

# KOHA : LES WEB SERVICES AUX LECTEURS

Corinne Hayet



# Principes généraux

- ✦ Qu'est-ce qu'un web service ?
- ✦ API et web services



## Les API Koha

Plusieurs API mises à disposition par Koha permettent d'accéder à ses données ou à certaines de ses fonctionnalités

- OAI-PMH
- SRU/Z3590
- JSON
- SVC
- ILS-DI
- REST FULL



## SVC

- Historiquement la plus ancienne
- Développée par Biblime
- Permet de récupérer des enregistrements au format marcxml
- Discussion sur remplacement depuis 2014
- Ne respecte pas totalement le HTTP et ses « verbes » (discussion autour de POST et PUT)
- Limité fonctionnellement (recherche on passe par API Zebra et non SVC)
- Étendre les fonctionnalités proposées
- Question de la compatibilité ascendante avec certaines API l'utilisation (OCLC, MarcEdit)



## ILS-DI

- 📄 ILS Discovery Interface
- 📄 Prend la suite de SVC
- 📄 C'est ce qui est actuellement le « plus utilisé »
- 📄 Ce que cela permet : <http://catalogue.demo1811-koha.test.bibliobre.eu/cgi-bin/koha/ilsdi.pl>
- 📄 Les limites : accès sécurisé via préférences système pour limiter l'appel à certaines adresses IP mais pas de notation CDIR (fichier de configuration Apache pour contourner le problème)
- 📄 Les limites : appel en deux temps des WS : liste des références puis disponibilité des exemplaires
- 📄 Les limites : pas toutes les fonctionnalités (exemple : les listes publiques de Koha)



## REST FULL (1)

- 🖥️ Une API rest full est basée sur le protocole http
- 🖥️ Elle utilise : une URI de base (URI d'appel), des méthodes (fonctions) http (get, post, patch, delete)
  - GET : Récupération d'une ressource
  - POST : Ajout d'une ressource
  - PUT : Mise à jour complète/partielle d'une ressource
  - DELETE : Suppression d'une ressource
- 🖥️ Elle s'appuie sur un « format » : atom, json...
- 🖥️ Premiers développements effectués par certains prestataires (Biblibre, Progilone) pour pallier les manques de ILS-DI, SVC
- 🖥️ Suite aux premières discussions entamées en 2014, décision de mettre en place une API communautaires versionnée respectant les standard HTTP



## REST FULL (2)

- 📖 Qu'est-ce que cela permet actuellement ?
- 📖 Première version de l'api : mon-opac/api/v1 (exemple : <http://catalogue.demo1811-koha.test.bibliobre.eu/api/v1/>)
- 📖 <https://demo.bibkat.no/api/v1/.html> (documentation en format HTML plus lisible)
- 📖 Beaucoup de nouvelles fonctionnalités ajoutées depuis la 17.11 qui viennent pour l'instant compléter/remplacer ce qui est proposé par ILS-DI
  - Gestion des fournisseurs (17.11)
  - Suppression, mise à jour, ajout d'utilisateurs – authentification OAuth 2 – gestion des clés API (18.05)
  - Gestion des amendes – utilisation de plugins pour ajouter de nouvelles API (18.11)
  - Prêts et renouvellements - gestion des sites, changement du mot de passe utilisateur (API privée, API publique) – gestion des budgets – gestion des suspensions de demandes de réservations (19.05)



## REST FULL (3)

- 📄 Les pistes d'évolutions, discussions en cours
  - Amélioration de la sécurisation des API
    - Clé publique, clé privée (réorganisation des API dans ce sens)
    - Notation CIDR serait aussi un « plus » ?
  - Ajouter l'ensemble des fonctionnalités ajout – suppression – modifications des notices Koha (gestion des enregistrements en dehors de l'interface professionnelle de Koha)
  - Extension via des plugins
  - Gestion des suggestions d'achat (bgz 17314)
  - Gestion des commandes (bgz 18731)...
- 📄 Plus d'informations sur : [https://wiki.koha-community.org/wiki/REST\\_api\\_RFCs](https://wiki.koha-community.org/wiki/REST_api_RFCs)





## REST FULL (4)

### Exemples Progilone

Pour les résultats de recherche, on passe directement par solr, mais pour accéder à une notice on utilise

- `/biblio/#biblionumber` en GET avec `#biblionumber` qui est le numéro de la notice.

### Accès au compte lecteur

- `/borrower/checkpassword` : POST avec, comme donnée, un json avec les clés 'username' et 'password'

### Récupération d'une liste publique

- `/borrower/#borrowernumber/list/#shelfnumber` avec `#borrowernumber` à 0 pour une liste publique et `#shelfnumber` l'identifiant de la liste