

HELLOASSO

GUIDE D'UTILISATION

API V5

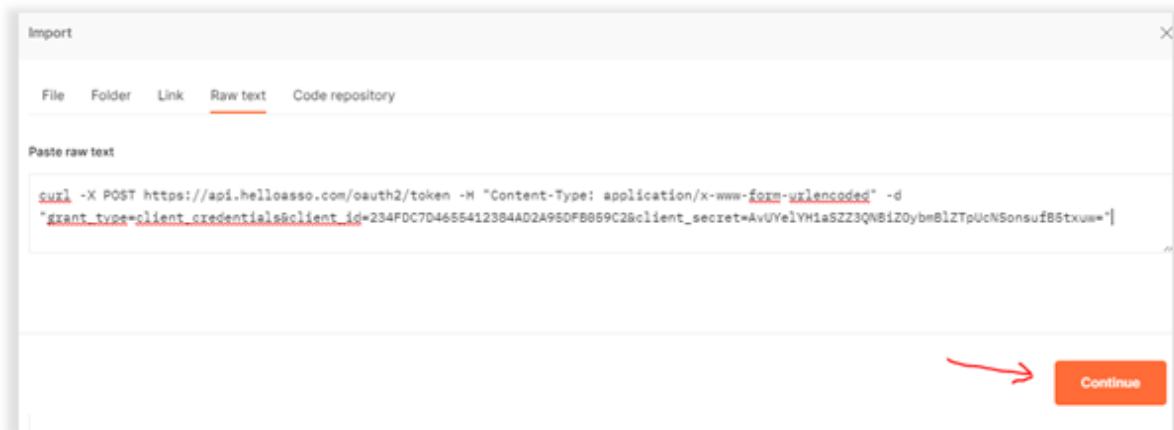
AVANT DE COMMENCER

Ce guide contiendra des exemples d'appels en cURL que vous pourrez tester sur Postman. Postman est un logiciel qui permet de tester et interroger des webservices et API.

Il est disponible gratuitement ici : <https://www.postman.com/downloads/>

Pour importer des appels cURL, cliquez sur :

File -> Import -> Raw text -> Coller le code cURL -> Continue



ENVIRONNEMENTS

Helloasso possède un environnement de test et de production.

Environnement de production :

Serveur authentification - <https://api.helloasso.com/oauth2>

Serveur API - <https://api.helloasso.com/v5>

Environnement de test :

Serveur authentification - <https://api.helloasso-rc.com/oauth2>

Serveur API - <https://api.helloasso-rc.com/v5>

AUTHENTIFICATION

Demander un access token

Pour pouvoir utiliser les points d'API HelloAsso vous devez d'abord obtenir un token d'accès en appelant le serveur d'authentification OAuth 2.0 : <https://api.helloasso.com/oauth2>

Pour obtenir un token d'accès (access_token), procédez à un appel POST sur l'URL <https://api.helloasso.com/oauth2/token>

Maintenant que vous disposez de l'`access_token`, vous allez pouvoir effectuer des appels API pendant 1799 secondes soit une demi-heure. Une fois ce laps de temps dépassé vous allez devoir rafraîchir votre `access_token` à l'aide du `refresh_token`.

Rafraichir un token :

Pour rafraîchir un access_token vous devez procéder à un appel POST sur l'URL <https://api.helloasso.com/oauth2/token>

Header :

content-type: application/x-www-form-urlencoded

Body :

client_id : your client id

refresh_token : your refresh_token

grant_type : refresh_token

Exemple cURL :

```
curl -X POST https://api.helloasso.com/oauth2/token -H "Content-Type: application/x-www-form-urlencoded" -d "grant_type=refresh_token&client_id=d063c18bf2624378889f4e5285876557&refresh_token=8A7dUtri6QNT3ZzdlPmrnkum9zOy8yMSZRJehwlKUaCNB2tOscg9HURq1PuZTLblzwmReowneTb1jLEa0mfsbsuLlinrLtO_OgmEaQ2r_OGaZi1Qkyxqlsxe42S5JhNMwKLD5Y-1YPXo7GXR5-ijs8mKdUXM9YnME4hfB_1mifYJyGL0m_1ujBSStc5qlJiEFiAM_0DjqTRKCNv1aobl5BZV3lNkKk_eWKMvwRK1LJPxilZTs2lpWC8LaV2okpTiL8ohx8kcYVFRatwR-_J4K7xskXXoljENkv90NaVOPLl6m9SfVG04yYcVzslDdHllopPQ_s_2i_sML_2fZAZaCbQuwEcFAlom9-Kc7X1J-pWvUwZA1rLj4zc3vfbtg2oIm5S9ZaqwLOr2ZrnHGOpqtTv-XKt_7vpziYdqhAlqvEGBfRQ"
```

Réponse attendue :

access_token : le token qui vous permettra de faire des appels à l'API Helloasso

refresh token : le token qui vous permettra de rafraichir l'access token

token_type : le type de token, sera toujours « bearer »

expires_in : la durée de vie en seconde de l'access token

Exemple de réponse :

```
{  
  "access_token":  
    "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiJlODdhMDA0ZTYzY2M0ODNmOTk1ZmQzMjcwOTg0NzVmNSIsImNwcyI6WyJBYSZlc3NkdWJsYWNEYXRhIiwiaWF0IjE5MTYyvdDhY2tvdXQILCJHcmFudEF1dGhvcmcl6YXRpb25Db2RII0I0InBhcil6IkZhY2lsYXNzbyslIm5iZil6MTYvODA4NzMyNiwiZXhwIjoxNjI4MDg5MTI2LCApc3MiOiJodHRw
```

```

czovL2FwaS5oZWxsb2Fzc28uY29tliwiYXVkljoiZDA2M2MxOGJmMjYyNDM3ODg4OWY0ZTUyODU4NzY1NTcifQ.d7vJN-u
L8xI_YyT6RG5wtJaTscPrDHEUOJwu37KO9Ykhj5dv9lhZqjRKftlfy_NzWcGoXCg_AmXvhOEYIXDeVPX2AahUVeZNrwHxHa
LXdueexuXmO8zSo3xhQgszmMJA3fQSi6allZNvjpryXOMPNIzWJ-P_qXd0J20Ycrtve831S3AtLNUxZRUF3CJgame1mZVRY
Lg86l46CxEcprRYbztwtl5wKDqIKN27t6am3kWrKo87Um4OdVQ9p8OAsEWIV9YIJdtJbZ9NXA2bFilOsSY11_DReCAArLU-
n1h4mX2-N_8qHcTauyYWuegcEC5WV9nM7NRQEz-blG7mtj1-Qg",
  "token_type": "bearer",
  "expires_in": 1799,
  "refresh_token":
"oDeJF3wQ0gCFuZNpnYO7-G3yB521sEamT7-Ov9D6VhcUOKlq8DM0OkkcxowkotsuVGSFFjDC2R-TeBiGVek_B_eqqmtN
1vtQzJqpFD1pOCphqu1XZ3Rzav8cQKDLySg3jS_wR-W9EmoCzDGXz7C35B9l2ztZlkpVCgvZW9AidAYBBVdKSG6MCLpm
jvmVKkAganM01wyOAho9IMBQUxGdZJUuGoe-ouucdEC438ZFCC-NibrDepiYRNxKmfW302XCTsPhxkEiZbenAtPpwXWu2l
lfDpnMXhp7trE5CMzfYirQG_sChgOSB0Ql6GFAE6YmciCWpAEUUVhTMmx5W6tmfXnCVi0eYm-5ksR3bQFWVhaTn6rXpE
5VS3sjQh1mW7xhUW1HyyQdGGOQCXoRGmU6Qs2zJOnjQdJpTPZx-_QEjr88nP0W"
}

```

Vous disposez maintenant d'un nouvel `access_token` d'une validité de 1799 secondes (30minutes) et d'un nouveau `refresh_token` pour le rafraîchir.

PRIVILÈGES ET RÔLES

Pour effectuer des appels API sur <https://api.helloasso.com/v5> vous devez posséder des privilèges qui vous seront attribués au préalable.

Certains points d'API nécessitent également d'avoir un rôle. Les rôles sont obtenus via la mire d'autorisation, et donc une autorisation explicite d'une association à accéder à ses données.

Privilèges :

AccessPublicData : Vous permet d'appeler des points d'API pour récupérer la donnée publique de HelloAsso accessible sur le site.

FormOpenDirectory : Vous permet d'appeler des points d'API pour récupérer une liste précise de formulaires à l'aide de filtres.

AccessTransactions : Vous permet d'appeler des points d'API pour récupérer les détails sur les transactions.

OrganismDirectory : Vous permet d'appeler des points d'API pour récupérer une liste d'associations avec des filtres.

Checkout : Vous permet d'utiliser le type de formulaire checkout et de permettre aux contributeurs d'effectuer des paiements, plus d'information sur ce [document](#).

Rôles :

OrganizationAdmin : Vous permet d'appeler des points d'API pour récupérer des données sensibles d'une association comme les paiements ou de créer certains types de formulaires.

Pour décomposer votre access_token et regarder les détails qui le compose (privilèges, rôles, expiration ...) vous pouvez le copier-coller sur le site : <https://jwt.io/>

Récupérer de la donnée :

Voici un exemple d'appel API pour récupérer le détail d'une association.

Il s'agit d'un GET sur l'URL : <https://api.helloasso.com/v5/organizations/{organizationSlug}>

{organizationSlug} étant le nom normalisé de l'association que l'on veut récupérer

Header :

authorization: "Bearer " + votre access_token

Exemple cURL :

```
curl --location --request GET 'https://api.helloasso.com/v5/organizations/hugobordeauxha' \
```

```
--header 'Authorization: Bearer
```

eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiI0Mjg5YTFiODViY2l0NTM2OGY0YWQ1NDRMyNjlyNDRhMCIsImNwcyI6WyJBW2Nlc3NGdWxsRGF0YSIsIkFjY2Vzc1B1YmxpY0RhdGEiLCJBW2Nlc3NUcmFuc2FjdGlbnMiLCJDYXJ0TWFuYWdlbWVudCIsIkZvcmlBZG1pbilSkZvcmlEaXJlY3RvcnkiLCJHcmFudFBhc3N3b3JkQ3JIZGVudGIhbmHMILCJHcmFudFNpZ25VcCIsIk9yZ2Fu

aXNtRGlyZWNoY3J5IiwiaW50cmF0aW9uQWRtaW4iLCJPcmdhbml6YXRpb25QcmVTaWduVXAiLCJQZXJmb3JtUGF5bWVudCjdlLCJ1aWQiOiIxODQ1NjY1IiwidXJzljpbIk9yZ2FuaXphdGlvbKfkbWluliwiRm9ybUFkbWlull0slm5iZil6MTYyOTI4MDMxOSwiZXhwIjoxNjl5MTgyMTE5LCJpc3MiOiJodHRwczovL2FwaS5oZWxsY2Fzc28uY29tliwiYXVkljoizTRhNGY1MTVmYWZkNGQ4YT

hhMGUwNzZhZTNmNWU2YWYifQ.VByeQzRWuJeStE2QQv6DeLs9UU07gr_ZGRp2BDtyCpvkdYoIONVoUkvoGJmj7FNRcXCM0Y8yAjBPW0gFty12z8LNHNctC4WLdf42nOutHVRBBBoHJCiH

HkzAWTjA1sRkk-uMuK8QNKMdrt_4g4CCq1jJvpnYE64F4u7PG1Ami3xo80-EvAONiNLVxMVXWoMUuhb-TQ9xJzsKCMizwkQuKdU6e8w1o5N05doOia3bxXEthZcEQMRBbGIK9PPBJRNP81

RigsCt1p2xs9Cyn1LtNEgDpgIP9dW4ZVDhsJkijqhNHw-UY439hNhWrNZZN3PtaGXppa8G9UH

WirtStEzTKw' \

Exemple de réponse :

```
{
  "isAuthenticated": true,
  "banner":
    "https://www.helloasso.com/assets/img/photos/my-little-pony-wallpaper-80s-toybox-33629715-1191-670-d8c7f7d2d7a0438299d959defdf253bb.png",
  "contact": {
    "email": "hugo@helloasso.org",
    "phoneNumber": ""
  },
  "fiscalReceiptEligibility": true,
  "fiscalReceiptIssuanceEnabled": true,
  "type": "Association1905Rup",
  "logo": "https://www.helloasso.com/assets/img/logos/croppedimage-d86a56727f81468d877fce8bd0f0fce1.png",
  "name": "HugoBordeauxHa",
  "organizationSlug": "hugobordeauxha",
  "role": "OrganizationAdmin",
  "url": "https://www.helloasso.com/associations/hugobordeauxha",
  "city": "Bordeaux",
  "zipCode": "33100",
  "description": "Association de test"
}
```

MIRE AUTHORIZATION

La mire d'autorisation permet aux applications du partenaire d'accéder à des ressources protégées (ressources appartenant à une association).

L'association donne son autorisation de partager ses données, vous pourrez ensuite obtenir un token d'accès qui permettra de faire des appels API avec le rôle nécessaire.

Typiquement, vous devez afficher un bouton à votre utilisateur pour lui demander de se connecter sur HelloAsso, et vous autoriser à accéder à ses ressources.



Voici le code du bouton en CSS / Html

```
<button class="HaAuthorizeButton">
  
  <span class="HaAuthorizeButtonTitle">Connecter à HelloAsso</span>
</button>
```

```

<style>
.HaAuthorizeButton {
  align-items: center;
  -webkit-box-pack: center;
  -ms-flex-pack: center;
  background-color: #FFFFFF;
  border: 0.0625rem solid #49D38A;
  border-radius: 0.125rem;
  display: -webkit-box;
  display: -ms-flexbox;
  display: flex;
  padding: 0;
}
.HaAuthorizeButton:disabled {
  background-color: #E9E9F0;
  border-color: transparent;
  cursor: not-allowed;
}
.HaAuthorizeButton:not(:disabled):focus {
  box-shadow: 0 0 0 0.25rem rgba(73, 211, 138, 0.25);
  -webkit-box-shadow: 0 0 0 0.25rem rgba(73, 211, 138, 0.25);
}
.HaAuthorizeButtonLogo {
  padding: 0 0.8rem;
  width: 2.25rem;
}
.HaAuthorizeButtonTitle {
  background-color: #49D38A;
  color: #FFFFFF;
  font-size: 1rem;
  font-weight: 700;
  padding: 0.78125rem 1.5rem;
}
.HaAuthorizeButton:disabled .HaAuthorizeButtonTitle {
  background-color: #E9E9F0;
  color: #9A9DA8;
}
.HaAuthorizeButton:not(:disabled):hover .HaAuthorizeButtonTitle,
.HaAuthorizeButton:not(:disabled):focus .HaAuthorizeButtonTitle {
  background-color: #30c677;
}
</style>

```

Lorsque l'utilisateur clique sur le bouton ou le lien, vous devez ouvrir une fenêtre pop-up vers cette URL : <https://auth.helloasso.com/authorize> avec tous les paramètres de requête appropriés.

client_id : votre client id

redirect_uri : L'URL de redirection qui sera utilisée lorsque l'autorisation sera terminée (succès ou erreur). Pour des raisons de sécurité, le domaine de l'URL de redirection doit être le même que celui configuré sur votre client dans notre base de données (lorsque nous avons créé votre client). Doit utiliser le protocole sécurisé https. Veuillez nous contacter pour définir ou mettre à jour ce domaine si nécessaire.

Exemple : <https://partenaire.com/success>

code_challenge : PKCE (Proof Key for Code Exchange) est une mesure de sécurité pour l'octroi de l'autorisation.

La spécification peut être trouvée ici : <https://tools.ietf.org/html/rfc7636>

Nous vous demandons d'utiliser la méthode challenge_method S256.

Vous devez générer une chaîne de texte aléatoire de 43 à 128 caractères, parmi les caractères suivants :

ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789-._~

Gardez la chaîne générée, elle sera utile pour obtenir le nouveau token d'accès possédant les nouveaux droits.

Vous créez ensuite le Code Challenge : vous hachez la chaîne de texte avec la fonction de hachage SHA-256 puis encodez-le en base64 et enfin l'encoder par URL.

Si vous souhaitez tester votre générateur de challenge de code, vous pouvez le faire ici avec cet outil en ligne : <https://tonyxu-io.github.io/pkce-generator/>

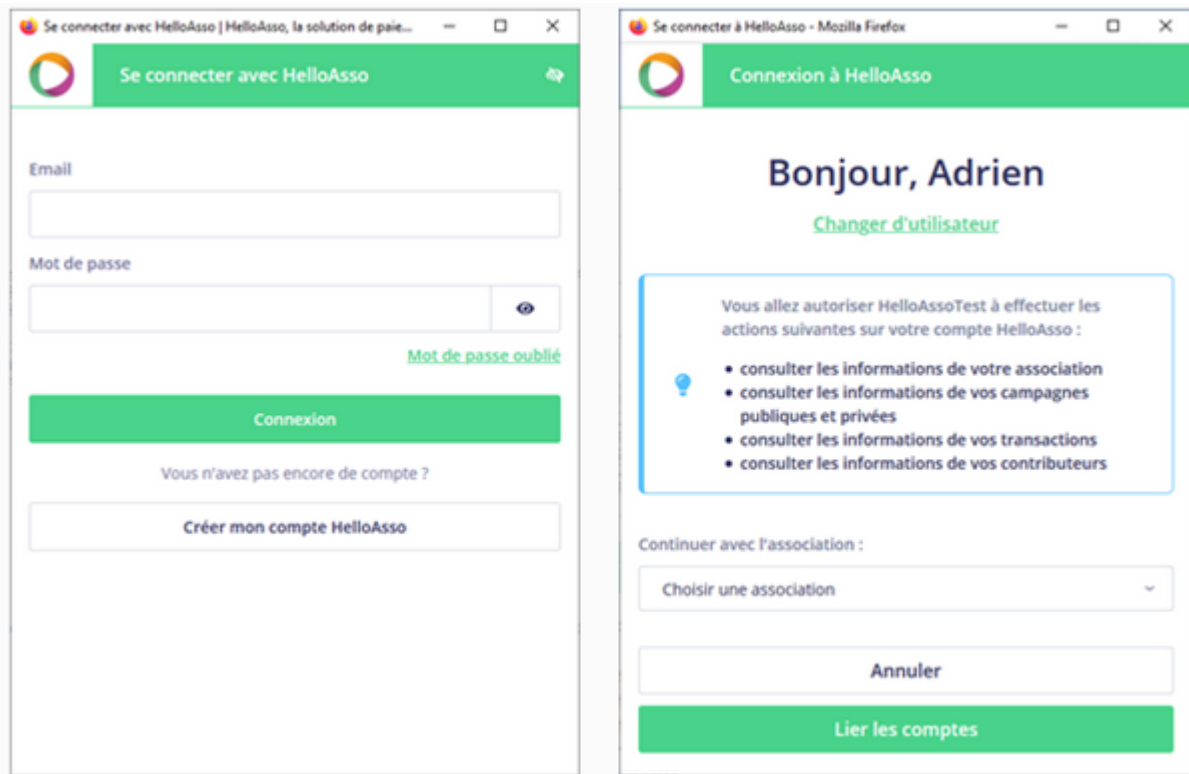
code_challenge_method : "S256"

state : Une valeur qui vous sera renvoyée lors du callback, il peut être utilisé pour éviter des appels frauduleux. Le paramètre state du callback devra correspondre à celui que vous avez envoyé. Doit être une chaîne de texte (de moins de 500 caractères). Vous pouvez également l'utiliser pour encoder toutes les données que vous souhaitez.

Exemple de requête : <https://auth.helloasso.com/authorize>

```
?client_id=9fdc22226bf24ff99b875f4a7c503715
&redirect_uri=YOUR_REDIRECT_URI
&code_challenge=YOUR_CODE_CHALLENGE
&code_challenge_method=S256
&state=abc
```

Cela affichera la fenêtre de connexion, puis la fenêtre d'autorisation pour l'utilisateur :



L'utilisateur a la possibilité de s'inscrire et d'enregistrer son organisation, s'il n'en a pas déjà une.

Lorsque l'utilisateur termine le processus, la fenêtre redirige vers la `redirect_uri` donnée, avec l'`authorization_code` en paramètre (ou avec un code d'erreur si une erreur s'est produite).

Paramètres de la réponse :

code : Il s'agit de l'`authorization_code` généré, il est valable 5 minutes.
Vous pouvez échanger ce code contre un `access_token` et un `refresh_token`

state : Si vous avez envoyé un `state` dans le premier appel il sera renvoyé ici

Exemple d'URL de retour :

<https://www.partenaire.com/success?code=xipPrCkrZ7DkAx9pzGYafMn39Zbfgc978j2cvBxBHfmDaRGutn6zQKH4fE6ZXqVwVqHW35RYAbtSWGx95i3AQci52hQZ4pzF7ARC&state=abc>

Echanger un authorization_code contre un access et refresh token

Vous pouvez échanger un authorization_code contre un token d'accès en procédant à un appel POST sur l'URL <https://api.helloasso.com/oauth2/token>

Header :

content-type: application/x-www-form-urlencoded

Body :

client_id : votre client id

client_secret : votre client secret

grant_type : "authorization_code"

code : le code que vous avez reçu en retour du premier appel.

redirect_uri : la même URL de redirection utilisé dans le premier appel

code_verifier : la chaîne de caractère que vous avez généré aléatoirement avant que vous ne l'encoder

Réponse attendue :

access_token : le token qui vous permettra de faire des appels à l'API Helloasso

refresh_token : le token qui vous permettra de rafraîchir l'access_token, il est **valable 1 mois**

token_type : le type de token, sera toujours "bearer"

expires_in : la durée de vie en seconde de l'access_token

organization_slug : le nom normalisé de l'association concernée

Conseils d'utilisation :

Ces nouveaux access_token / refresh_token ne remplacent pas ceux fournis par HelloAsso. Ils doivent uniquement servir à récupérer des informations concernant l'association qui vient de se lier.

Le fait d'enregistrer ces tokens et la date d'obtention dans votre base de données vous permettra de connaître leur date d'expiration, 30 minutes pour l'access_token et 30 jours pour le refresh_token.

Si au bout de trente jours vous n'avez pas renouvelé ces tokens (voir plus haut « rafraîchir un token »), ils seront alors expirés et la liaison avec l'association sera brisée. Il faut donc prévoir un mécanisme automatique et régulier pour rafraîchir ces tokens.

Vous pouvez avoir autant de liaisons que vous le souhaitez (et donc de tokens).