

Veřejné API verze 2.1

Změny oproti verzi 2.0 jsou vyznačeny oranžově.

Obecné informace

- Plug-in je realizován pomocí **jediného skriptu**, který musí být **spustitelný pomocí URL požadavku**
- Veškeré volání skriptů mezi centrální aplikací Backlinks Genius a plug-inem je prováděno pomocí **POST** požadavku, který vždy obsahuje parametr **function** a další parametry, které jsou popsány níže.
- Data se vyměňují ve formátu **JSON** obalená řetězcem `<!-- BGCODE -->` a `<!-- /BGCODE -->`
- Pokud váš webhosting nepodporuje funkci `json_encode`, můžete použít naši funkci `bsf_json_encode` pro PHP, která je plnohodnotnou náhradou.
- API je umístěno na adrese <https://www.backlinksgenius.com/Services/api.php>. Pozor, je třeba dodržet velká a malá písmena v URL.

Uživatelské rozhraní plug-inu

Uživatel potřebuje pouze **formulář pro registraci plug-inu**.

Vazby plug-in, doména, článek, uživatel

- Jeden uživatel má n domén
- Jedna doména má n článků
- Jeden uživatel má n plug-inů
- Neexistuje vazba mezi doménou a plug-inem. Jeden plug-in může obsluhovat více domén.

Přehled komunikace plug-inu a centrální aplikace

Předmětem komunikace jsou následující 4 typy přenosů:

1. **Jednorázová registrace** plug-inu v centrální aplikaci
2. Pravidelné **načítání seznamu článků** centrální aplikací
3. Načtení obsahu **konkrétního článku** centrální aplikací
4. Pravidelná **aktualizace obsahu článku** z centrální aplikace do plug-inu

1. Jednorázová registrace plug-inu v centrální aplikaci

1. Uživatel se zaregistruje do centrální aplikace Backlink Genius
2. Centrální aplikace vygeneruje kontrolní kod uživatele
3. Uživatel v centrální aplikaci založí doménu
4. Doménu schválí administrátor systému
5. Uživatel nainstaluje plug-in do svého publikačního systému
6. V rozhraní publikačního systému uživatel pomocí formuláře vyplní kontrolní kod a klikne na „připojit k Backlink Genius“. Formulář je popsán níže. Je nutné zaslat číslo verze 2, aby se plug-in zaregistroval pro pasivní verzi komunikace.
7. Po odeslání formuláře se provede test na existenci URL plug-inu a testovací zaslání dat.
8. API zobrazí informaci o úspěšné registraci popř. chybové hlášení.

Parametry POST požadavku pro registraci plug-inu

V tomto případě bude API voláno přímo pomocí POST požadavku z formuláře.

Dotaz nebude zabalen JSONem.

Parametr	Povinný	Typ	Popis	Hodnota
function	ano	string	Rozlišuje typ přenosu	inFirstConnect
url	ano	string	URL adresa vašeho skriptu	
accessHash	ano	string	Kod uživatele	
version	ano	float	Číslo verze API	2.1

API zobrazí informaci o úspěšné registraci popř. chybové hlášení.

Číslo verze API musí být 2.

URL adresa skriptu bude testována při registračním procesu.

Příklad POST požadavku

Array

```
(  
    [function] => inFirstConnect  
    [url] => http://www.mojedomena.com/plugin/test.php  
    [accesshash] => 4a6c81c19b5a75a7521264d6f856cdf4  
    [version] => 2.1  
)
```

Příklad formuláře

```
<form action="https://www.backlinksgenius.com/Services/api.php"
target="_blank" method="post">
  <label for="key">Backlinks Genius control code: </label>
  <input id="key" name="accessHash" type="text" value="" />
  <input type="hidden" name="function" value="inFirstConnect" />
  <input type="hidden" name="url"
value="<?="http://".$_SERVER['HTTP_HOST'].$_SERVER[PHP_SELF];?>" />
  <input type="hidden" name="version" value="2.1" />
  <input type="submit" name="submit" value="Register in Backlinks
Genius" />
</form>
```

2. Pravidelné načítání seznamu článků centrální aplikací

Centrální aplikace Backlinks Genius bude pravidelně volat plug-in s parametrem **function=„actualizeINList“**. Volání je prováděno 1x za hodinu. Seznam článků slouží k předání informace aplikaci Backlinks Genius o nových článcích a o změnách článků.

Parametry volání Backlinks Genius >> plug-in

Parametr	Povinný	Typ	Popis	Hodnota
function	ano	string	Rozlišuje typ přenosu	actualizeINList
controlCode	ano	String	Kontrolní kod, který obsahuje první 4 znaky z accessHash. Slouží pro identifikaci Centrální aplikace	
limit	ne	Integer	Slouží pro stránkování – definuje počet článků, které má plug-in vrátit na jedné stránce	
offset	ne	Integer	Slouží pro stránkování – definuje od kterého článku má začít výpis dané stránky. Číslo stránky je rovno offset/limit	

Příklad volání plug-inu

Array

```
(  
    [function] => actualizeINList  
    [controlCode] => 4a6c  
    [limit] => 500  
    [offset] => 2500  
)
```

Odpověď plug-inu

Plug-in vypíše stránkovaný seznam článků, které chcete zapojit do systému Backlinks Genius. Je zcela na vás, jaké podmínky uplatníte pro výběr zasílaných článků.

Pokud není zaslaný parametr limit a offset, doporučujeme poslat prvních 100 článků.

Parametr	Povinný	Typ	Popis	Hodnota
status	ano	numeric array		
status[0]	ano	integer	Stavový kod	200 = OK
status[1]	ne	string	Popis chyby	
version	ano	float	Číslo verze API	2.1
data	ano	numeric array	Seznam vašich článků	
data[i]	ano	associative	Detail konkrétního článku	
data[i][id]	ano	array	Vaše ID článku	
data[i][url]	ano	integer	URL článku	
data[i][title]	ano	string	Název článku	
data[i][created]	ano	datetime	Datum a čas vytvoření článku	
data[i][changed]	ano	datetime	Datum a čas poslední změny článku	

- Odpověď je provedena jednoduchým výpisem (v PHP pomocí příkazu **echo**)
- Odpověď je ve formátu **JSON** obalená řetězcem **<!-- BGCODE -->** a **<!-- /BGCODE -->**
- Články se posílají **seřazené dle ID vzestupně**
- Datum vytvoření článku a změny článku musí být ve formátu YYYY-MM-DD HH:MM:SS např. 2010-04-08 09:15:01

Příklad odpovědi plug-inu před zabalením do JSON

Array

```
(
  [status] => Array
  (
    [0] => 200
    [1] => OK
  )
  [version] => 2.1
  [data] => Array
  (
    [0] => Array
    (
      [id] => 14
      [url] => http://www.mojedomena.com/prvni-clanek
      [title] => První článek
      [created] => 2010-04-01 12:34:11
      [changed] => 2010-05-15 21:01:42
    )
    [1] => Array
    (
      [id] => 16
      [url] => http://www.mojedomena.com/druhý-clanek
      [title] => Druhý článek
      [created] => 2010-04-08 09:15:01
    )
  )
)
```

```
                [changed] => 2010-05-15 21:02:19
            )
        )
    )
```

Příklad výsledné odpovědi plug-inu

Po zabalení pomocí JSON a přidání počátečního a koncového řetězce.

```
<!-- BGCODE -->
{"version":"2.1","status":[200,"OK"],"data":[{"id":14,"url":"http://www.mo
jedomena.com/prvni-clanek","title":"První článek","created":"2010-04-01
12:34:11","changed":"2010-05-15
21:01:42"}, {"id":16,"url":"http://www.mojedomena.com/druhy-
clanek","title":"Druhý článek","created":"2010-04-08
09:15:01","changed":"2010-05-15 21:02:19"}]}<!-- /BGCODE -->
```

Příklad části skriptu v jazyku PHP:

// V poli \$results bude načtený seznam článků

```
$data = array();
$articlesCnt = count($results);
for($i = 0;$i<$articlesCnt;$i++){
    $data[] = array( "id" => $result[$i]->id,
        "url" => $result[$i]->url,
        "title" => $result[$i]->title,
        "created" => date('Y-m-d H:i:s',strtotime($result[$i]->created)),
        "changed" => date('Y-m-d H:i:s',strtotime($result[$i]->changed))
    );
}
$status = array(200,"OK");
$response =
    array(
        "version" => 2,
        "status" => $status,
        "data" => $data);

echo "<!-- BGCODE -->".bsf_json_encode($response)."<!-- /BGCODE -->";
```

3. Načtení obsahu konkrétního článku centrální aplikací

Centrální aplikace Backlinks Genius zavolá plug-in s parametrem **function=„actualizeINDetail“**. Načítání probíhá po té, co se načte seznam článků v bodu 2. Obsah článků se načítá pouze v případě nového nebo změněného článku. Změna článku se detekuje na základě zaslání data změny v bodu 2.

Parametry volání Backlinks Genius >> plug-in

Parametr	Povinný	Typ	Popis	Hodnota
function	ano	string	Rozlišuje typ přenosu	actualizeINDetail
article_id	ano	Integer	Vaše ID článku	
controlCode	ano	String	Kontrolní kod, který obsahuje první 4 znaky z accessHash. Slouží pro identifikaci Centrální aplikace	

Příklad volání plug-inu

Array

```
(  
    [function] => actualizeINDetail  
    [article_id] => 14  
    [controlCode] => 4a6c  
)
```

Odpověď plug-inu

Plug-in vypíše obsah článku a stavové hlášení.

Parametr	Povinný	Typ	Popis	Hodnota
status	ano	numeric array		
status[0]	ano	integer	Stavový kod	200 = OK 305 = ID článku nebylo zasláno 306 = Článek se zasláním ID v plug-inu už neexistuje
status[1]	ne	String	Popis chyby	
text	ano	String	Text článku	

- Odpověď je provedena jednoduchým výpisem (v PHP pomocí příkazu **echo**)
- Odpověď je ve formátu **JSON** obalena řetězcem **<!-- BGCODE -->** a **<!-- /BGCODE -->**

Příklad odpovědi plug-inu před zabalením do JSON

```
Array
(
    [status] => Array
        (
            [0] => 200
            [1] => OK
        )
    [text] => Obsah mého prvního článku
)
```

Příklad výsledné odpovědi plug-inu

Po zabalení pomocí JSON a přidání počátečního a koncového řetězce.

```
<!-- BGCODE -->{"status":[200,"OK"],"text":"Obsah mého prvního článku"}<!--
/BGCODE -->
```

Příklad části skriptu v jazyku PHP:

```
$text = '';
if ($_POST['article_id']){
    $article_id = $_POST['article_id'];

    // V proměnné $text bude načtený obsah článku

    if($text){
        $status = array(200,"OK");
    }else{
        $status = array(306,"Article does not exist");
    }
}else{
    $status = array(305,"Missing article_id");
}
$response =
    array(
        "status" => $status,
        "text" => $text);

echo "<!-- BGCODE -->".bsf_json_encode($response)."<!-- /BGCODE -->";
```

4. Pravidelná aktualizace obsahu článku z centrální aplikace do plug-inu

Centrální aplikace zavolá plug-in s parametrem **function=„actualizeOUT“** a pošle aktuální obsah článku.

V případě, že plug-in nebude dostupný, tato skutečnost se zaznamená v centrální aplikaci a cron při příštím spuštění provede opětovný request na actualizeOUT. ActualizeOUT posílá každý článek zvlášť, plug-in musí potvrdit přijetí článku, jinak zůstává v centrální aplikaci ve frontě.

Parametry volání Backlinks Genius >> plug-in

Parametr	Povinný	Typ	Popis	Hodnota
function	ano	string	Rozlišuje typ přenosu	actualizeOUT
controlCode	Ano	String	Kontrolní kod, který obsahuje první 4 znaky z accessHash. Slouží pro identifikaci Centrální aplikace	
article_id	Ano	Integer	ID článku	
text	Ano	String	Obsah článku	

Příklad volání plug-inu

Array

```
(  
    [function] => actualizeOUT  
    [article_id] => 14  
    [controlCode] => 4a6c  
    [text] => Obsah mého prvního <a href="http://domena.com">článku</a>  
)
```

Odpověď plug-inu

Plug-in uloží obsah článku a vypíše status.

Parametr	Povinný	Typ	Popis	Hodnota
status	ano	numeric array		
status[0]	ano	integer	Stavový kod	200 = OK 305 = chybějící zasláné ID článku nebo obsah, 310 = Chyba při ukládání článku na straně plug-inu
status[1]	ne	String	Popis chyby	

Příklad odpovědi plug-inu před zabalením do JSON

```
Array
(
    [status] => Array
        (
            [0] => 200
            [1] => OK
        )
)
```

Příklad výsledné odpovědi plug-inu

Po zabalení pomocí JSON a přidání počátečního a koncového řetězce.

```
<!-- BGCODE -->{"status":[200,"OK"]<!-- /BGCODE -->
```

Příklad části skriptu v jazyku PHP:

```
if ($_POST['article_id'] && $_POST['text']){
    $article_id = $_POST['article_id'];
    $text = $_POST['text'];

    // uloží obsah článku a naplní proměnnou $updateSuccess

    if ($updateSuccess){
        $status = array(200,"OK");
    }else{
        $status = array(310,"Plug-in error");
    }
}else{
    $status = array(305,"Missing article_id or text");
}
$response = array("status" => $status);
echo "<!-- BGCODE -->".bsf_json_encode($response)."<!-- /BGCODE -->";
```