

API Isogeo

Documentation pour les développeurs tiers

Rappels	3
L'API en action	3
S'abonner à la lettre d'informations	3
Principes généraux	3
Authentification	4
Utilisation d'un proxy	4
Routes	4
Généralités	4
Sous-entités	5
Ressource	5
Langue	6
Ségrégation par partage	7
Visualiser les requêtes à l'API	7
La recherche	8
Les paramètres	8
q - recherche	8
Description	8
Exemples	8
box - emprise spatiale (bounding box)	8
Description	8
Exemples	9
geo - enveloppe de recherche	9
Description	9
Exemples	9
rel - relation géométrique	9
Description	9
Exemples	9
ob - tri (order by)	10
Description	10
Exemples	10
od - sens du tri (order direction)	10
Description	10
Exemples	10
_limit - longueur de la liste de résultats	10
Description	10
Exemples	10
_offset - index dans la liste de résultats	10
Description	10
Exemples	10
_include - sous-ressources à récupérer	11

Description	11
_id - liste d'identifiants	11
Description	11
Exemples	11
Résultat d'une recherche	11
Les tags	11
Utilisation	11
Les types de tag	11
Entités	12
Ressource	13
Liens	13
Description	13
Matrice des actions possibles par type de lien	14
Exemples	14
Les liens de type link	15
Les liens téléversés	16
Contacts	16
Description	16
Exemples	17
Couches	17
Service	18
Donnée	18
Opérations	18
Limitations	18
Description	18
Les directives	19
Exemples	19
Conditions (CGUs)	20
Description	20
Exemples	20
Authentification	21
Concepts	21
Généralités	21
Applications utilisateur	22
Caractéristiques	22
URIs de redirection	22
Type	22
Authorization Code Grant	23
Requête d'autorisation	23
Demande du jeton	24
Implicit Grant	24
Applications de groupe	25
Caractéristiques	25
Client Credentials Grant	25
Refresh Token	25

Proxy	25
API Isogeo	26
Fichiers	26
Développement	27
Annexes	27
Webinaire de présentation et d'introduction à l'API (19 janvier 2016)	28
Authentification via OpenCatalog [dépréciée]	28

Rappels

- l'API n'est pas publique et est susceptible d'être modifiée ;
- la plupart des actions disponibles sur <https://app.isogeo.com> sont disponibles dans l'API (sauf le SCAN et fonctionnalités particulières) ;
- l'API ne concerne pour l'instant que l'accès en lecture des informations d'Isogeo. La possibilité d'écriture sera ouverte par la suite.
- la documentation s'appuie sur les éléments de vocabulaire liés au secteur de l'information géographique généralement et plus spécifiquement à la plateforme Isogeo. Un détour par [le glossaire de l'aide en ligne](#) peut aider à mieux comprendre les termes métier employés.

L'API en action

Applicatifs permettant de voir l'API en action :

- l'OpenCatalog, module applicatif d'Isogeo, voir un exemple [ici](#) ou [ici intégré à un portail](#).
- un catalogue de données : <https://www.geomayenne.fr/portail/acces-aux-donnees/donnees/#catalogue>.
- un portail de données ouvertes : <https://demo.isogeo.net>.

Un webinaire de présentation et d'introduction a été réalisé le 19 janvier 2016 dont la vidéo est accessible en ligne : <https://www.youtube.com/watch?v=sOv9LeMycbo>.

S'abonner à la lettre d'informations

Une lettre d'information (sans spam ni marketing ajoutés) est mise à disposition pour être averti des évolutions, des nouvelles ressources ou des problèmes de l'API : [s'inscrire](#).

Principes généraux

L'API Isogeo est disponible à l'adresse de base suivante : <https://v1.api.isogeo.com>. En cas de problème de vérification du certificat SSL :

- vérifiez que [les certificats racine de GoDaddy](#) sont bien installés sur votre plateforme (c'est en général le cas).
- l'API Isogeo est également disponible en HTTP (sauf pour l'authentification).

L'API Isogeo est [une API REST](#) qui communique en [JSON](#) ([type MIME](#) `application/json`). À ce titre elle donne accès à des entités via [des routes](#).

Une description de l'API [au format Swagger](#) est disponible à l'adresse : <http://help.isogeo.com/api/swaggerui/>.

Authentification

L'authentification à l'API Isogeo se fait via [le protocole OAuth 2.0](#) :

- vous devez être en possession de vos identifiants d'application (*client_id* et *client_secret*) qui vous ont été fournis directement par l'équipe Isogeo.
- votre application peut récupérer un *access token* via les routes d'authentification disponibles :
 - autorisation : <https://id.api.isogeo.com/oauth/authorize>
 - token : <https://id.api.isogeo.com/oauth/token>
- les flots d'autorisation dépendent de la configuration de votre application (à voir avec l'équipe Isogeo) :
 - application utilisateur confidentielle : Authorization Code Grant.
 - application utilisateur publique : Implicit Grant.
 - application de groupe : Client Credentials Grant.
- un *access token* valide doit être envoyé à la plateforme Isogeo lors de chaque appel à l'API via [un en-tête d'authentification de type Bearer](#).
- un *access token* a une date d'expiration et doit être renouvelé régulièrement (éventuellement avec un *refresh token*).

L'implémentation du protocole OAuth 2.0 par Isogeo est conforme au standard :

- la documentation officielle de OAuth 2.0 fait donc référence pour l'authentification Isogeo ([RFC 6749](#)).
- il est également fortement conseillé d'utiliser [des bibliothèques standard d'authentification OAuth 2.0](#). Par exemple :
 - Javascript : [oauth](#).
 - .NET : [Microsoft.Owin.Security.OAuth](#).

Une description technique plus complète de l'authentification Isogeo est disponible [plus bas dans ce document](#).

Utilisation d'un proxy

Dans le cas d'une application web il est en général nécessaire de faire transiter les appels vers l'API Isogeo via un proxy, entre autres pour des raisons de sécurité (confidentialité des identifiants d'application ou des tokens d'authentification). C'est la raison pour laquelle l'API Isogeo ne supporte pas les appels [Cross Origin](#).

Le développeur de l'application est responsable de la sécurisation de son proxy :

- contre les attaques ([DoS](#) par exemple).
- contre la fuite de données confidentielles dans le cas d'une application à usage restreint (contre les moteurs de recherche par exemple).

Une description plus complète des proxies est disponible [plus bas dans ce document](#).

Routes

Généralités

L'API Isogeo est une API REST qui donne accès à des entités via des routes. Lorsqu'on souhaite accéder à une entité en particulier il faut appeler la route correspondante, qui contient l'identifiant de l'entité en question.

Les principales routes de l'API sont les suivantes :

Verbe HTTP	URI	Description
GET	/resources/search	Moteur de recherche des fiches de métadonnées
GET	/resources/{:rid}	Retourne la fiche de métadonnées
GET	/resources/{:rid}.xml	Télécharge la fiche de métadonnées en ISO 19139. Nécessite soit : • d'initialiser l'en-tête Accept avec le type MIME <code>datasetapplication/xml</code>. • de rajouter le paramètre <code>_format=xml</code>. Si un proxy vers les fichiers est présent, il est possible de l'indiquer via le paramètre <code>proxyUrl</code> .
GET	/resources/{:rid}/links	Retourne les ressources associées de la fiche
GET	/shares	Retourne les caractéristiques des partages accessibles à l'application de groupe (nom, groupe de travail...)

Sous-entités

Certaines entités possèdent beaucoup d'informations, regroupées dans des sous-entités. Afin d'optimiser l'usage de notre plateforme, l'API ne renvoie par défaut qu'une partie de ces sous-entités. Afin d'accéder à plus d'informations :

- il est possible de les demander en ajoutant le paramètre `_include` sur les routes de l'API qui concernent l'entité principale (conseillé).
- il est possible de les demander via un appel spécifique lorsqu'une route dédiée existe.

Par exemple pour récupérer une ressource ainsi que ses liens il est possible d'appeler :

- la route principale avec le paramètre `_include:/resources/{:rid}?_include=links`
- la route principale `/resources/{:rid}` puis la route dédiée `/resources/{:rid}/links`.

Ce paramètre est également valable pour la recherche.

Les valeurs possibles pour ce paramètre dépendent de l'entité en question :

- les valeurs invalides sont ignorées par l'API (`_include=toto`).
- il est donc également possible de demander une sous-entité qui n'existe pas sur l'entité en question (`_include=layers` sur une donnée, par exemple).

Ressource

Les sous-entités renvoyées par défaut sont :

<code>_creator</code>	Le propriétaire de la ressource.
<code>tags</code>	Les étiquettes associées à la ressource.
<code>links</code>	Les liens attachés à la ressource. (renvoyés par défaut uniquement pour la route <code>GET /resources/{:id}</code>)

Les sous-entités disponibles sont :

<code>conditions</code>	Les conditions qui s'appliquent sur la ressource.
<code>contacts</code>	Les contacts de la ressource.
<code>coordinate-system</code>	Le système de coordonnées de la donnée géographique.
<code>events</code>	Les événements de la ressource (inutile en général : les événements importants font partie de la ressource elle-même).
<code>feature-attributes</code>	Les attributs de la donnée géographique vecteur.
<code>keywords</code>	Les mots-clés associés à la ressource (inutile en général : les mots-clés font également partie des <code>tags</code>).
<code>layers</code>	Les couches diffusées par un service.
<code>limitations</code>	Les limitations qui s'appliquent sur la ressource.
<code>links</code>	Les liens attachés à la ressource.
<code>operations</code>	Les opérations disponibles sur un service.
<code>serviceLayers</code>	Les couches (de services) associées à la donnée géographique.
<code>specifications</code>	Les spécifications associées à la ressource.

Exemple d'appel : `GET /resources/search?_include=contacts,links`

Langue

Certaines informations sont disponibles en plusieurs langues : les thèmes INSPIRE par exemple. Il est possible de spécifier la langue souhaitée en ajoutant le paramètre `_lang` sur toutes les routes de l'API :

- `fr`: français (défaut)
- `en`: anglais

Par exemple la recherche `/resources/search` remonte dans la liste des tags les thèmes INSPIRE suivants :

```
{
  "keyword:inspire-theme:addresses": "Adresses",
  "keyword:inspire-theme:administrativeunits": "Unités administratives",
  "keyword:inspire-theme:agriculturalandaquaculturefacilities": "Installations agricoles et aquacoles",
  "keyword:inspire-theme:areamanagementrestrictionregulationzonesandreportingunits": "Zones de gestion, de restriction ou de réglementation et unités de déclaration",
  "keyword:inspire-theme:atmosphericconditions": "Conditions atmosphériques",
```

```

"keyword:inspire-theme:biogeographicalregions": "Régions biogéographiques",
"keyword:inspire-theme:buildings": "Bâtiments",
"keyword:inspire-theme:cadastralparcels": "Parcelles cadastrales"
}

```

Avec la recherche /resources/search?_lang=en, on obtient :

```

{
  "keyword:inspire-theme:addresses": "Addresses",
  "keyword:inspire-theme:administrativeunits": "Administrative units",
  "keyword:inspire-theme:agriculturalandaquaculturefacilities": "Agricultural and aquaculture facilities",
  "keyword:inspire-theme:areamanagementrestrictionregulationzonesandreportingunits": "Area management/restriction/regulation zones and reporting units",
  "keyword:inspire-theme:atmosphericconditions": "Atmospheric conditions",
  "keyword:inspire-theme:biogeographicalregions": "Bio-geographical regions",
  "keyword:inspire-theme:buildings": "Buildings",
  "keyword:inspire-theme:cadastralparcels": "Cadastral parcels"
}

```

Ségrégation par partage

Par défaut les routes de l'API concernent toutes les entités qui sont accessibles aux applications :

- les entités accessibles à l'utilisateur dans le cas d'une [application utilisateur](#).
- les entités partagées à l'application dans le cas d'une [application de groupe](#). Par exemple si un administrateur crée 3 partages différents à la même application, la recherche s'effectuera sur l'ensemble des fiches contenues dans les 3 partages.

Dans le cas d'une application de groupe on peut donner accès au contenu partage par partage en ajoutant sur toutes les routes de l'API le paramètre `s={:id}`. Par exemple la requête <https://v1.api.isogeo.com/resources/search?s=c502e8f7c9da4c3aacdf3d905672d54c> est une recherche sur le partage spécifié, plutôt que sur l'ensemble des partages faits à l'application. L'application *OpenCatalog* fonctionne ainsi.

Visualiser les requêtes à l'API

Toutes les requêtes et réponses à l'API sont aisément consultables depuis [APP](#) (compte gratuit disponible sur demande par mail) via le mode développeur des navigateurs (touche F12 généralement). Notez que ces appels passent [par un proxy](#).

Name	Path	Method	Status	Domain	Type	Size	Time	Server	Timeline
memberships?	app.isogeo.com/api/v1/account	GET	200 OK	app.isogeo.com	xhr	43.6 KB	481 ms	Microsoft...	
limits?	app.isogeo.com/api/v1/groups/c18ddc43d1c045fc85c2ba...	GET	200 OK	app.isogeo.com	xhr	811 B	652 ms	Microsoft...	
c18ddc43d1c045fc85c2ba1b4233eef1?	app.isogeo.com/api/v1/groups	GET	200 OK	app.isogeo.com	xhr	1.2 KB	247 ms	Microsoft...	
statistics?	app.isogeo.com/api/v1/groups/c18ddc43d1c045fc85c2ba...	GET	200 OK	app.isogeo.com	xhr	847 B	3.20 s	Microsoft...	
format?	app.isogeo.com/api/v1/groups/c18ddc43d1c045fc85c2ba...	GET	200 OK	app.isogeo.com	xhr	1010 B	875 ms	Microsoft...	
search?	app.isogeo.com/api/v1/resources	GET	200 OK	app.isogeo.com	xhr	4.8 KB	3.31 s	Microsoft...	

Pour rappel, la bonne adresse pour requêter l'api est <https://v1.api.isogeo.com>.

Exemple de recherche :

- **basique** : <https://v1.api.isogeo.com/resources/search>
- **la même avec la plupart des paramètres et *includes* possibles** :
https://v1.api.isogeo.com/resources/search?_limit=100&_lang=fr&_offset=0&_include=conditions,contacts,coordinate-system,events,feature-attributes,keywords,limitations,links,specifications

Exemple de fiche individuelle :

- <https://v1.api.isogeo.com/resources/5dbf638e0da048fc82ddc4cd615ef168>

La recherche

Les paramètres

Voici les différents paramètres de la recherche. Aucun paramètre n'est obligatoire.

q - recherche

Description

Texte de la recherche + tags (concaténés avec des espaces)

La recherche textuelle n'est pas sensible à la casse ou aux accents :

- *EAU* et *eau* renvoient les mêmes résultats.
- *métro* et *metro* renvoient les mêmes résultats.

[Les tags](#) permettent de réaliser des filtres précis sur des entités du modèle Isogeo : mots-clés, catalogues, systèmes de coordonnées...

Le moteur de recherche n'implémente que l'opérateur ET : lorsque plusieurs mots ou tags sont spécifiés dans la recherche, seules les métadonnées qui correspondent à tous les critères sont remontées.

Notez que les résultats d'une recherche textuelle pourront être légèrement différents en fonction de [la langue spécifiée](#).

Exemples

`/resources/search?q=Bamako`

Renvoie les ressources contenant le mot Bamako

`/resources/search?q=keyword%3Aisogeo%3Aloire`

Renvoie les ressources possédant le mot-clé *loire*.

etc.

`/resources/search?q=owner%3A643f1035377b4ca59da6f31a39704c34`

Renvoie les ressources du groupe possédant l'identifiant `643f1035377b4ca59da6f31a39704c34`.

`/resources/search?q=coordinate-system%3A4326`

Renvoie les ressources ayant le système de coordonnées 4326.

`/resources/search?q=Bamako%20keyword%3Aisogeo%3Aloire`

Renvoie les ressources contenant le mot Bamako et possédant le mot-clé *loire*

box - emprise spatiale (*bounding box*)

Description

Emprise de recherche. Renvoie les ressources dont l'enveloppe possède la relation géographique spécifiée (intersection par défaut, cf. [rel](#)) avec l'emprise envoyée. Si la ressource ne possède pas d'enveloppe, elle n'est pas remontée. Les coordonnées de l'emprise sont en WGS84 ([EPSG 4326](#)) et respecte les [spécifications d'OpenSearch](#).

Exemples

/resources/search?box=-4.970, 30.69418, 8.258, 51.237

geo - enveloppe de recherche

Description

Enveloppe de recherche. Renvoie les ressources dont l'enveloppe possède la relation géographique spécifiée (intersection par défaut, cf. [rel](#)) avec l'enveloppe envoyée. Si la ressource ne possède pas d'enveloppe, elle n'est pas remontée. L'enveloppe doit être décrite au format WKT.

Exemples

/resources/search?geo=POLYGON((0.582%2040.496%2C%200.231%2040.737%2C%200.736%2042.869%2C%203.351%2042.386%2C%203.263%2041.814%2C%202.164%2041.265%2C%200.978%20%20%2040.957%2C%200.802%2040.781%2C%200.978%2040.649%2C%200.582%2040.496))

rel - relation géométrique

Description

Relation géométrique : permet de changer la requête spatiale appliquée pour filtrer les résultats dans une emprise ou un polygone.

Uniquement interprété si les paramètres [box](#) ou [geo](#) sont passés. Si les 2 paramètres sont présents, une intersection est réalisée.

Les valeurs possibles sont basées sur les relations standard définies par l'OGC dans [OGC 06-103r4](#) §6.1.15.

Valeurs :

- *contains* : les métadonnées sont remontées si elles contiennent complètement l'enveloppe spécifiée.
- *disjoint* : les métadonnées sont remontées si elles n'intersectent pas l'enveloppe spécifiée.
- *equals* : les métadonnées sont remontées si leur enveloppe est la même que l'enveloppe spécifiée.
- *intersects* : (défaut) les métadonnées sont remontées si elles intersectent l'enveloppe spécifiée. Il s'agit de l'option par défaut.
- *overlaps* : les métadonnées sont remontées si elles chevauchent l'enveloppe spécifiée.
- *within* : les métadonnées sont remontées si elles sont entièrement contenues dans l'enveloppe spécifiée.

En pratique, les valeurs les plus couramment utilisées sont :

- *intersects*

- *within*
- *contains*

Exemples

/resources/search?geo=POLYGON((0.582%2040.496%2C%200.231%2040.737%2C%200.736%2042.869%2C%203.351%2042.386%2C%203.263%2041.814%2C%202.164%2041.265%2C%200.978%20%2040.957%2C%200.802%2040.781%2C%200.978%2040.649%2C%200.582%2040.496))&rel=within

ob - tri (order by)

Description

Signifie *OrderBy*. Permet de trier les résultats de la recherche.

Valeurs:

- *_created* : (défaut) date de création de la fiche de métadonnées.
- *_modified* : date de la dernière modification de la fiche de métadonnées.
- *title*: titre de la ressource.
- *created* : date de création de la ressource (pas forcément renseignée).
- *modified* : date de dernière modification de la ressource (pas forcément renseignée).
- *relevance*: score de pertinence par rapport aux termes de recherche.

Exemples

/resources/search?ob=title

Renvoie les ressources triées par titre.

od - sens du tri (order direction)

Description

Signifie *OrderDirection*. Permet de choisir le sens du tri.

Valeurs:

- *desc*: (défaut) descendant
- *asc*: ascendant

Exemples

/resources/search?ob=title&od=asc

Renvoie les ressources triées par ordre alphabétique.

_limit - longueur de la liste de résultats

Description

Nombre de résultats maximum retourné. Utile pour effectuer une pagination

Défaut 20. Maximum 100.

Exemples

/resources/search?ob=_modified&od=desc&_limit=100

Renvoie les 100 dernières fiches modifiées.

/resources/search?ob=_created&od=desc&_limit=1

Renvoie la dernière fiche créée.

_offset - index dans la liste de résultats

Description

Placement du curseur dans les résultats trouvés. Utile pour effectuer une pagination

Défaut 0.

Exemples

/resources/search?ob=_created&od=desc&_offset=20

Renvoie les dernières fiches créées sauf les 20 premières.

_include - sous-ressources à récupérer

Description

Sous-entités à inclure avec la ressource. Permet de tout récupérer en une fois au lieu d'utiliser les routes spécifiques de chaque sous-ressources.

Voir le [détail de ce paramètre](#).

_id - liste d'identifiants

Description

Permet de limiter la recherche à l'ensemble des ressources correspondant aux identifiants fournis. Associé à une recherche sans paramètres c'est un moyen de pouvoir récupérer plusieurs ressources en 1 seule requête.

Exemples

/resources/search?_id=b46f086ec6b24f25ae2aca61c1fc80a2,14bc943cddd749e38047381dcc9e4bcd

Résultat d'une recherche

Le résultat d'une recherche possède les attributs suivants:

- **results**: tableau des ressources trouvées
- **envelope**: l'enveloppe convexe de l'ensemble des résultats, au format [GeoJSON](#).
 - ce champ est optionnel et peut apparaître explicitement avec la valeur `null` dans le cas où une erreur est survenue pendant son calcul.
- **tags**: ensemble des tags contenus dans les ressources trouvées (et pas forcément remontées)
- **query**: la recherche effectuée
- **offset**: le placement du curseur dans les résultats (commence à 0)
- **limit**: le nombre maximum de résultats remontés (20 par défaut)
- **total**: le nombre total de résultats trouvés par la recherche

Les tags

Les tags permettent à l'utilisateur de classer les fiches de métadonnées. Certains tags sont automatiquement attachés à la ressource.

Utilisation

Ces tags sont ré-utilisables via le moteur de recherche via le paramètre *q*.

Codes des tags mis à la suite séparés par des espaces.

Lorsque plusieurs tags sont présents dans la recherche, l'opérateur **ET** est utilisé.

`q=tag1 tag2 tag3`

`q=tag1%20tag2%20tag3` avec l'encodage de l'URL.

Les types de tag

action:<type>	Permet de chercher les ressources possédant au moins un lien avec l'action <type>. Toutes les données associées à au moins un service sont considérées comme visualisables. Toutes les données associées à un service WFS ou ESRI Feature Map sont considérées comme téléchargeables.	type: • download • view • other	action:download
coordinate-system:<code>	Permet de rechercher les fiches possédant le système de coordonnées dont le code est <code>	code: code EPSG du système de coordonnées	coordinate-system:2154
format:<code>	Permet de rechercher les fiches dont le code du format est <code>	code: code du format	format:shp
keyword:inspire-theme:<code>	Permet de rechercher les fiches attachées au thème Inspire dont le code est <code>	code: code du thème Inspire	keyword:inspire-theme:administrativeunits
keyword:isogeo:<code>	Permet de rechercher les fiches possédant le mot-clé dont le code est <code>	code: code du mot-clé	keyword:isogeo:administratif
owner:<id>	Permet de rechercher les fiches dont le propriétaire possède l'identifiant <id>	id: identifiant du groupe propriétaire	owner:643f1035377b4ca59da6f31a39704c34
type:<type>	Permet de rechercher les fiches de type <type>	type: • vector-data set • raster-data set • dataset • resource • service	type:vector-data set
contact:<id>	Permet de rechercher les fiches par contact affecté, qu'importe le rôle associé.	id: identifiant unique du contact	contact:4c5249c50dc445d484cb3419a0a3fbc5:8e3b0a499e2841b1863c12ab781a4137

license:<id>	Permet de rechercher les fiches par licence affectée.	id: identifiant de la licence	license:isogeo:63f121e14eda4f47b748595e0bcc31
--------------	---	-------------------------------	---

Entités

Ressource

Une ressource représente une fiche de métadonnées. Elle peut être de type :

- `vectorDataset` : donnée de type vecteur.
- `rasterDataset` : donnée de type raster.
- `service` : service web (WFS par exemple).
- `resource` : tout autre type (un site web par exemple).

Attributs d'une ressource :

- **_created** : date de création de la fiche de métadonnées.
- **_modified** : date de la dernière modification de la fiche de métadonnées.
- **title** : titre de la ressource.
- **name** : nom de la ressource.
- **created** : date de création de la ressource (calculée à partir des événements de la ressource).
- **modified** : date de dernière modification de la ressource (calculée à partir des événements de la ressource).
- **published** : date de dernière publication de la ressource (calculée à partir des événements de la ressource).

L'emplacement des ressources (`path`) a un comportement un peu particulier : une chaîne de caractères non normalisée pour les bases de données et [un chemin de type UNC](#) pour les fichiers. ~~Ce formalisme est amené à changer dans un futur proche, par souci de cohérence, et il faut prévoir que les chemins renvoyés deviendront [des URIs de type file](#).~~

Liens

Description

Les liens permettent de lier des ressources externes à une fiche de métadonnée.

Attributs d'une ressource associée :

- **type** : type de la ressource associée. Peut prendre les valeurs :
 - `url` : une simple URL.
 - `hosted` : une URL (relative) vers un fichier hébergé sur la plateforme Isogeo.
 - une application tierce devra en général rendre cette URL accessible via un proxy.
 - `link` : ressource associée d'une autre ressource (voir plus bas pour plus d'explications).
- **url** : l'URL de la ressource associée
- **title** : le titre de la ressource associée
- **isHeader** : (*déprécié*) indique si la ressource associée doit être affichée dans l'en-tête de la donnée dans la plateforme Isogeo
- **kind** : type de ressource vers lequel pointe la ressource associée. Peut prendre les valeurs :
 - `url` : pas d'information précise (site web, pdf, ...)
 - `data` : donnée (zip, rar, png, ...)
 - `wfs` : [Web Feature Service](#)
 - `wms` : [Web Map Service](#)
 - `wmts` : [Web Map Tile Service](#)
 - `esriFeatureService` : [service d'entité Esri](#)
 - `esriMapService` : [service de carte Esri](#)
 - `esriTileService` : [service de tuiles Esri](#)
- **actions** : tableau d'actions possibles sur cette ressource associée. Peut prendre les valeurs :

- *view* : la ressource associée est destinée à être visualisée
- *download* : la ressource associée est destinée à être téléchargée
- *other* : des actions autres que la visualisation ou le téléchargement sont possibles sur la ressource associée
- **size**: (uniquement disponible pour les ressources associées de type *hosted*) taille du fichier hébergé en octets
- **link**: (uniquement pour les ressources associées de type *link*): ressource associée pointée

Matrice des actions possibles par type de lien

Kind	Téléchargement	Visualisation	Autre
Lien	X	X	(X)
WFS	(X)	(X)	X
WMS		(X)	X
WMTS		(X)	X
Esri Feature	(X)	(X)	X
Esri Map		(X)	X
Esri Tiled Map		(X)	X
Donnée brute	(X)		X

Exemples

```
{
  _id: "14bc943cddd749e38047381dcc9e4bcd",
  type: "url",
  url: "http://monsiteweb.fr/",
  title: "autre",
  isHeader: false,
  kind: "url",
  actions:
  [
    "other"
  ]
}
```

Cette ressource associée est un simple lien vers un site web.

```
{
  _id: "19532aeca11a4efdaaf97760982364f2",
  type: "hosted",
  url: "/resources/374e6188c5a3e311bf6400155d000b06/links/19532aeca11a4efdaaf97760982364f2.bin",
  size: 806797,
  title: "Format_ASC.pdf",
  isHeader: false,
  kind: "data",
  actions:
  [
    "download"
  ]
}
```

```
]
}
```

Cette ressource associée est un fichier PDF nommé *Format_ASC* de 806797 octets hébergé sur la plateforme Isogeo. Il a pour but d'être téléchargé. L'URL fournie est relative, utilisable pour requêter l'API

Isogeo
(<https://v1.api.isogeo.com/resources/374e6188c5a3e311bf6400155d000b06/links/19532aeca11a4efd aaf97760982364f2.bin> dans ce cas) et/ou [le proxy fichiers de l'application](#).

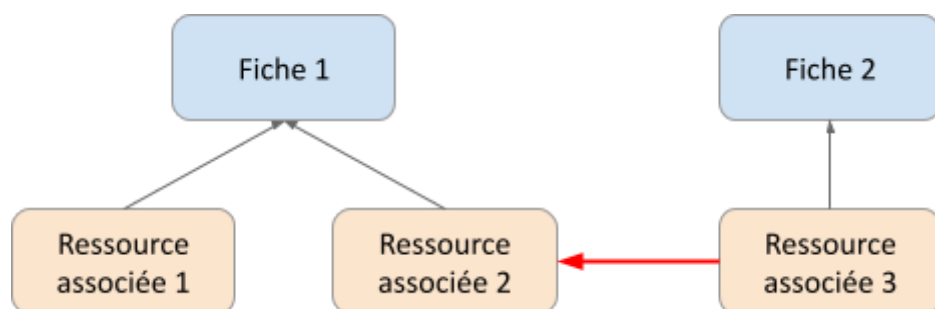
```
{
  _id: "b46f086ec6b24f25ae2aca61c1fc80a2",
  type: "url",
  url:
    "http://tiles.arcgis.com/tiles/ApmCsFVnpW3bVGyW/arcgis/rest/services/Aquifers_of_Alluvial_and_Glacia
    l_Origin/MapServer/",
  title: "Tiled Service",
  isHeader: false,
  kind: "esriTileService",
  actions:
    [
      "view"
    ]
}
```

Cette ressource associée est un service web de visualisation de donnée de type Esri Tile Service (service de tuiles Esri)

Les liens de type *link*

Les ressources associées de type *link* sont un peu particulières. Ce concept a été introduit dans le but de gérer la diffusion des ressources associées lors de la publication d'une ressource vers l'extérieur de la plate-forme Isogeo (voir [la documentation côté utilisateur](#)).

Ces ressources associées sont simplement un lien vers une ressource associée d'une autre fiche de métadonnée.



Voici une représentation de ce concept. La ressource associée 3 appartient à la Fiche 2. Elle est de type *link* et pointe vers la ressource associée 2 qui appartient à la Fiche 1.

```
{
  _id: "258c6205cbc84bbc85884ea38ca637ff",
  link:
    {
      _id: "6f95e7d75c31425692b0195a41f1c8b0",
      type: "url",
      url: "http://monwms.fr/",
    }
}
```



```

    title: "Service de visualisation",
    kind: "wms",
    actions:
    [
        "view"
    ]
},
type: "link",
url: "http://monwms.fr/",
title: "Lien vers le service de visualisation",
kind: "url",
actions:
[
    "view"
]
}

```

La ressource associée est remontée dans l'objet sous l'attribut *link*. Entre les deux ressources, le titre et les actions peuvent différer. La ressource associée de type *link* prend comme valeur de l'attribut *url* la valeur de la ressource associée vers laquelle elle pointe.

Les liens téléversés

Les ressources associées de type téléversées (`type: "hosted"`) sont remontées comme des URLs relatives. La raison est que ces adresses sont soumises à authentification et ne peuvent pas être fournies telles quelles aux utilisateurs dans une interface web par exemple.

Si vous souhaitez donner accès à vos utilisateurs au téléchargement de ces ressources associées, vous devez mettre en place un proxy (cf. [authentification](#)).

Si vous souhaitez également donner accès au téléchargement de ces ressources via la fiche ISO 19139 (cf. plus bas), votre proxy doit :

- posséder une URL de base fixe (exemple <http://open.isogeo.com/api/>).
- donner accès aux ressources téléversées à une URL qui correspond à la concaténation de l'URL de base ci-dessus et de l'URL relative renvoyée par l'API Isogeo.

Contacts

Description

L'utilisateur a la possibilité d'associer des contacts aux fiches de métadonnées. Les contacts d'une fiche sont accessibles via la route suivante :

GET `/resources/:id/contacts`

La route renvoie un tableau de contacts.

Attributs d'un contact :

- **contact**: contient l'ensemble des informations sur le contact (nom, adresse mail, adresse postale...).
- **role**: rôle du contact par rapport à la données. Peut prendre les valeurs :
 - *author*: Organisme ou personne qui a complété la donnée. Il en détient les droits moraux.
 - *pointOfContact*: Organisme ou personne que l'on peut contacter pour avoir des renseignements détaillés sur la donnée.
 - *custodian*: Organisme ou personne responsable de la gestion et de la mise à jour de la donnée.
 - *distributor*: Organisme ou personne qui délivre physiquement la donnée, soit de manière directe au destinataire, soit par l'intermédiaire d'un diffuseur

- *originator*
- *owner*
- *principalInvestigator*
- *processor*
- *publisher*
- *resourceProvider*
- *user*

Exemples

```
[
  {
    contact: {
      _id: "647ce48061a64faf9f75577a41a63229",
      _deleted: false,
      available: false,
      hash: "23bc09e00e945c3c2cc0d13f7448c261",
      type: "user",
      name: "Julien MOURA",
      email: "julien.moura@isogeo.com",
      phone: "+33 6 58 00 30 64",
      fax: "+33 9 81 40 00 66",
      addressLine1: "26 rue du faubourg Saint-Antoine",
      city: "Paris",
      zipCode: "75012",
      countryCode: "FR"
    },
    role: "custodian"
  },
  {
    contact: {
      _id: "fe3e8ef97b8446be92d3c315ccbc70f9",
      _deleted: false,
      organization: "Institut national de l'information géographique et forestière (IGN-F)",
      type: "custom",
      name: "IGN - BD",
      email: "sav.bd@ign.fr",
      countryCode: "FR"
    },
    role: "pointOfContact"
  }
]
```

Couches

Une couche géographique est définie par :

- `_id` : l'identifiant Isogeo.
- `id` : l'identifiant de la couche dans le service.
 - WFS : champ *Name* en forme étendue (cf. <https://www.w3.org/TR/xml-names/#concepts>) avec la syntaxe `{namespace}localname`.
 - le namespace pourra être utilisé dans le paramètre `namespaces` des différentes opérations WFS (cf. [OGC 09-025r2](https://www.w3.org/TR/xml-names/#ns-qualnames) §7.6.6).
 - sur GeoServer (toutes versions) : en raison d'un bug sur la gestion du paramètre `namespaces`, il est nécessaire d'utiliser le même préfixe (cf. <https://www.w3.org/TR/xml-names/#ns-qualnames>) que celui utilisé par le

serveur. S'il n'est pas connu il faudra le récupérer avec l'opération `GetCapabilities`.

- sur la version 1.0.0 du WFS, le paramètre `namespaces` n'est pas spécifié. Il est donc nécessaire d'utiliser le même préfixe (cf. <https://www.w3.org/TR/xml-names/#ns-qualnames>) que celui utilisé par le serveur. S'il n'est pas connu il faudra le récupérer avec l'opération `GetCapabilities`.
 - WMS : champ *Name* (les éventuelles couches où ce champ est absent ne sont pas renvoyées).
 - WMTS : champ *Identifier*.
- `mimeTypes` : les formats supportés par la couche
 - WMTS : c'est ici que sont renseignés les formats supportés par l'opération `GetTile` pour chaque couche.
- `titles` : les titres de la couche
 - un titre est défini par un couple langue (`lang`) et valeur (`value`).
 - chaque langue n'apparaît qu'une fois dans la liste.
 - il peut exister un titre par défaut, sur lequel la langue n'est pas définie. Ce titre est assuré d'apparaître en premier dans la liste.

Service

Pour une ressource de type service, il est possible de récupérer les couches géographiques proposées par ce service : `_include=layers`. Dans le cas où une couche est associée à une donnée, l'entité principale de la donnée est également renvoyé (`dataset`).

Donnée

Pour une ressource de type donnée, il est possible de récupérer les couches des services auxquelles elle est associée : `_include=serviceLayers`. Pour chacune de ces couches, l'entité principale du service est également renvoyé (`service`).

Opérations

Pour une ressource de type service il est possible de récupérer les opérations associées à ce service : `_include=operations`. Une opération est définie par :

- `_id` : l'identifiant Isogeo.
- `name` : le nom de l'opération.
- `mimeTypesIn` : les formats d'entrée supportés par l'opération.
- `mimeTypesOut` : les formats de sortie supportés par l'opération.
- `url` : l'URL de l'opération, dans le cas où elle serait différente de l'URL du service.
- `verb` : le verbe HTTP (*GET* ou *POST*).
 - seules les opérations de type HTTP sont supportées (par opposition à SOAP, par exemple).
 - une opération disponible sous plusieurs verbes apparaît plusieurs fois dans la liste.

Limitations

Description

L'utilisateur a la possibilité d'associer des limitations aux fiches de métadonnées. Les limitations d'une fiche sont accessibles via la route suivante :

`GET /resources/:id/limitations`

La route renvoie un tableau de limitations.

Attributs d'une limitation :

- **_id** : identifiant de la limitation
- **type** : type de la limitation. Peut prendre les valeurs :
 - *legal*
 - *security*
- **description** : description renseignée par l'utilisateur
- **restriction** : (uniquement si le type est *legal*) type de restriction. Peut prendre les valeurs :
 - copyright
 - patent
 - patentPending
 - trademark
 - license
 - intellectualPropertyRights
 - restricted
 - other
- **directive** : (uniquement si le type est *legal*) directive associée à la limitation. Voir chapitre suivant

Les directives

Les directives sont accessibles à l'aide de la route suivante :

GET /directives

Les directives sont gérées par Isogeo.

Exemples

Exemple 1 :

```
"limitations": [
    {
      "_id": "b36801353d0e493299c175d488f4c507",
      "type": "legal",
      "description": "",
      "restriction": "license",
      "directive": {
        "_id": "6756c1875d06446982ed941555102c72",
        "name": "Pas de restriction d'accès public selon INSPIRE",
        "description": "Aucun des articles de la loi ne peut être invoqué pour justifier d'une restriction d'accès public."
      }
    }
  ],
```

Exemple 2 :

```
"limitations": [
    {
      "_id": "d2391fa4c9704d7b8f67398424a40657",
      "type": "legal",
      "description": "Droit d'auteur / Droit moral (Copyright EEA / Meeddat)",
      "restriction": "copyright"
    },
    {
      "_id": "e46922a2ffdf4e98bdd6dc3d8fb67f6b",
      "type": "security"
    }
  ],
```

Conditions (CGUs)

Description

Exemples

Exemple 1 :

```
"conditions": [  
  {  
    "_id": "4c45a33adddc44638a759c3feece7633",  
    "description": "",  
    "license": {  
      "_id": "63f121e14eda4f47b748595e0bcccc31",  
      "name": "Licence ouverte ETALAB 1.0",  
      "link":  
        "http://www.etalab.gouv.fr/licence-ouverte-open-licence"  
    }  
  },  
  {  
    "_id": "8319a18d6cbb44cb8ed8e0063fbad5ae",  
    "description": "Base de données soumise aux conditions de la licence ouverte  
Etalab"  
  }  
],
```

Exemple 2 :

```
"conditions": [  
  {  
    "_id": "8324028599424acf98c8be29f06e46b7",  
    "description": "Libre de droit"  
  }  
],
```

Authentification

Concepts

La plateforme Isogeo distingue 2 types d'applications :

- les applications **utilisateur** :
 - elles accèdent aux ressources de la plateforme au nom d'un utilisateur, qui doit donc être personnellement authentifié.
 - elles accèdent uniquement aux ressources auquel l'utilisateur a droit : pas d'utilisateur, pas de ressources.
 - exemple : Isogeo App (<https://app.isogeo.com>).
- les applications de **groupe** :
 - elles accèdent aux ressources de la plateforme en leur propre nom, sans notion d'utilisateur.
 - elles accèdent uniquement aux ressources qui leur ont été partagées (menu *Administration* > *Partages* dans Isogeo App).
 - exemple : OpenCatalog (<https://open.isogeo.com>).

L'authentification à l'API Isogeo se fait via [le protocole OAuth 2.0](#). La documentation officielle de OAuth 2.0 fait donc référence pour l'authentification Isogeo ([RFC 6749](#)).

Chaque type d'application, en fonction de ses caractéristiques peut utiliser un certain nombre de flots OAuth (ou *grants*). Les applications doivent être déclarées sur la plateforme Isogeo (par l'équipe Isogeo) : vous devez être en possession de vos identifiants d'application (*client_id* et *client_secret*).

Généralités

Le but de l'authentification est de récupérer un jeton d'accès à l'API (*access token*).

La récupération de ce jeton dépend du type d'application et de ses caractéristiques, mais est conforme au standard OAuth 2.0. Pour simplifier, fiabiliser et pérenniser le processus d'authentification il est donc fortement conseillé d'utiliser [des bibliothèques standard d'authentification OAuth 2.0](#). Par exemple :

- Javascript : [oauth](#).
- .NET : [Microsoft.Owin.Security.OAuth](#).

L'*access token* ainsi récupéré permet d'accéder aux ressources fournies par l'API via [un en-tête d'authentification de type Bearer](#). Exemple :

Authorization: Bearer mF_9.B5f-4.1JqM

La plupart des méthodes d'authentification disponibles nécessitent le transfert des identifiants de votre application : ces identifiants sont l'équivalent d'un couple login / mot de passe et ils ne doivent donc **jamais** être visibles par les utilisateurs de votre application (dans le code source javascript d'une application web par exemple). Les méthodes courantes pour l'éviter sont :

- l'utilisation d'un proxy (pour les applications web, donc).
- modifier les caractéristiques de l'application pour éviter d'avoir à transmettre ces identifiants (application utilisateur publique, cf. ci-dessous).

Dans certains cas un *refresh token* est fourni avec l'*access token*. Ce jeton peut être utilisé afin de renouveler simplement l'*access token*. Il doit donc être protégé, de la même manière que les identifiants d'application.

Applications utilisateur

Caractéristiques

Une application utilisateur possède :

- un nom.
- des identifiants.
- des URIs de redirection.
- un type.

Une application utilisateur accède en lecture seule à l'ensemble des fiches accessibles à l'utilisateur qui se connecte.

URIs de redirection

Les flots d'authentification disponibles nécessitent l'utilisation d'un navigateur (potentiellement embarqué pour une application de bureau) et se basent sur un principe de redirections ([HTTP 302](#)) pour fournir un code d'autorisation à l'application. Dans un souci de sécurité il est donc nécessaire d'enregistrer préalablement la ou les URIs de redirection qui correspondent au(x) domaine(s) de votre application. Les URIs enregistrées doivent correspondre exactement aux routes acceptées par votre application.

Dans le cas d'un développement étalé sur plusieurs plateformes (par exemple développement, intégration, pré production, production), il est conseillé d'enregistrer les URLs spécifiques de chaque plateforme (dans la même application).

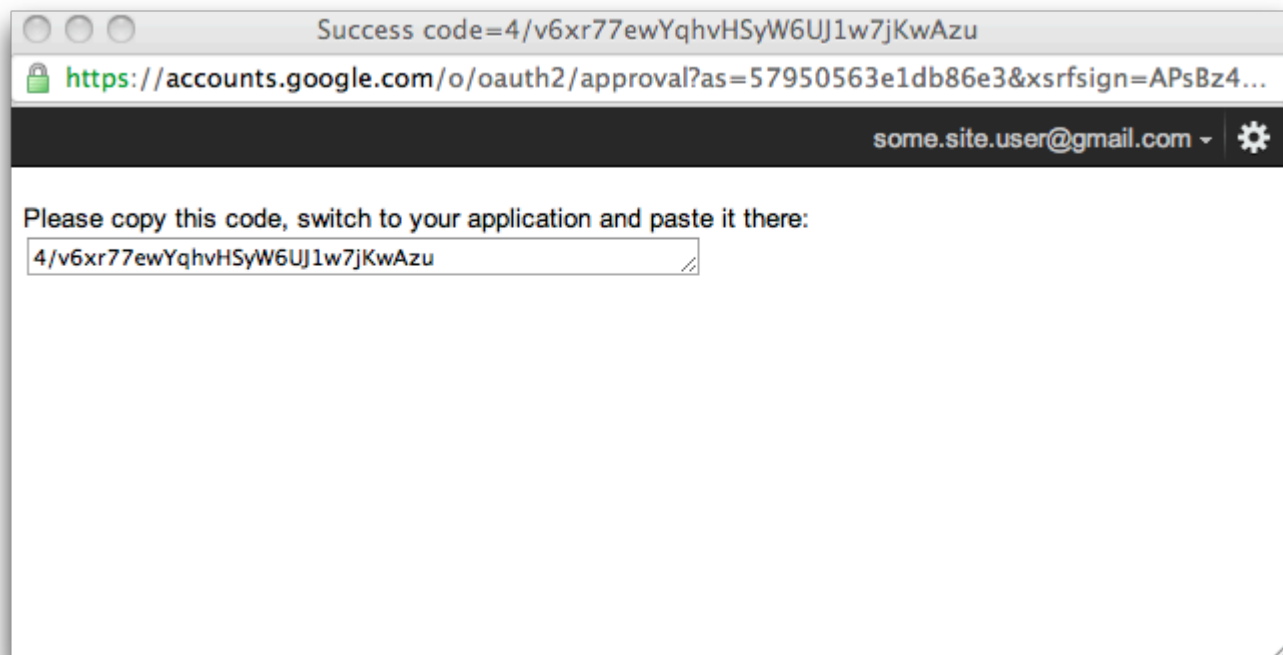
Dans le cas d'une application native (application de bureau, ou mobile par exemple) ce mécanisme de redirections nécessite l'appel à un navigateur externe (Internet Explorer, Firefox, Chrome...) ou à un navigateur embarqué. La récupération du code d'autorisation peut alors se faire :

- via une redirection vers un serveur local instancié par l'application sur un port dédié (<http://localhost:9873> par exemple). Cette méthode est parfaitement valide, mais peut s'avérer lourde à l'usage.
- via l'utilisation d'URIs prédéfinies qui fournissent ce code au navigateur et qui permettent à une application qui maîtrise le navigateur qu'elle a créé de récupérer ce code directement dans le navigateur :
 - assez simple en général dans le cas d'un navigateur embarqué.
 - potentiellement plus complexe dans le cas d'un navigateur externe (sous Windows utilisation des APIs Win32 [ShellExecute](#) ou [ShellExecuteEx](#) par exemple).

Les URIs prédéfinies (qui doivent quand même être enregistrées application par application) sont :

- `urn:ietf:wg:oauth:2.0:oob` : affiche une page qui permet à l'utilisateur de copier/coller le code d'autorisation depuis le navigateur vers l'application.
- `urn:ietf:wg:oauth:2.0:oob:auto` : renseigne le code d'autorisation dans le titre de la page, ce qui permet à une application intelligente de le récupérer automatiquement avant de fermer (automatiquement si possible) la fenêtre du navigateur.

Ce système est identique à celui utilisé par Google.



Type

Une application utilisateur peut être de 2 types :

- *confidentielle* : une application qui est capable de stocker et de transmettre ses identifiants en toute sécurité (sans les dévoiler à l'utilisateur).
 - par exemple (et typiquement) une application web, qui stocke les identifiants sur son serveur (dont l'accès est normalement restreint) et qui les transmet sans passer par le navigateur.
 - une application *confidentielle* utilise le flot [Authorization Code Grant](#) pour permettre à l'utilisateur de s'authentifier à la plateforme Isogeo.
- *publique* : une application qui est incapable de stocker ou de transmettre ses identifiants sans les dévoiler à l'utilisateur.
 - par exemple une application qui s'exécute entièrement dans le navigateur (application en javascript en général) et dont le code source doit être entièrement téléchargé par le client.
 - une application *publique* utilise le flot [Implicit Grant](#) pour permettre à l'utilisateur de s'authentifier à la plateforme Isogeo.

Le niveau de confidentialité à affecter à une application utilisateur native (application de bureau ou application mobile par exemple) est un problème non encore résolu dans le monde de OAuth :

- en principe elle doit être *publique* (les identifiants devraient être fournis avec l'application qui est téléchargée, donc en pratique visible par tous même si cela requiert parfois des techniques plus ou moins avancées de décompilation).
- en pratique est parfois *confidentielle* afin principalement d'améliorer l'expérience utilisateur.

Pour plus d'informations il est possible de se référer à la RFC en cours de rédaction sur la sécurisation des applications natives : [OAuth 2.0 for Native Apps](#).

Authorization Code Grant

La documentation officielle de ce flot est disponible [dans la RFC 6749](#).

Requête d'autorisation

La récupération du code d'autorisation se fait via un navigateur (le navigateur en cours d'utilisation dans le cas d'une application web, un navigateur externe ou un navigateur embarqué dans le cas

d'une application native). Le client doit être redirigé sur la route d'autorisation <https://id.api.isogeo.com/oauth/authorize> :

- la requête est un GET vers <https://id.api.isogeo.com/oauth/authorize>
- avec des paramètres qui indiquent :
 - `response_type` : `code`
 - `client_id` : l'identifiant OAuth de l'application.
 - `redirect_uri` : l'adresse de l'application vers laquelle renvoyer le code d'autorisation (préalablement enregistrée dans ses paramètres, par exemple <https://app.isogeo.com/login/callback>).

En cas de succès le navigateur est redirigé vers la route de l'application spécifiée ci-dessus, avec les paramètres :

- `code` : le code d'autorisation.

Demande du jeton

Muni du code d'autorisation précédent, la récupération d'un *access token* se fait sur la route <https://id.api.isogeo.com/oauth/token>. Donc :

- la requête est un POST vers <https://id.api.isogeo.com/oauth/token>
- avec un contenu qui contient les paramètres :
 - `grant_type` : `authorization_code`
 - `code` : le code d'autorisation.
 - `redirect_uri` : la même adresse que pour la requête d'autorisation.
- avec [un en-tête d'authentification de type Basic](#), où l'on considère que le nom d'utilisateur à encoder est le `client_id` et le mot de passe à encoder est le `client_secret` (ce qui revient à encoder en [Base 64](#) la chaîne `{client_id}:{client_secret}`, sans les accolades). Exemple :
Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW

L'*access token* est renvoyé au format JSON. Il permet l'accès aux ressources d'Isogeo en lecture seule et est valide pendant 1 heure.

Un *refresh token* est également fourni.

Implicit Grant

La documentation officielle de ce flot est disponible [dans la RFC 6749](#).

La récupération d'un *access token* se fait via un navigateur (le navigateur en cours d'utilisation dans le cas d'une application web, un navigateur externe ou un navigateur embarqué dans le cas d'une application native). Le client doit être redirigé sur la route d'autorisation <https://id.api.isogeo.com/oauth/authorize> :

- la requête est un GET vers <https://id.api.isogeo.com/oauth/authorize>
- avec des paramètres qui indiquent :
 - `response_type` : `token`
 - `client_id` : l'identifiant OAuth de l'application.
 - `redirect_uri` : l'adresse de l'application vers laquelle renvoyer le code d'autorisation (préalablement enregistrée dans ses paramètres, par exemple <https://app.isogeo.com/login/callback>).

En cas de succès le navigateur est redirigé vers la route de l'application spécifiée ci-dessus, avec les paramètres ajoutés en [fragment](#) :

- `access_token` : le jeton d'accès.
- `token_type` : `bearer`
- `expires_in` : la durée de validité du jeton, en secondes.

Applications de groupe

Caractéristiques

Une application de groupe possède :

- un nom.
- des identifiants.

Elle peut être associée à 0, 1 ou plusieurs groupes de travail qui peuvent ainsi lui partager des catalogues.

Une application de groupe accède en lecture seule à l'ensemble des fiches contenues dans l'ensemble des catalogues qui lui sont partagés. En revanche elle n'a pas directement accès à ces catalogues : la notion même de catalogue n'a pas de sens pour une application de groupe.

Une application de groupe utilise le flot [Client Credentials Grant](#) pour s'authentifier à la plateforme Isogeo.

Client Credentials Grant

La documentation officielle de ce flot est disponible [dans la RFC 6749](#).

Pour paraphraser, la récupération d'un *access token* se fait sur la route <https://id.api.isogeo.com/oauth/token>. Donc :

- la requête est un POST vers <https://id.api.isogeo.com/oauth/token>
- avec un contenu qui indique
`grant_type=client_credentials`
- avec [un en-tête d'authentification de type Basic](#), où l'on considère que le nom d'utilisateur à encoder est le *client_id* et le mot de passe à encoder est le *client_secret* (ce qui revient à encoder en [Base 64](#) la chaîne `{client_id}:{client_secret}`, sans les accolades). Exemple :
Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW

L'*access token* est renvoyé au format JSON. Il permet l'accès aux ressources d'Isogeo en lecture seule et est valide pendant 1 heure.

Un *refresh token* est également fourni.

Refresh Token

La documentation officielle de l'utilisation d'un *refresh token* est disponible [dans la RFC 6749](#).

Pour paraphraser, le renouvellement d'un *access token* se fait sur la route

<https://id.api.isogeo.com/oauth/token>. Donc :

- la requête est un POST vers <https://id.api.isogeo.com/oauth/token>
- avec un contenu qui indique :
 - `grant_type:refresh_token`
 - `refresh_token`: le refresh token précédemment récupéré
- avec [un en-tête d'authentification de type Basic](#), où l'on considère que le nom d'utilisateur à encoder est le *client_id* et le mot de passe à encoder est le *client_secret* (ce qui revient à encoder en [Base 64](#) la chaîne `{client_id}:{client_secret}`, sans les accolades). Exemple :
Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW

Proxy

Dans le cas d'une application web typiquement, l'utilisation d'un proxy est nécessaire :

- afin d'autoriser et de sécuriser les appels vers l'API Isogeo.
- afin de rendre accessible à l'utilisateur certaines ressources particulières, images ou fichiers, en téléchargement par exemple (fiche de métadonnée ISO 19139, fichiers hébergés sur la plateforme Isogeo...).

API Isogeo

Dans le cas d'une application web dynamique qui réalise des appels vers l'API Isogeo via le client (le navigateur), ces appels doivent être réalisés via un proxy spécifique à l'application. Le rôle de ce proxy est multiple :

- l'API Isogeo ne supporte pas les appels [Cross Origin](#). L'utilisation d'un proxy est donc obligatoire.
- le proxy protège les identifiant/secret d'application :
 - les identifiant/secret d'application qui permettent de récupérer un token doivent être protégés comme un login/mot de passe : le seul véritable moyen est de les stocker derrière un serveur, comme le font App ou OpenCatalog.
- le proxy protège le token, qu'il rajoute à chaque requête et peut renouveler automatiquement avant expiration :
 - ce token doit également être protégé et confidentiel : le seul véritable moyen est de le stocker derrière un serveur, comme le font App ou OpenCatalog (via un mécanisme de [session](#)).

Dans le cas d'une application web qui réalise tous les appels côté serveur pour générer des pages web complètes, l'utilisation d'un proxy n'est pas nécessaire.

La forme des URLs gérées par ce proxy est à la discrétion du développeur. Par exemple pour l'application Isogeo, le proxy est à l'adresse <https://app.isogeo.com/api/v1> et la requête <https://app.isogeo.com/api/v1/groups/643f1035377b4ca59da6f31a39704c34/resources/search> réalise côté serveur un appel sécurisé vers la route <https://v1.api.isogeo.com/groups/643f1035377b4ca59da6f31a39704c34/resources/search>.

Afin d'éviter les possibilités d'attaques ([DoS](#) typiquement) ou de fuite d'informations confidentielles (si votre application est réservée à un public restreint), il est de la responsabilité du développeur de vérifier que son application est la seule autorisée à réaliser des appels vers son proxy.

Fichiers

La diffusion de certaines ressources (images ou fichiers à télécharger) doivent être réalisées derrière un proxy (en général différent du précédent). Ces ressources sont :

- les fichiers hébergés sur la plateforme Isogeo (*links* de type *hosted*).
- les fiches XML ISO 19139 à télécharger par le client.
- les vignettes.

En plus des raisons d'autorisation et de sécurité générales (sécurisation des identifiants et des tokens, gestion du [Cross Origin](#)), ce proxy répond à des problématiques supplémentaires : les URLs qui permettent d'afficher une image ou de télécharger un fichier apparaissent en clair dans le code HTML d'une page web et sont directement gérées par le navigateur. L'utilisation d'un proxy est donc

nécessaire, y compris pour une application web qui réalise tous les appels côté serveur pour générer des pages web complètes.

La forme des URLs gérées par ce proxy doit être conforme au schéma suivant :

- l'URL absolue de base reste à la discrétion du développeur : <https://app.acme.org/p> par exemple.
- les URLs relatives sont du type :
 - `/resources/{:rid}.xml` pour renvoyer une fiche XML en ISO 19139.
 - `/resources/{:rid}/links/{:lid}.bin` pour renvoyer un fichier hébergé sur la plateforme Isogeo (URL relative renvoyée par l'API Isogeo pour les *links* de type *hosted*).
 - `/resources/{:rid}/thumbnail/{:tid}.png` pour renvoyer la vignette de la ressource (URL relative renvoyée par l'API Isogeo).

Dans cet exemple, la route

<https://app.acme.org/p/resources/374e6188c5a3e311bf6400155d000b06/links/19532aeca11a4efdaaf97760982364f2.bin> transfèrera le fichier renvoyé par l'API <https://v1.api.isogeo.com/resources/374e6188c5a3e311bf6400155d000b06/links/19532aeca11a4efdaaf97760982364f2.bin>.

Dans le cas où la confidentialité des données n'est pas un problème (portail public par exemple) il est possible d'ajouter (côté serveur) le paramètre *proxyUrl* à la route qui génère la fiche XML (<https://v1.api.isogeo.com/resources/374e6188c5a3e311bf6400155d000b06.xml?proxyUrl=https://app.acme.org/p> par exemple) afin que la fiche soit générée avec des liens vers les fichiers disponibles.

Le développeur d'une application web est responsable d'assurer la sécurité de son proxy contre des possibilités d'attaques ([DoS](#) typiquement). Dans le cas d'une application réservée à un public restreint, il est également responsable d'assurer la confidentialité de son proxy ([vis à vis des moteurs de recherche](#) par exemple).

Développement

Toutes les plateformes modernes de développement web offrent des moyen simples de réaliser un proxy, et il est donc conseillé de les utiliser. Par exemple :

- node.js : [node-http-proxy](#).

Les risques liés au développement d'un proxy sont liés à la confidentialité des données partagées à vos applications : si vous devez vérifier que seule l'application peut faire appel à son proxy (quelques techniques sont disponibles ici : https://en.wikipedia.org/wiki/Cross-site_request_forgery#Prevention).

Dans la mesure du possible, nous serons reconnaissants à nos développeurs si leur proxy nous transmet des informations sur l'origine des requêtes qui sont transmises à l'API ([X-Forwarded-For](#) par exemple).

Annexes

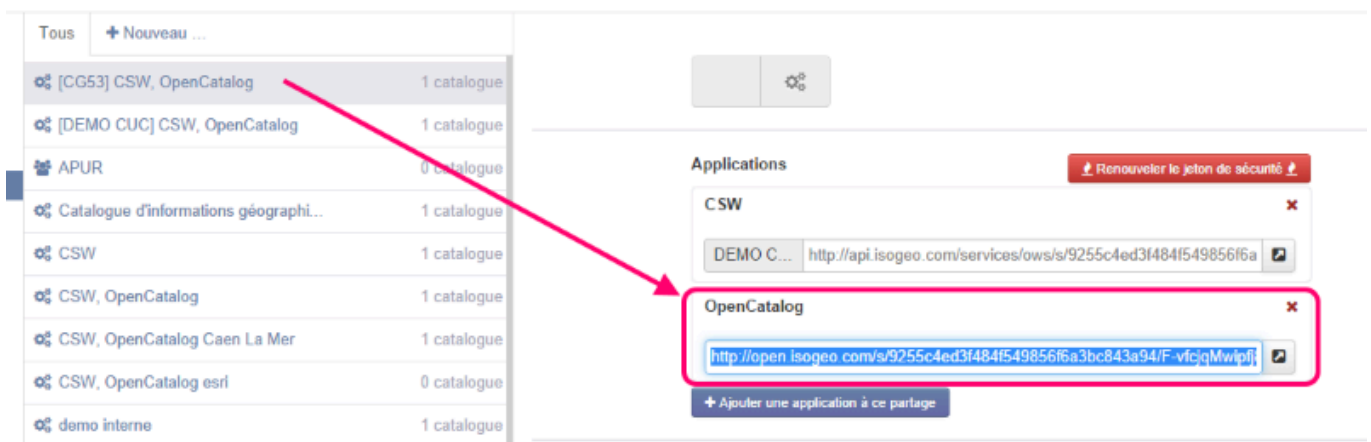
Webinaire de présentation et d'introduction à l'API (19 janvier 2016)

Voir la vidéo : <https://www.youtube.com/watch?v=sOv9LeMycbo>

Authentification via OpenCatalog [dépréciée]

Système maintenu pour des raisons historiques et documenté pour les mêmes raisons. Cette méthode sera désactivée à une date qui reste encore à préciser. Ne pas utiliser.

L'idée est de créer un [partage OpenCatalog dans App](#) et de récupérer le *token* : l'adresse est du genre `http://open.isogeo.com/s/identifiant_du_partage/token`



Il s'agit d'isoler l'*identifiant* et le *token*

Il suffit ensuite de rajouter ces paramètres aux requêtes à l'API de la manière suivante :

Pour utiliser l'API (/resources/search par exemple), ajouter les paramètres suivants :

- **token:** *token*
- **s:** *identifiant du partage*

L'URL finale aura donc la forme suivante :

<https://v1.api.isogeo.com/resources/search?token={token}&s={identifiant-du-partage}>