Formation Kohala -   
Personnalisation de l'OPAC Koha

Frédéric Demians - Tamil

**http://koha-fr.org/formations-2019/ ---> remplir le questionnaire d'évaluation !**

Version 17.11 - Emmanuel Betemps (BM Caluire et Cuire)

Version 17.11 - Sylvain Deflache (BM Caluire et Cuire)

Version 18.05 - Aurélie Bérut (ENSSIB)

Version 17.11 - Roméo Isarte (ENTPE) + s'occupe d'une association avec Koha 18.11, sans prestataire

Version 3.xx : ???

OPAC précédemment géré par des feuilles de style

En 18.05, géré différemment

**Langages à utiliser pour faire tout ça : HTML, CSS, Javascript / Jquery, JSON, XSLT**

Architecture de Koha

Serveur et interface professionnelle web

Accès possible en dehors de l'interface web

Koha est une grande base de données, accessible avec des scripts Pearl, avec des services web en dehors de l'interface web de Koha (avec le protocole HTTP), accès aux données brutes avec des interrogations SQL.

Le cœur de Koha : des notices bibliographiques accessibles via des tables de données (biblioitems, biblio, items en 17.11)

Base de données relationnelle avec des tables (lignes, colonnes)

emprunteurs = borrowers

notices bibliographiques stockées au format MarcXML pas évident à lire car possédant des champs répétables (ex. champs 700, 200) et des sous-champs eux aussi répétables

Avantage de Koha : les données bibliographiques sont toutes dans le champ MarcXML ou dans la table biblio.metadata (à partir de la version 17.11)

Un outil tiers est nécessaire pour interroger ces données brutes (Zebra ou ElisaticSearch pour Koha) L'indexeur normalise, classe, flèche les données bibliographiques.

La notice, une fois enregistrée, doit être traitée par Zebra ou Elisaticsearch pour être trouvée. Elisaticsearch est plus rapide que Zebra pour ce traitement.

Les résultats de recherche affichés dans l'OPAC ou sur l'interface de recherche sont le résultat d'un traitement par Zebra ou Elasticsearch.

**LES NOTICES DE KOHA SONT ENREGISTRÉES AU FORMAT MARCXML**

notice XML

<?xml version ?xxx> début d'une notice XML

ISO2709 / XML / JSON sont des formats de notices

<leader></leader>

<controlfield></controlfield>

<datafield tag="090" </datafield>

<subfield code=9> </subfield>

# XSLT

Passage du MARCXML au HTML

Feuille de style standard fournie avec Koha

### 4 feuilles de style principales sous Koha

XSLTDetaildisplay : page de détail

XSLTListsDetaildisplay : page des listes de Koha

XSLTResultdisplay : page de résultat

Page par défaut sur le serveur Koha

UNIMARCDetailKoha

Chemin dans le Koha classique :

/home/koha/webdatas/public/xslt/

--> demander un accès à ces pages au prestataire (ex.Biblibre, Tamil)

Export de notices sur Koha

MARC = ISO2709

XML = Format MARCXML

CSV = Format dégradé

En plus dans Koha les données sont mappées dans la base SQL

ex. le champ titre correspond au champ 200$a mais également au mapping biblio.title

La table biblio / items / biblioitems est un moyen plus facile d'accéder aux données que de chercher directement dans le fichier MarcXML.

# OPAC Koha

--> Préférences de l'OPAC

Accueil ›Administration › Préférences système > Préférences OPAC

Depuis la 18.11, il est possible de stocker de manière propre les images sur le serveur Koha pour faire afficher des images sur l'interface professionnelle et/ou sur l'OPAC (auparavant, les images étaient stockées DANS la base SQL, ce qui risquait de la faire grossir de façon exponentielle...

Possibilité de passer par un mandataire pour récupérer les images de couverture

(ex. http://coce.tamil.fr) qui permet de cherche des images à plusieurs endroits (Amazon, Google books, OpenLibrary)

Open Library est un projet de stockage d'images libre dont le projet est arrêté.

Types de document définis soit au niveau de la notice biblio soit au niveau de la notice exemplaire

Accueil ›Administration › Préférences système > Préférences Catalogage > Structure des notices

Structure des notices : **item-level\_itypes**

Attention, cette définition est aussi utilisée pour gérer le type de prêt

Champs affichés pour recherche professionnelle :

Accueil ›Administration › Préférences système > Formulaire de recherche > AdvancedSearchTypes

Actuellement paramétré sur le Koha de Caluire :

**type\_doc|loc|instrument**

avec :

**type\_doc** : type de document utilisé pour le prêt

**loc** : localisation de l'exemplaire

**instrument** : instrument de musique

Pour faciliter la modification du code HTML

Accueil ›Administration › Préférences système > UseWYSIWYGinSystemPreferences

(déjà activé sur le Koha Caluire, mais ne fait pas apparaître grand-chose...)

Accueil ›Administration › Préférences système > Interface professionnelle

# CSS

Utiliser les outils pour développeurs des navigateurs

**Touche F12** OU **Outils / Développement web / Outils de développement**

Firefox

Chrome

Connaître la syntaxe propre au CSS

Exe

.label {font weight : bold}

*Met en gras tous le texte signalé sous l'étiquette label*

Sur l'interface de développement sous Chrome ou Firefox

- affichage des styles et des empilements de style utilisés dans la zone "Style" des outils de développement.

Les règles sont affichées de haut en bas, de la règle prioritaire à la règle à la plus faible priorité.

- l'interface de développement propose lorsque l'on rajoute des éléments de style une saisie semi-automatique

- utiliser les code HTML le plus brut possible dans Koha et ensuite les mettre en forme avec les outils propres au navigateur

- utiliser des polices différentes est un bon moyen de changer rapidement l'aspect d'un site

https://fonts.google.com/

Annihiler l'affichage ISBD

Cacher certains affichages sur l'interface

ID --> si on traite d'un ID, syntaxe à utiliser

#nomdel'ID (dièse)

display : none

Classe si on traite d'une classe

.nomdelaClass (point)

display : none

# Javascript

pour modifier le code DÉJÀ dans Koha

pour afficher une image particulière en fonction de l'utilisateur connecté

### 1ère solution

Modification du programme Pearl spécifique à une page, qui construit chaque page HTML de Koha

detail.pl : détails d'une notice

search.pl : liste de résultats d'une recherche biblio

member.pl : liste d'abonnés

moremember.pl : détail d'une notice d'abonné

MAIS en cas de mise à jour de la donnée de Koha, ces modifications sur les templates sont perdues...

MAIS il faut modifier les templates sur la version anglaise et la version française...

MAIS il faut avoir l'accès aux fichiers Pearl sur le serveur

### 2ème solution

Manipulation HTML en Javascript ou JQuery (sur-langage de Javascript)

(C’est ce qu'on a utilisé à Caluire et Cuire)

**jQuery Tutorial**

<https://www.w3schools.com/jquery/default.asp>

WikiKoha pour des exemples de jQuery

<https://wiki.koha-community.org/wiki/JQuery_Library>

**jQuery - Les bases du débutant (96 p.)**

<https://www.decitre.fr/ebooks/jquery-9791034805761_9791034805761_2.html>

**jQuery - Pense-bête**

<https://www.decitre.fr/ebooks/jquery-9782212097276_9782212097276_11.html>

### Syntaxe basique

**$(zoneàmodifier).*action*()**

**$("p").hide() - cache tous les éléments <p>**

**$(".test").hide() - cache tous les éléments avec la class="test".**

**$("#test").hide() - cache tous les éléments avec l'id="test".**

### Document ready

**$(document).ready(function(){**

***// jQuery methods go here...***

**});**

Cette syntaxe permet d'empêcher d'exécuter la fonction AVANT que toute la page ne soit prête à être affichée (afin de pouvoir appliquer les modifications correctement)

Le code JavaScript est ainsi prêt à être exécuté sur la page AVANT que celle-ci ne soit affichée.

### Texte à modifier, contenu dans une classe particulière

**$(document).ready(function(){**

**$("td[class='ci-shelvinglocation']:contains('Réserve')" ).css({"color": "white","background-color": "#FF0040","font-weight": "bold"})**

**});**

Recherche la classe **"**ci-shelvinglocation" dans un tableau (TD) contenant le mot "Réserve"

et colorie le texte en BLANC, sur fond rouge, avec des caractères en gras

### Texte dans un bouton à effacer

**$(document).ready(function(){**

**$('button:contains("Imprimer un ticket et confirmer")').hide();**

**});**

$('#zoneà modifier).html('texteplacédanslazoneamodifier);

**Syntaxe du JSON / JQuery à apprendre et à connaître...**

**if ( $ body and id=opacmain....**

**pour limiter la modification à une page spécifique....**

# Rapport SQL à utiliser sur un site web

Cocher l'option "rapport public" dans la création du rapport SQL

Utiliser l'URL du Koha de l'établissement

Applications Web permettant de mettre en forme un tableau

JSON beautifier

# Plugin KPZ

(Koha Plugin Zip)

Attention aux problèmes de sécurité que leur installation peut engendrer...

# Changer un intitulé dans l'interface pro

Ex. changer l'intitulé "réserve de cours"

Recherche dans le code avec Firefox

Recherche de l'ID

recherche dans la liste numérotée

document ready fonction

$('EMPLACEMENT').html('<br>CONTENU HTML AVEC SES BALISES</br>');

$('EMPLACEMENT').hide

# Ajout de champ (transition bibliographique)

Accueil ›Administration ›Grilles de catalogage MARC

Le champ 219 est appelé à remplacer le champ 210 (BnF)

Ajout d'un champ MAIS ceci ne suffit pas : il faut que le champ puisse ET être affiché en résultat de recherche ET que l'on puisse le rechercher (donc indexation...)

# Visibilité du catalogue en ligne

Recherches par Google dans le catalogue de la bibliothèque

Penser à faire un sitemap du catalogue de la bibliothèque, ce qui permettra au moteur de Google de chercher rapidement toutes les notices plutôt que de parcourir le catalogue au petit bonheur (web "gris" qui n'est pas très visible par les moteurs de recherche...)

À demander à Biblibre/prestataire