

JSON API pro zjišťování cen MtG karet

Autor: Ing. Jiří Bažant
Verze: 1.0
Datum: 20.9.2014

Changelog

Verze	Datum	Autor	Poznámka
1.0	17.9.2014	Ing. Jiří Bažant	
	20.9.2014	Ing. Jiří Bažant	Oprava příkladu SHA-1 hashe a jeho použití.

Obsah

Changelog.....	1
Úvod.....	3
Bezpečnost.....	3
Volání rozhraní.....	3
Hlavičky odpovědi.....	3
Tělo odpovědi.....	3
Metody.....	4
shops-available.....	4
Struktura požadavku.....	4
Příklad požadavku.....	4
Příklad OK odpovědi.....	5
Příklad chybové odpovědi.....	5
find-price.....	6
Struktura požadavku.....	6
Sestavení podpisu požadavku.....	7
Příklad požadavku.....	8
Příklad OK odpovědi.....	8
Příklad chybové odpovědi.....	9
Limity volání.....	10
Testovací účet.....	10
Přepnutí do ostrého provozu.....	10

Úvod

API umožňuje třetí straně zjistit cenu jednotlivých MtG karet v českých internetových obchodech.

Rozhraní implementuje klient-server architekturu, kdy jako server vystupuje portál mtgim.cz. Klient pro komunikaci se serverem používá zasílání JSON požadavků.

Rozhraní implementuje dvě metody, které jsou popsány dále v tomto dokumentu.

Bezpečnost

Všechna volání API metod musí obsahovat parametr *AccessKey*. Některé metody dále vyžadují podpis požadavku pomocí parametru *Signature*. *AccessKey* slouží k identifikaci uživatele v rámci systému. *Signature* slouží ke kontrole původu požadavku.

Signature je hash získaný algoritmem *SHA-1*, jehož vstupem je zřetězení hodnot *AccessKey*, *Secret*, názvu metody a všech hodnot požadavku. Přesný postup sestavení *Signature* je popsán v sekci „Sestavení podpisu požadavku“ u jednotlivých metod, které podpis vyžadují.

Hodnoty *AccessKey* a *Secret* jsou uživateli předány při zřízení účtu. Jsou pro každého uživatele unikátní a jsou vázány na IP, nebo rozsah IP uvedený při podpisu smlouvy.

Volání rozhraní

JSON Server se nachází na adrese <http://mtgim.cz/mtg-api>.

Volání rozhraní se provádí pomocí POST požadavku, který v těle požadavku obsahuje JSON objekt s parametry požadavku.

Odpověď od serveru v hlavičce obsahuje návratový kód a v těle JSON objekt s odpovědí.

Hlavičky odpovědi

Návratové kódy jsou řízeny HTTP standartem. Tzn. 200 identifikuje validní odpověď. V těle odpovědi jsou poté požadovaná data. Všechny ostatní kódy znamenají chybový stav.

Chybové stavy 4XX jsou vyhrazeny pro chyby klienta (neplatná metoda, chybný podpis požadavku, ...).

Chyby 5XX jsou vyhrazeny pro chyby na straně serveru (rozhraní nedostupné, chybná odpověď od e-shopu, ...).

Tělo odpovědi

Tělo odpovědi je vždy tvořeno JSON objektem. V případě návratového kódu 200 obsahuje tělo požadovaná data. Konkrétní tvar odpovědi je závislý na volané metodě.

V případě chybové odpovědi tělo obsahuje položku *error* s doplňujícím popisem chyby.

Metody

shops-available

Metoda slouží pro získání obchodů, pro které je možné se dotazovat na cenu karty. Vrací seznam klíč-hodnota párů, kde klíč je id obchodu a hodnota je český název obchodu.

ID obchodu je povinný parametr metody find-price.

Tento požadavek není nutné podepisovat. Výsledek volání je možné dlouhodobě cachovat.

Struktura požadavku

Adresa: /shops-available

Struktura požadavku:

Název	Typ	Popis	Příklad
AccessKey	String	Unikátní identifikátor přidělený při vytvoření účtu.	testKey

Struktura odpovědi:

Název	Typ	Popis	Příklad
data	Object	Objekt obsahující klíč-hodnota dvojice s názvy obchodů. (Viz výše)	{ 'cernyrytir': 'Černý rytíř', 'rishada': 'Rishada' }

Příklad požadavku

```
curl -H 'Accept: application/json' -H 'Content-type: application/json' -X POST  
-d '{"AccessKey": "testKey"}' http://mtgim.cz/mtg-api/shops-available
```

```
> POST /mtg-api/shops-available HTTP/1.1  
> User-Agent: curl/7.37.0  
> Host: mtgim.cz  
> Accept: application/json  
> Content-type: application/json  
> Content-Length: 24  
>  
> {"AccessKey": "testKey"}
```

Příklad OK odpovědi

```
< HTTP/1.1 200 OK
< Date: Tue, 16 Sep 2014 22:29:19 GMT
< Server: Apache/2.4.9 (Unix) PHP/5.5.13
< X-Powered-By: PHP/5.5.13
< Content-Length: 123
< Content-Type: application/json
<
< {"data":{"cernyrytir":"\u010cern\u00fd Ryt\u00edd\u0159","mystic":"Mystic
Shop","najada":"Naj\u00e1da","rishada":"Rishada"}}
```

Příklad chybové odpovědi

```
< HTTP/1.1 400 Bad Request
< Date: Tue, 16 Sep 2014 22:29:19 GMT
< Server: Apache/2.4.9 (Unix) PHP/5.5.13
< X-Powered-By: PHP/5.5.13
< Content-Length: 37
< Content-Type: application/json
<
< {"error": "Invalid params specified"}
```

find-price

Metoda slouží k dohledání cen karet v daném obchodě. Vstupem je hledaný řetězec z názvu karty, ID obchodu, ve kterém se má cena hledat, a provedení karty (obyčejné / foil).

Výsledkem je seznam karet obsahujících daný řetězec ve svém názvu a jejich cena v daném obchodě. Maximální počet vrácených položek je závislý na e-shopu. Obvykle se pohybuje mezi 30 – 50 kartami.

Struktura požadavku

Adresa: /find-price

Struktura požadavku:

Název	Typ	Popis	Příklad
AccessKey	String	Unikátní identifikátor přidělený při vytvoření účtu.	testKey
CardName	String	Hledaný řetězec z názvu karty.	Disenchant
Shop	String	Identifikátor obchodu, ve kterém se má cena vyhledávat.	Rishada
FoilType	String	Identifikátor typu karty: R – pouze obyčejné karty F – pouze foilové karty A – všechny karty	R

Struktura odpovědi:

Název	Typ	Popis	Příklad
data	Object	Pole obsahující objekty s jednotlivými výsledky vyhledávání.	
	Object	Struktura jednoho záznamu.	
name	String	Název karty.	Disenchant
expansion	String	Název edice karty.	Mercedian Masques
amount	Number	Počet karet skladem.	4
value	Number	Cena karty v Kč.	15
quality	String	Kvalita karty. V případě nearmint nebo mint non-foil karty je pole prázdné.	foil, lightly played

Sestavení podpisu požadavku

Každý požadavek na cenu karty musí být podepsán. Podpis je *SHA-1* hash ze zřetězení *AccessKey*, *Secret*, názvu metody a všech hodnot požadavku. Tedy:

```
sha1(AccessKey + Secret + 'find-price' + CardName + Shop + FoilType)
```

Příklad sestavení řetězce:

AccessKey: *testKey*

Secret: *testSecret*

CardName: *Disenchant*

Shop: *rishada*

FoilType: *R*

Sestavený řetězec: *fdfc0f4016a5d24ede15d610a7598c46e0d26a8a*

SHA-1: *6a7c77f1401063992cf47fcadf5b16519c87a14c*

Příklad implementace v PHP5:

```
function getRequestFindPriceBody (
    $accessKey, $secret, $cardName, $shop, $foilType
) {
    $signature = getFindPriceSignature(
        $accessKey, $secret, $cardName, $shop, $foilType
    );
    $req_params = array(
        'AccessKey' => $accessKey,
        'CardName' => $cardName,
        'Shop' => $shop,
        'FoilType' => $foilType,
        'Signature' => $signature,
    );
    return json_encode($req_params);
}

function getFindPriceSignature (
    $accessKey, $secret, $cardName, $shop, $foilType
) {
    return sha1(
```



```

        $accessKey
        . $secret
        . 'find-price'
        . $cardName
        . $shop
        . $foilType
    );
}

```

Příklad požadavku

```

curl -H 'Accept: application/json' -H 'Content-type: application/json' -X POST
-d '{"AccessKey": "testKey", "CardName": "Disenchant", "Shop": "rishada",
"FoilType": "R", "Signature": "fdfc0f4016a5d24ede15d610a7598c46e0d26a8a"}'
http://mtgim.cz/mtg-api/find-price

```

```

> POST /mtg-api/find-price HTTP/1.1
> User-Agent: curl/7.37.0
> Host: mtgim.cz
> Accept: application/json
> Content-type: application/json
> Content-Length: 143
>
> {"AccessKey": "testKey", "CardName": "Disenchant", "Shop": "rishada",
"FoilType": "R", "Signature": "fdfc0f4016a5d24ede15d610a7598c46e0d26a8a"}

```

Příklad OK odpovědi

```

< HTTP/1.1 200 OK
< Date: Tue, 16 Sep 2014 22:34:09 GMT
< Server: Apache/2.4.9 (Unix) PHP/5.5.13
< X-Powered-By: PHP/5.5.13
< Content-Length: 596
< Content-Type: application/json
<
< {"data":[{"name":"Testovací karticka","expansion":"Return to
Ravnica","amount":0,"value":0,"quality":""},{name":"Testovací
karticka","expansion":"Return to Ravnica","amount":1,"value":100,"quality":""},
{"name":"Testovací karticka","expansion":"Return to
Ravnica","amount":2,"value":200,"quality":""},{name":"Testovací
karticka","expansion":"Return to Ravnica","amount":3,"value":300,"quality":""},
{"name":"Testovací karticka","expansion":"Return to
Ravnica","amount":4,"value":400,"quality":""},{name":"Testovací
karticka","expansion":"Return to
Ravnica","amount":5,"value":500,"quality":""}]}

```

Příklad chybové odpovědi

```
< HTTP/1.1 401 Unauthorized
< Date: Tue, 16 Sep 2014 22:29:19 GMT
< Server: Apache/2.4.9 (Unix) PHP/5.5.13
< X-Powered-By: PHP/5.5.13
< Content-Length: 26
< Content-Type: application/json
<
< {"error": "Invalid signature"}
```

Limity volání

Počet volání rozhraní z jednoho účtu je omezen na 200 volání za den. V případě překročení této hranice může server na každé další volání odpovědět HTTP hlavičkou 403 a chybou „Day limit exceeded“.

```
< HTTP/1.1 403 Forbidden
< Date: Tue, 16 Sep 2014 22:29:19 GMT
< Server: Apache/2.4.9 (Unix) PHP/5.5.13
< X-Powered-By: PHP/5.5.13
< Content-Length: 31
< Content-Type: application/json
<
< {"error": "Day limit exceeded"}
```

Pokud by byl tento limit byl pro vaše použití příliš nízký, lze smluvně upravit jeho hodnotu.

Testovací účet

Pro testování by měl být využit testovací účet. Ten umožňuje libovolný počet volání rozhraní. Při použití tohoto účtu jsou vždy vrácena statická testovací data. (Nelze jej tedy použít pro koncové uživatele.)

Tento účet také není závislý na použitém eshopu a tudíž nedochází k neovlivnitelným výpadkům služby. Je ovšem třeba brát zřetel, že odpověď od ostrého eshopu bude trvat značně delší dobu. (Obvykle řádově v jednotkách sekund.)

Pro testovací účet použijte tyto údaje:

- *AccessKey*: testKey
- *Secret*: testSecret

Tyto údaje jsou také použity ve všech příkladech v této dokumentaci.

Přepnutí do ostrého provozu

Přepnutí do ostrého provozu se provádí po podpisu smlouvy a úspěšné implementaci komunikace přes testovací přístupy. V rámci smlouvy musíte uvést rozsah IP adres, ze kterých budete k API přistupovat v ostrém provozu. Od mtgim.cz vám budou následně zaslány ostré přístupové údaje (*AccessKey* a *Secret*) pro tento rozsah.