Diffusion de données depuis une base XML - Projet OiDB

Retour d’expérience

Patrick Bernaud (JMMC)

Plan

- eXist-db - base de données XML
- OiDB - présentation

eXist-db - Introduction

Base de données orientée document développée depuis 2001 sous LGPL 2.1 (http://www.exist-db.org/).

Version 2.2 sortie la semaine dernière.

- amélioration performances
- mise à jour modèle de sécurité
- nouveau indexes

eXist-db - Caractéristiques générales

Écrit en Java.

Jeu de servlets nécessitant un conteneur (Jetty).

Stocke des documents XML ou binaires organisés en collections dans un file system virtuel.

eXist-db - Caractéristiques générales (2)

- solution NoSQL / sans schéma
- XML
- rapid prototyping
- application web “embarquée” en XQuery

eXist-db - Interfaces

- WebDAV
- XML-RPC
- XML:DB
- SOAP
- REST

eXist-db - Index

Une recherche du genre `//foo[bar="xyz"]` nécessite scan complet de la collection → définition d’indexes adaptés à l’application.

Plusieurs types d’index : Range, Full Text, Spatial, xml:id, Structural...

Configuration par des documents XML dans chaque collection.

eXist-db - Index (example)

```
<collection xmlns="http://exist-db.org/collection-config/1.0">
    <index>
        <lucene>
            <text qname="SPEECH">
                <ignore qname="SPEAKER"/>
            </text>
            <text qname="TITLE"/>
        </lucene>

        <range>
            <create qname="title" type="xs:string"/>
            <create qname="author" type="xs:string"/>
            <create qname="year" type="xs:integer"/>
        </range>
    </index>
</collection>
```

eXist-db - XQuery

Langage programmation fonctionnelle orienté recherche/transformation/création documents structurés ou non structurés (extensions).

Recommandation W3C depuis 2007, version 3.0 - Avril 2014.

Similaire à XSLT sur l’aspect transformation, en moins verbeux.

Utile pour manipuler de larges collection de documents.

eXist-db - XQuery (2)

- High Order Function
- Pure function
- XPath
- FLWOR expression: FOR, LET, WHERE, ORDER BY, RETURN

eXist-db - XQuery (example)

```
for $book in doc("catalog.xml")/books/book
let $title := $book/title/text()
let $price := $book/price/text()
where xs:decimal($price) gt 50.00
order by $title
return
   <book>
      <title>{$title}</title>
      <price>{$price}</price>
   </book>
```

eXist-db - Services

- XUpdate (non standard)
- XSLT
- Security Manager
- Scheduler
- templating
- trigger
- RESTXQ
- XForms
- eXide

eXist-db - Security Manager

File system-like, permissions sur le modèle UNIX.

ACL (Access Control List) pour gestion plus fines des droits.

Authentification par mot de passe, LDAP, OpenID, OAuth.

eXist-db - Scheduler

Permet d’éxécuter du code régulièrement, en tâche de fond à la manière d’un cron.

Scripts en XQuery ou en Java, avec droits administrateur.

eXist-db - Templating

Module permettant de scripter la vue : insertion, tests simples, substitution depuis le modèle.

⊖ Limité, verbeux.

eXist-db - Triggers

Exécution de code déclenchée sur des évenements sur les documents de la base.

Déclaration en XML, script en XQuery ou en Java.

eXist-db - RESTXQ

RESTXQ 1.0: RESTful Annotations for XQuery 3.0.

Jeu d’annotations à ajouter à une fonction XQuery (resource function).

“Routage” automatique, serialization automatique.

eXist-db - RESTXQ (example)

```
declare
    %rest:GET
    %rest:path("/oidb/keyword")
    %rest:query-param("q", "{$q}")
    %output:media-type("application/json")
    %output:method("json")
function kw:list($q as xs:string*) {
    let $all := collection($kw:keywords-uri)//keyword
    let $keywords := if (exists($q)) then
            $all[contains(upper-case(.), upper-case($q))]
        else
            $all
    return <keywords> {
        for $kw in $keywords
        return <json:value>{ $kw/text() }</json:value>
    } </keywords> };
```

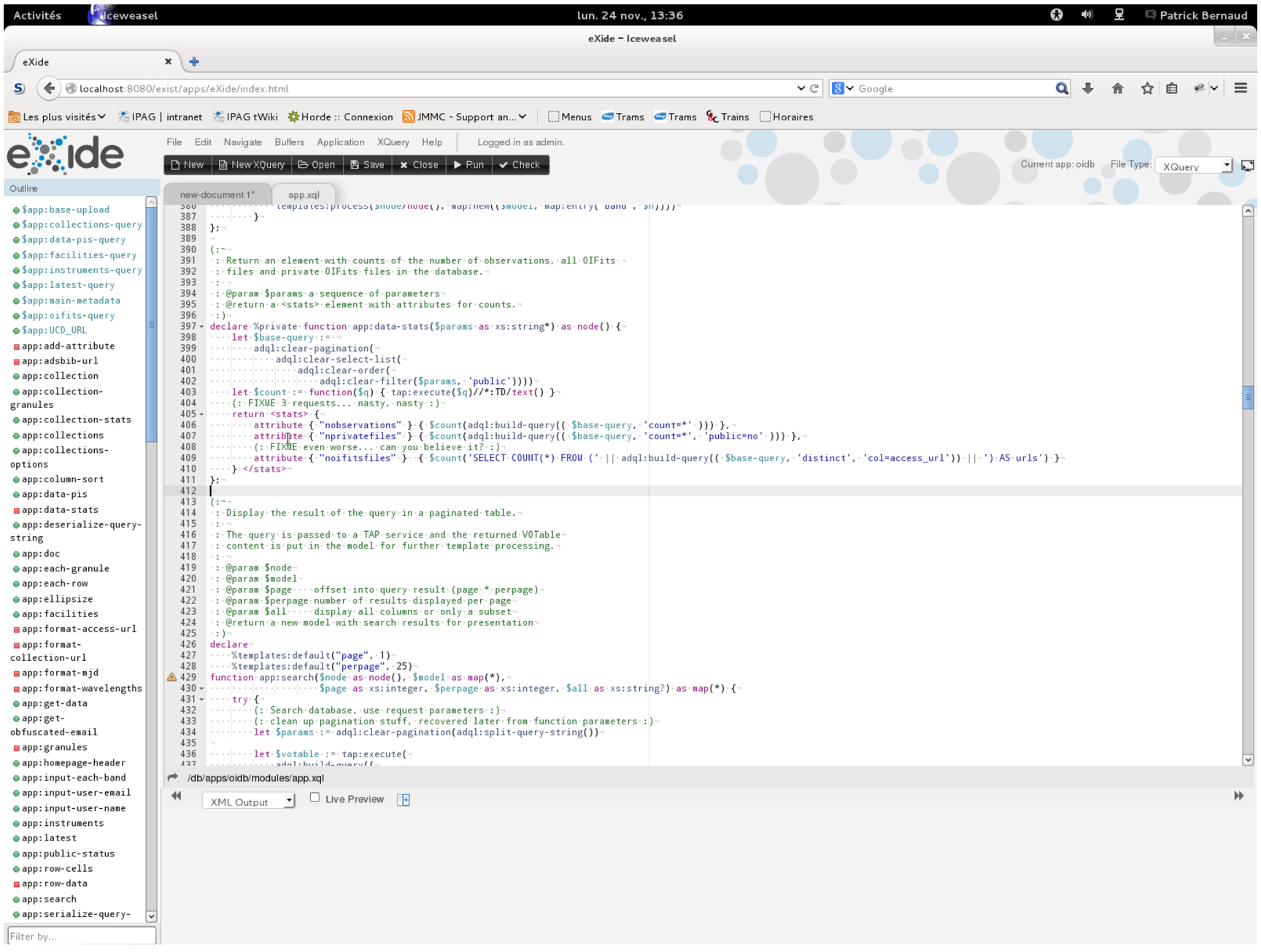
eXist-db - eXide

IDE dans le navigateur.

Fourni avec eXist-db.

Coloration syntaxique (XQuery, HTML, JavaScript), validation, exécution de code.

eXist-db - eXide (2)



eXist-db - Web Applications

Packaging de modules XQuery avec format prédéfini (MVC).

Serialization en HTML5

eXist-db - XForms

Solution générique d’écriture de formulaires.

Pas de support natif d’XForms dans les navigateurs → Betterform (server side) ou XSLForms (client side).

Échange de fragments XML avec eXist.

⊘ Intéressant mais mise en route difficile (eXist-db 2.1). À réévaluer.

OiDB

Portail de diffusion de metadonnées observationnelles en interférométrie optique avec interface Observatoire Virtuel.

Projet JMMC.

Début: décembre 2013 à partir prototype.

Prochainement version 1.0 en accès libre.

OiDB (2)

- extraction de metadonnées / indexation
- interface Web de consultation
- interface Web de soumission
- interface Web d’administration
- interopérabilité VO

OiDB - Data Model

ObsCore DM: modèle de données adapté aux observations en astronomie.

Ajout de colonnes spécifiques au domaine (extensions).

Définition d’une granule: une target, un instrument, une nuit.

OiDB - Interopérabilité

- ObsCore DM
- TAP = Table Access Protocol. Standard IVOA.
- Langage de requête (ADQL) proche du SQL.
- SAMP.
- Registry: TODO.

OiDB - Extraction

Formats multiples: OIFITS, observation logs.

Sources multiples: fichiers locaux, fichiers distants, base de données externes...

Pour les OIFITS, utilisation outils existants JMMC : écriture module interface/wrapper Java → XQuery.

OiDB - Data storage

Dual storage:

- SQL pour la table dérivée ObsCore
- XML pour les données textuelles.

Seules les données SQL sont requêtables en TAP.

OiDB - Authentification

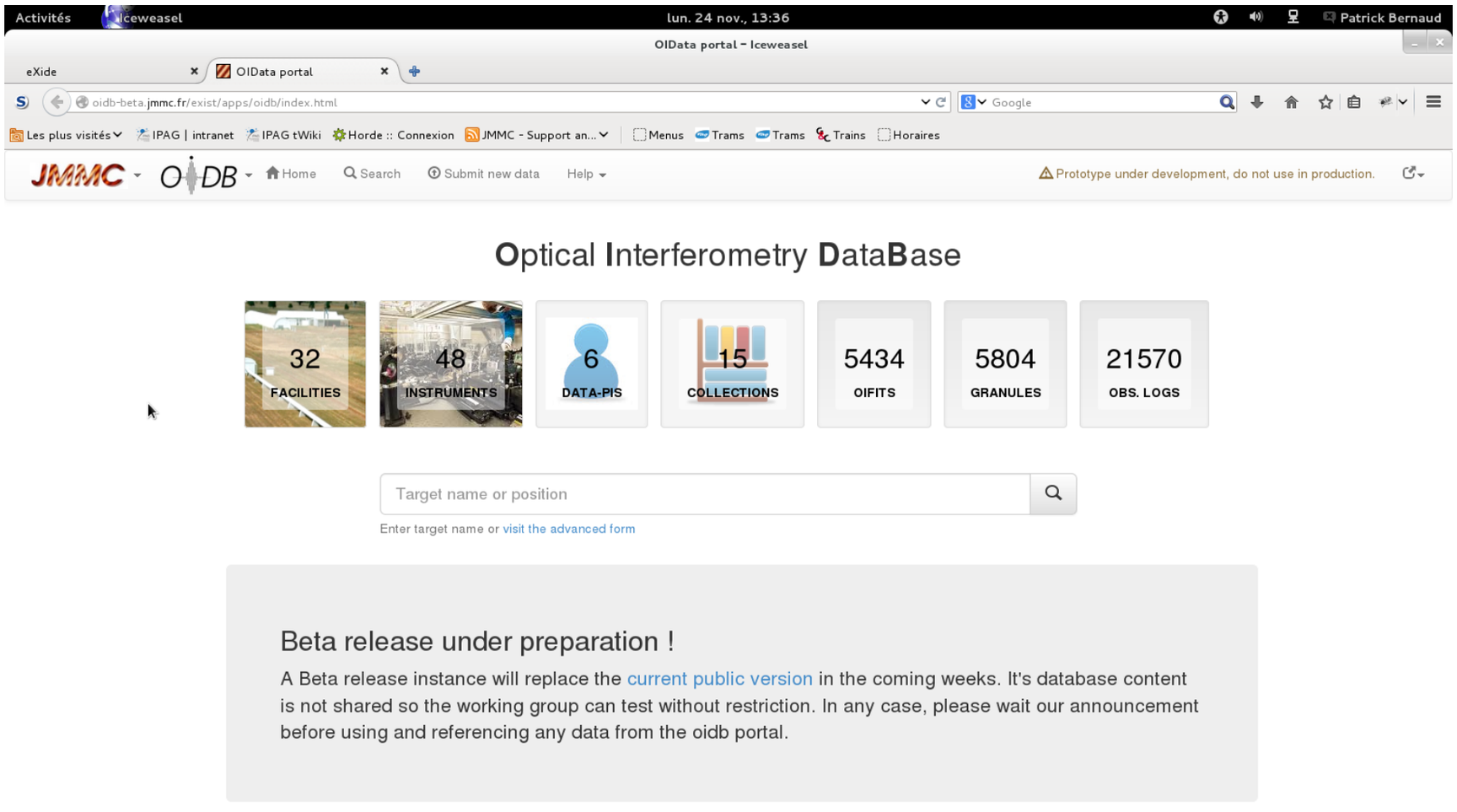
En attendant LDAP, écriture module interface avec la base utilisateur JMMC existante.

Utilisation du Security Manager intégré → Persistent login pour l’interface Web, HTTP Basic Auth pour le REST headless.

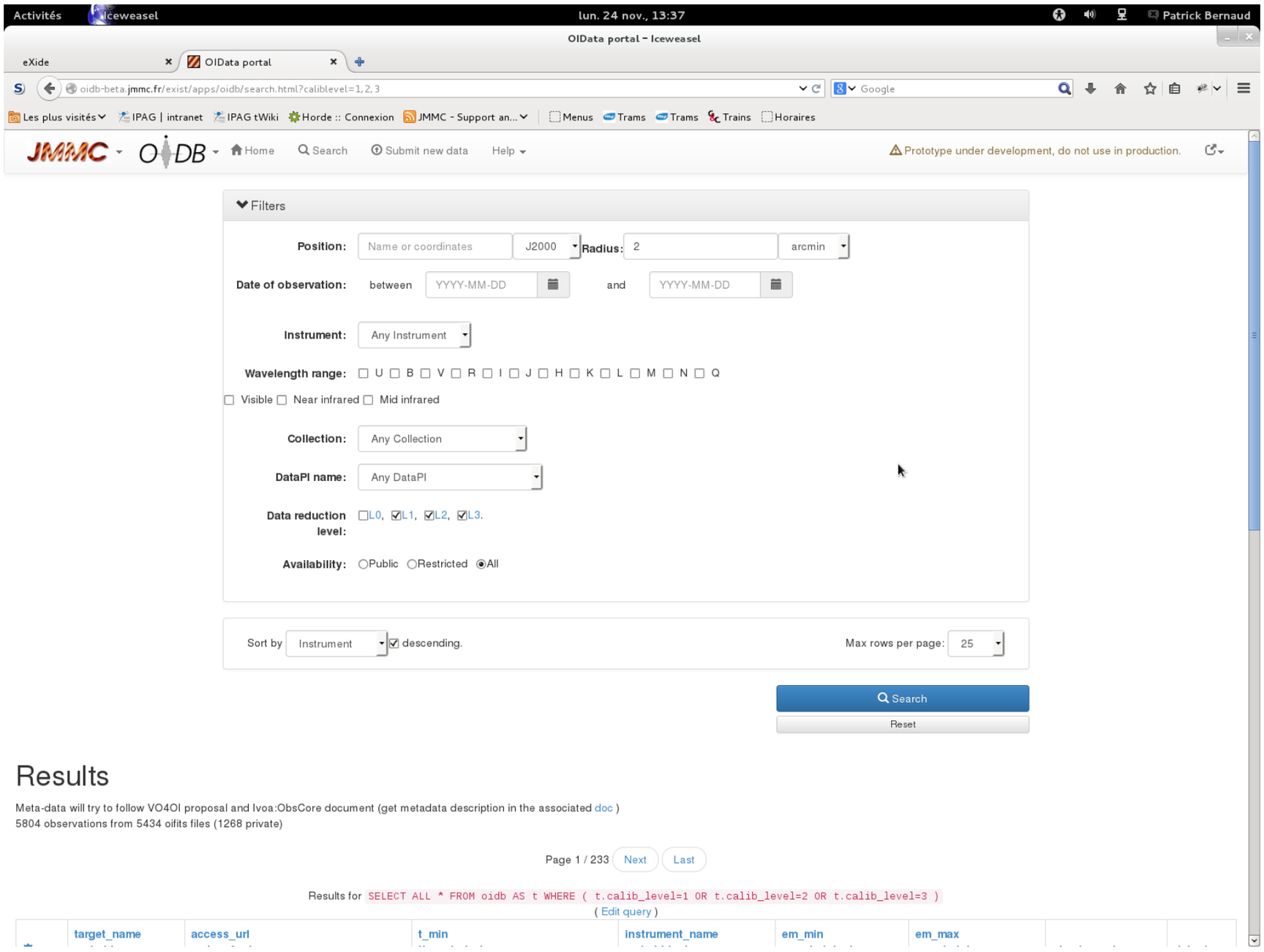
OiDB - Tools

- RESTXQ
- Bootstrap
- jQuery

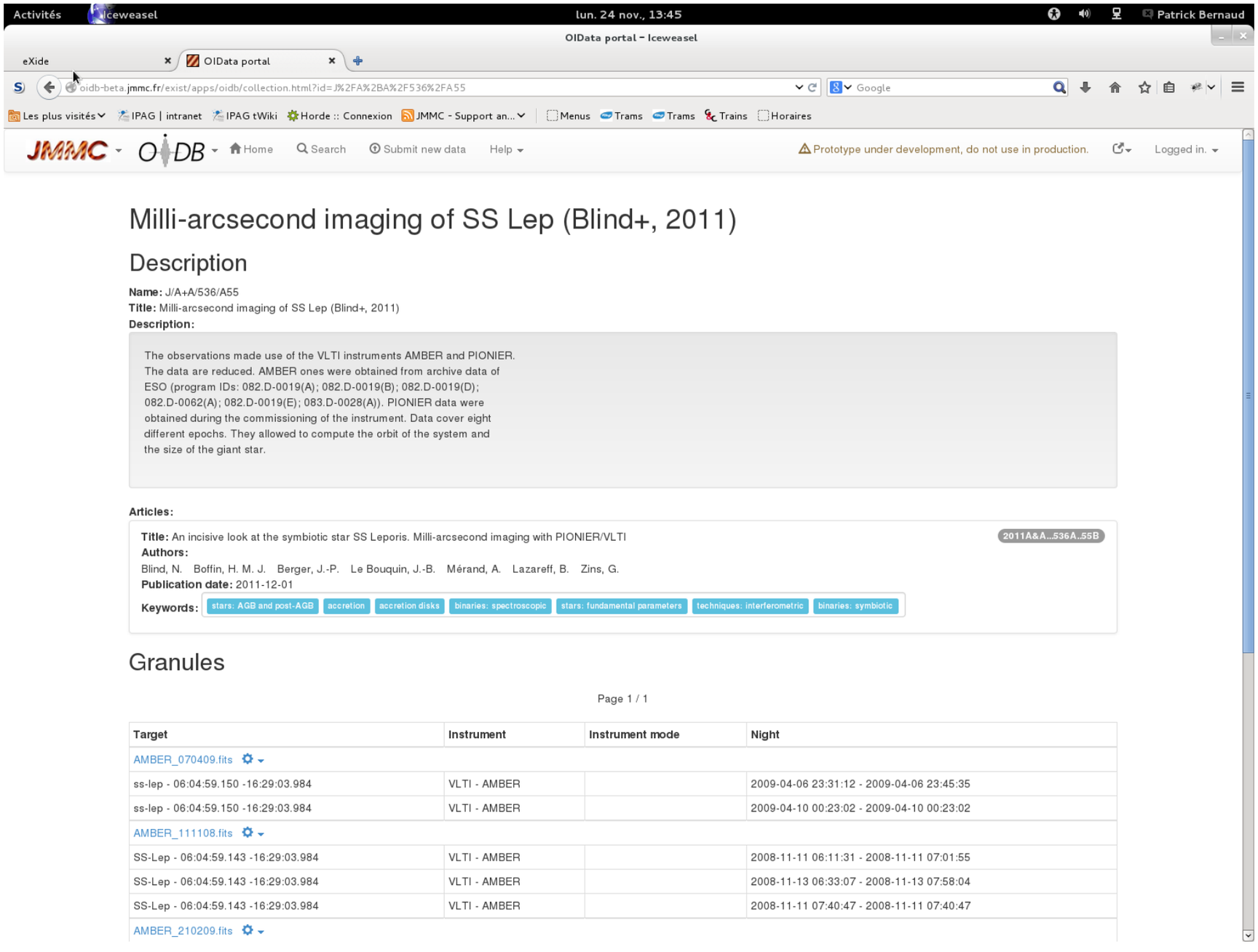
OiDb - Screenshots



OiDb - Screenshots (2)



OiDb - Screenshots (3)



Questions?