

API Server-to-Server (FR)

Documentation technique pour l'intégration Server-to-Server (S2S) permettant de notifier Powerspace des conversions générées.

Informations clés :

- Base URL: `https://a.pwspace.com`
- Méthode: **GET**
- Format: Query string parameters
- Authentification: Click ID fourni par Powerspace

1. Aperçu

L'API Server-to-Server (S2S) de Powerspace permet aux annonceurs de notifier en temps réel les **conversions** générées suite à un clic publicitaire. Deux types de conversions sont supportés :

- **Lead** : Génération d'un prospect qualifié
- **Order** : Finalisation d'une commande avec valeur

Principe de fonctionnement

Lorsqu'un utilisateur clique sur votre publicité Powerspace, il est redirigé vers votre site avec un paramètre `clickId` unique. Ce paramètre doit être conservé et utilisé pour notifier les conversions via l'API S2S.

2. Flux de conversion

Étape 1 : Clic publicitaire

L'utilisateur clique sur votre publicité Powerspace et arrive sur votre site avec l'URL :

`https://votre-site.com/landing?clickId=abc123xyz789&utm_source=powerspace`

Étape 2 : Conservation du Click ID

Votre site doit **conserver le clickId** (cookie, session, base de données) pour l'associer aux actions de l'utilisateur.

Étape 3 : Notification de conversion

Lorsqu'une conversion se produit, appelez l'API S2S correspondante avec le clickId conservé.


3. Endpoints disponibles

Lead Notification

GET `https://a.pwspace.com/ld?qci={clickId}`

Notifie la génération d'un **lead** (prospect qualifié).

Paramètres :

| Nom | Type | Obligatoire | Description |
|-----|--------|---|---|
| qci | string |  Oui | Click ID fourni par Powerspace lors du clic initial |

Order Notification

GET `https://a.pwspace.com/ord?qci={clickId}&ov={order-value}`

Notifie la finalisation d'une **commande** avec sa valeur.

Paramètres :

| Nom | Type | Obligatoire | Description |
|-----|--------|--|---|
| qci | string |  Oui | Click ID fourni par Powerspace lors du clic initial |
| ov | float |  Recommandé | Valeur de la commande (ex: 49.99, 125.50) |

Importance de la valeur de commande

Bien que facultatif, le paramètre `ov` (order value) est **fortement recommandé**. Il permet de remonter la valeur dans vos statistiques Powerspace et d'optimiser le pilotage de vos campagnes publicitaires.

4. Exemples d'implémentation

Scénario Lead

Bash

```
# L'utilisateur s'inscrit à votre newsletter
```

```
curl "https://a.pwspace.com/ld?qci=abc123xyz789"
```

Scénario Order

Bash

```
# L'utilisateur finalise un achat de 89.99€  
curl "https://a.pwspace.com/ord?qci=abc123xyz789&ov=89.99"
```

PHP - Gestion complète

PHP

```
<?php  
// 1. Récupération du clickId à l'arrivée sur le site  
if (isset($_GET['clickId'])) {  
    $_SESSION['powerspace_click_id'] = $_GET['clickId'];  
}  
  
// 2. Notification d'un lead  
function notifyLead($clickId) {  
    $url = "https://a.pwspace.com/ld?qci=" . urlencode($clickId);  
    file_get_contents($url);  
}  
  
// 3. Notification d'une commande  
function notifyOrder($clickId, $orderValue) {  
    $url = "https://a.pwspace.com/ord?qci=" . urlencode($clickId) . "&ov=" .  
$orderValue;  
    file_get_contents($url);  
}  
  
// Exemple d'utilisation  
if (isset($_SESSION['powerspace_click_id'])) {  
    // Lead généré  
    notifyLead($_SESSION['powerspace_click_id']);  
  
    // Ou commande finalisée  
    notifyOrder($_SESSION['powerspace_click_id'], 129.99);  
}  
?>
```

Python - Intégration serveur

Python


```
import requests  
from urllib.parse import urlencode  
  
def notify_lead(click_id):  
    """Notifie un lead généré"""
```

```
url = f"https://a.pwspace.com/ld?qci={click_id}"
try:
    response = requests.get(url, timeout=5)
    return response.status_code == 200
except requests.RequestException:
    return False

def notify_order(click_id, order_value):
    """Notifie une commande finalisée"""
    params = {'qci': click_id, 'ov': order_value}
    url = f"https://a.pwspace.com/ord?{urlencode(params)}"
    try:
        response = requests.get(url, timeout=5)
        return response.status_code == 200
    except requests.RequestException:
        return False

# Exemple d'utilisation
click_id = "abc123xyz789"
notify_lead(click_id)
notify_order(click_id, 89.99)
```

5. Codes de réponse

| Code | Signification | Action |
|---|---------------|------------------------------------|
| :large_green_circle: 200 | OK | Conversion enregistrée avec succès |
| :large_yellow_circle: 400 | Bad Request | Click ID manquant ou invalide |
| :large_yellow_circle: 404 | Not Found | Click ID expiré ou inexistant |
|  5xx | Server Error | Réessayer plus tard |

6. Bonnes pratiques

Gestion du Click ID

- Conservez le clickId dès l'arrivée de l'utilisateur
- Associez-le à la session ou au profil utilisateur
- Persistez-le en base de données pour les conversions différées
- Nettoyez les anciens clickIds (expiration ~30 jours)

Appels API

- Appelez immédiatement après la conversion
- Gérez les timeouts (5 secondes max recommandé)
- Ne pas bloquer l'expérience utilisateur en cas d'échec

- Loggez les appels pour le debugging

Valeur de commande

- Envoyez toujours la valeur réelle de la commande
 - Utilisez le format décimal (ex: 49.99, pas 4999)
 - Excluez les frais de port si demandé
 - Incluez les taxes selon votre configuration
-

7. Cas d'usage avancés

Conversions multiples

Un même clickId peut générer plusieurs conversions :

Bash

```
# D'abord un lead
curl "https://a.pwspace.com/ld?qci=abc123xyz789"

# Puis une commande (même clickId)
curl "https://a.pwspace.com/ord?qci=abc123xyz789&ov=159.99"
```

Attribution différée

Pour les conversions qui arrivent plusieurs jours après le clic :

PHP

```
// Stockage en base lors du clic
$pdo->prepare("INSERT INTO user_attribution (user_id, click_id, created_at) VALUES
(?, ?, NOW())")
->execute([$userId, $clickId]);

// Récupération lors de la conversion
$stmt = $pdo->prepare("SELECT click_id FROM user_attribution WHERE user_id = ?");
$stmt->execute([$userId]);
$clickId = $stmt->fetchColumn();

if ($clickId) {
    notifyOrder($clickId, $orderValue);
}
```

8. Debugging et monitoring

Vérification des appels

- Loggez tous les appels S2S avec timestamp

- **Surveillez** le taux de succès des appels
- **Alertez** sur les échecs répétés

Tests en développement

Bash

```
# Test avec un clickId fictif
curl "https://a.pwspace.com/ld?qci=test_click_id_123"
curl "https://a.pwspace.com/ord?qci=test_click_id_123&ov=99.99"
```

9. Check-list d'intégration

- ☒ Récupération et stockage du clickId depuis l'URL
- ☒ Conservation du clickId pendant toute la session utilisateur
- ☒ Appel lead lors de la génération d'un prospect
- ☒ Appel order avec valeur lors d'une commande
- ☒ Gestion des timeouts et erreurs réseau
- ☒ Logging des appels pour le monitoring
- ☒ Tests en environnement de développement
- ☒ Validation en production avec des conversions réelles

Support technique

Pour toute question sur l'intégration S2S :

- **Documentation** : Cette page
- **Support** : Contactez votre chargé de compte Powerspace
- **Tests** : Utilisez des clickIds de test en développement



Updated about 1 month ago

← [API Guide \(EN\)](#)

[API Server-to-Server \(EN\)](#) →