkontrolována pomocí DTD specifikace a pokud něco nesouhlasí, je zavolána metoda „reportError“ s popisem problému.

VSTUPNÍ PARAMETRY:AuthKey *string*[100] – Autentizační klíč k přidělenému profilu**VÝSTUPNÍ HODNOTY:**ConfigData *base64* – Konfigurační XML soubor encodovaný pomocí Base64**sendOffer**

Zaslání základních údajů o nabídce na server. Tato metoda obsahuje fixní data, která se mohou měnit pouze s dalšími verzemi rozhraní, ale nelze je žádným způsobem konfigurovat (kromě povolení/zákazu CodeBooku). Každá nabídka je identifikována unikátním kódem v rámci realitní kanceláře. Pokud už nabídka s tímto kódem na serveru existuje, dojde k aktualizaci dat, v opačném případě bude přidána. Parametry uvedené v tabulkách níže jsou povinné a budou vždy vyplněné.

VSTUPNÍ PARAMETRY:AuthKey *string*[100] – Autentizační klíč k přidělenému profiluCode *string*[10] – Unikátní kód nabídky v rámci RKOfferData *struct* – Základní informace o nabídce ve strukturovaném poliLocation *struct* – Informace o umístění nemovitosti**VÝSTUPNÍ HODNOTY:**OfferId *integer* – Unikátní číselné Id pod kterým je nabídka uložena na serveru nebo 0**Struktura parametru OfferData**

OfferData struct (
Name	string[1000]	Název nabídky
Description	string[1000]	Popis nabídky
Class	string[255]:class	Zatřídění nemovitosti
Transaction	string[255]:trans	Typ transakce (prodej, pronájem...)
Stage	string[255]:stage	Stav nabídky
Price	string[20]	Celková cena nabídky
PriceUnit	string[255]:priceunit	Jednotka ceny (m ² , objekt...)
Currency	string[5]:currency	Měna (CZK, USD, EUR...)
TotalArea	integer	Celková plocha nemovitosti
Contract	string[255]:contract	Typ smlouvy (exklusivní...)
ContractFrom	integer	UNIX timestamp zač. platnosti smlouvy
ContractTo	integer	UNIX timestamp kon. platnosti smlouvy
FreeDate	string[255]	Odkdy je nabídka k dispozici
HotOffer	boolean	Příznak označující zajímavou nabídku
Broker	integer	Identifikátor makléře
)		

Pro popsání umístění nemovitosti je využíván veřejný číselník UIR-ADR, který je udržován Ministerstvem práce a sociálních věcí. Veškeré informace o tomto číselníku najdete na adrese <http://forms.mpsv.cz/uir/>. Exportní rozhraní posílá vždy veškeré dostupné informace o umístění nemovitosti (kromě popisných čísel). Pokud nechcete použít UIR-ADR číselníky, tak kódy jednoduše ignorujte

a používejte textovou podobu jednotlivých částí lokality.

Struktura parametru Location		
Location struct (
RegionId	integer	Kód okresu z číselníku UIR-ADR
RegionName	string[255]	Jméno okresu
CityId	integer	Kód města z číselníku UIR-ADR
CityName	string[255]	Název města
DistrictId	integer	Kód části obce z číselníku UIR-ADR
DistrictName	string[255]	Název části obce
CityPartId	integer	Kód městské části z číselníku UIR-ADR
CityPartName	string[255]	Název městské části
StreetId	integer	Kód ulice z číselníku UIR-ADR
StreetName	string[255]	Název ulice
Cadastral	string[255]	Název katastrálního území
)		

deleteOffer

Metoda pro odstranění nabídky ze serveru neboli informování serveru o tom, že nabídka již není aktuální a neměla by se nikde prezentovat. Pokud jsou data pouze archivována, nemusí tato metoda nic provést, ale musí být implementována a vrátit úspěšnost operace.

VSTUPNÍ PARAMETRY:

AuthKey *string*[100] – Autentizační klíč k přidělenému profilu

Code *string*[10] – Unikátní kód nabídky v rámci RK

reportError

Ohlášení problémů s importním rozhraním. Pokud je problém detekovatelný automaticky, zasílá tato metoda textový popis problému. V opačném případě pouze obecnou informaci o výskytu problému. Je ve Vašem zájmu, abyste tyto chyby zachytávali a zaznamenali pro pozdější opravu. Pokud máte jistotu, že jste dodrželi instrukce uvedené v tomto dokumentu a systém stále hlásí problém, obraťte se na naše Centrum technické pomoci na adrese support@multiestate.cz spolu s podrobným popisem problému.

VSTUPNÍ PARAMETRY:

string ErrorMessage – Textový popis problému

3.4 Nepovinné metody

Existence těchto metod v serverové implementaci není vyžadována, neboť jejich volání je podmíněno konkrétním nastavením konfigurace. Neexistence metody na serveru může vyvolat pouze upozornění a export bude i přesto

dokončen úspěšně.

checkBroker

Metoda pro ověření existence makléře na serveru a jeho aktuálnosti. Volání této metody je podmíněno povolením exportu makléřů dle konfigurace. V případě, že makléř není na serveru aktuální nebo vůbec neexistuje, zavolá se metoda sendBroker.

VSTUPNÍ PARAMETRY:

AuthKey *string*[100] – Autentizační klíč k přidělenému profilu

BrokerId *integer* – Unikátní ID makléře v rámci aplikace

VÝSTUPNÍ HODNOTY:

ChangeTime *integer* – UNIX timestamp poslední změny makléře, 0 pokud makléř neexistuje

sendBroker

Metoda pro přidání nebo aktualizaci makléře na serveru. Volání této metody je podmíněno jednak povolením zasílání makléřů a také, zda je makléř na serveru aktuální. Každý makléř je identifikován unikátním číslem v rámci aplikace. Pod stejným číslem budou zasílány i případné aktualizace makléře. Toto číslo také slouží pro spojení nabídky s nabízejícím makléřem (parametr Broker v poli OfferData u metody sendOffer).

VSTUPNÍ PARAMETRY:

AuthKey *string*[100] – Autentizační klíč k přidělenému profilu

BrokerId *integer* – Unikátní ID makléře v rámci aplikace

FirstName *string*[255] – Křestní jméno makléře

LastName *string*[255] – Příjmení makléře

Title *string*[32] – Titul makléře

Phone *string*[32] – Primární telefonní spojení na makléře

PhoneOther *string*[32] – Další telefonní spojení na makléře

Email *string*[255] – E-mailová adresa makléře

PhotoData *base64* – Fotografie makléře ve formátu dle konfigurace

sendFreeltems

Metoda pro zaslání doplňujících informací o nabídce. V současné verzi jsou zasílány veškeré dostupné informace.

VSTUPNÍ PARAMETRY:

AuthKey *string*[100] – Autentizační klíč k přidělenému profilu

OfferCode *string[10]* – Unikátní kód nabídky v rámci RK

Items *struct* – Zaslané doplňující informace

Struktura vstupního parametru Items

```
Items struct (
  ItemId array (      string[255]:freeitem  Identifikátor položky v klíči
    value             string[255]:foptions  Seznam hodnot
  )
  ...
)
```

Identifikátor položky je unikátní řetězcová konstanta složená z malých písmen anglické abecedy, popřípadě čísel. Pro oddělení více slov je použita pomlčka. Pole hodnot u každé položky může obsahovat 1 a více hodnot. Pokud je hodnot více, tak platí, že první hodnota (na indexu 0) je označena uživatelem aplikace jako primární, ostatní jsou spíše doplňující. Každá jednotlivá hodnota je typu *string[255]*.

Příklad vstupního parametru Items (při použití CodeBooku)

```
struct (
  stav-budovy = array
    dobry,
    po-rekonstrukci
),
vybavenost = array(
  popis občanské vybavenosti
)
)
```

V tomto konkrétním případě je neprve vhodné ověřit, zda položka existuje v CodeBooku *freeitem*. Pokud neexistuje, jde o skalární položku a hodnota na indexu 0 v poli pod touto položkou je zároveň konečnou hodnotou (obvykle řetězec nebo číslo zadávané uživatelem). V opačném případě je nutné přes CodeBook *foptions* dohledat slovní vyjádření pro danou volbu.

Příklad vstupního parametru Items (bez CodeBooku)

```
struct (
  Stav budovy = array
    Dobrý,
    Po rekonstrukci
),
vybavenost = array(
  Popis občanské vybavenosti
)
)
```

Velmi podobné jako předchozí příklad, ale hodnoty není potřeba vyhledávat v CodeBooku.

checkPhotos

Metoda pro načtení seznamu všech exportovaných fotografií ke konkrétní nabídce. Pomocí této metody aplikace ověřuje jaké fotografie je potřeba na server odeslat, aktualizovat či smazat.

VSTUPNÍ PARAMETRY:

AuthKey *string*[100] – Autentizační klíč k přidělenému profilu

OfferCode *string*[10] – Unikátní kód nabídky v rámci RK

VÝSTUPNÍ HODNOTY:

Photos *struct* – Pole s informacemi o fotografiích spojených s nabídkou

Struktura výstupní hodnoty Photos

<pre>Photos struct (PhotoId struct (Position integer, LastChange integer) ...)</pre>	<p>Pro každou fotku jeden blok Poslední zaznamenaná pozice fotky UNIX timestamp poslední zmeny</p>
--	--

sendPhoto

Metoda pro zaslání jedné fotografie ve všech požadovaných formátech včetně jejího propojení na nabídku. Binární data fotografie jsou posílána pouze při prvním importu nebo pokud dojde ke změně formátů v konfiguraci. Při dalších aktualizacích se může změnit název fotografie a její pozice.

VSTUPNÍ PARAMETRY:

AuthKey *string*[100] – Autentizační klíč k přidělenému profilu

OfferCode *string*[10] – Unikátní kód nabídky v rámci RK

PhotoId *integer* – Unikátní číselné označení fotografie

Title *string*[255] – Název fotografie

Position *integer* – Pořadové číslo fotografie, hlavní fotografie má hodnotu 0

PhotoData *struct* – Indexované pole s binárními daty jednotlivých formátů fotografií (klíč 0 = první formát, klíč 1 = druhý formát, ...) dle konfigurace

deletePhoto

Metoda pro odstranění již neaktuální fotografie z nabídky.

VSTUPNÍ PARAMETRY:

AuthKey *string*[100] – Autentizační klíč k přidělenému profilu

PhotoId *integer* – Unikátní číselné označení fotografie

sendCodeBook

Metoda zajišťující zaslání aktuálních CodeBooku. Tato metoda je volána pouze pokud je to vyžádáno metodou *check*. Jsou zasílány pouze CodeBooky, které jsou povoleny přes konfiguraci.

VSTUPNÍ PARAMETRY:

AuthKey *string*[100] – Autentizační klíč k přidělenému profilu

CodeBook *struct* – Strukturované pole s jednotlivými tabulkami CodeBooku

Pole CodeBook obsahuje jednotlivé tabulky. Klíčem je vždy název tabulky a pod tímto klíčem je obsaženo další strukturované pole, kde klíčem již jsou jednotlivé identifikátory a hodnotami jejich slovní vyjádření.