

<http://www.orthanc-server.com/static.php?page=web-viewer> Installation d'un plugin (Web Viewer)

Bouton WebViewer : OrthancExplorer.js (dans le code source, et dossier Ressources de WebViewer)

```
$('#series').live('pagebeforecreate', function() {  
    //$('#series-preview').parent().remove();  
  
    var b = $('<a>')  
        .attr('data-role', 'button')  
        .attr('href', '#')  
        .attr('data-icon', 'search')  
        .attr('data-theme', 'e')  
        .text('Orthanc4vet');  
  
    b.insertBefore($('#series-delete').parent().parent());  
    b.click(function() {  
        if ($.mobile.pageData) {  
            var series = $.mobile.pageData.uuid;  
            window.open('/web-viewer/app/viewer.html?series=' + series);  
        }  
    });  
});
```

On peut voir directement que le créateur a utilisé du jquery mobile afin de permettre l'utilisation et l'affichage de ce bouton.

On peut y voir la méthode live qui permet d'attacher un gestionnaire d'évènement sur un élément du dom : <http://api.jquery.com/live/>

On a aussi cherché un sélecteur : ici un div, “#series-delete” qui nous permet de choisir un emplacement, si il n'y a pas de div, on rajoute “document”.

Voici notre code créé :

```
//todo : utiliser la méthode on() de jquery mobile cf live() est deprecated  
$(document).live('pagebeforecreate', function() {  
  
    var d = $('<a>')  
        .attr('data-role', 'button')  
        .attr('href', '#')  
        .attr('data-icon', 'search')  
        .attr('data-theme', 'e')
```

```

        .text('Orthanc4vet');

d.insertBefore($('#all-patients'));
d.click(function() {
    if ($.mobile.pageData) {
        var series = $.mobile.pageData.uuid;
        window.open('/web-viewer/app/orthanc4vet.html');
    }
});
});

```

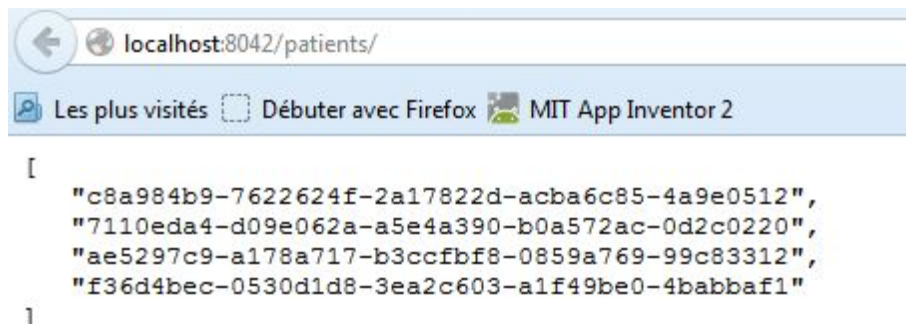
N'oubliez pas de re-build afin que le changement appliqué fonctionne, ainsi que de relancer Orthanc. (cf : <https://orthanc.googlecode.com/hg/LinuxCompilation.txt>)

Pas-sur : Ne pas oubliez d'effacer le cache avant le lancement d'Orthanc

Nous pensons qu'il faudra faire des triggers qui inséreront des données dans 4vet, puis ces données seront recopiées dans la base d'Orthanc afin qu'il n'y est pas de conflit.

Objectif :

Ce que je dois faire c'est interroger une des URLs, ici : localhost:8042/patients/ afin d'obtenir ceci :



Voici les tâches décrites par Mickaël, se trouvant dans le fichier 4vet.html :

Todo : dans cette page, en javascript (avec ajax & jquery mobile), interroger l'api rest de orthanc :

**
afficher la liste des patients**
</p>

<p>

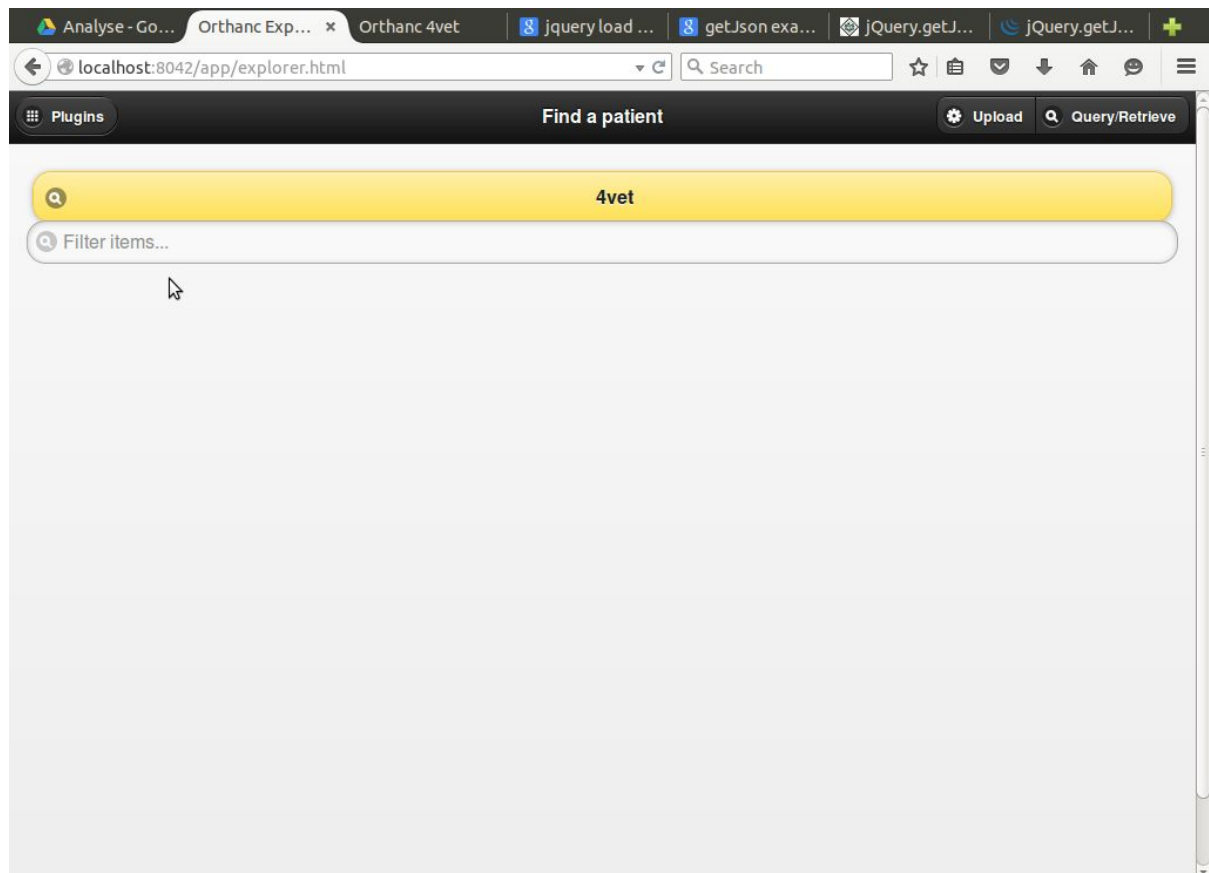
Ensuite, quand l'étape précédente marche :

**
**

coder en C++ pour exécuter une requête sur notre base de données orthanc4vet (c'est donc une 2eme bdd en plus de celle d'orthanc)

exposer le résultat de cette requête sous la forme d'une nouvelle url dans notre api rest (l'api rest do'rthanc4vet va compléter celle d'orthanc)

Voici ce que l'on obtient sur la page d'accueil d'Orthanc avec le plugin 4vet :



Puis en cliquant sur 4vet, vous devriez obtenir la page dans laquelle je cherche à afficher pour l'instant, l'idi des patients :



Bienvenue sur Orthanc 4vet

Todo : dans cette page, en javascript (avec ajax & jquery mobile), interroger l'api rest de orthanc :
afficher la liste des patients

Ensuite, quand l'étape précédente marche :

*Coder en C++ pour exécuter une requête sur notre base de données orthanc4vet (c'est donc une 2eme bdd en plus de celle d'orthanc)

*Exposer le résultat de cette requête sous la forme d'une nouvelle url dans notre api rest (l'api rest d'Orthanc 4vet va compléter celle d'orthanc)

J'essaie donc d'afficher l'id des patients en utilisant du JQuery, voici les deux fichiers importants se trouvant dans le dossier 4vet:

- 4vet.html
- 4vet.js

Samples aide code :

/* Un exemple intéressant :

<http://stackoverflow.com/questions/19491336/get-url-parameter-jquery>

```
document.URL == "http://helloworld.com/quotes?id=1337&author=kelvin&message=hello"
var currentURL = document.URL;
var params = currentURL.extract();
console.log(params.id); // 1337
console.log(params.author) // "kelvin"
console.log(params.message) // "hello"
```

<http://api.jquery.com/jquery.getJSON/> :Document API GetJSON

<http://www.lm-tech.it/Blog/post/2013/05/08/How-to-consume-a-RESTful-service-using-jQuery.aspx> : Autre lien intéressant.

<http://www.sitepoint.com/ajaxjquery-getjson-simple/> Exemple facile

<https://spring.io/guides/gs/consuming-rest-jquery/> : exemple ci-dessous

```
$(document).ready(function() {
    $.ajax({
        url: "localhost:8042/patients/"
    }).then(function(data) {
        alert(data);
    });
});
```

<http://www.tutorialspoint.com/jquery/ajax-jquery-getjson.htm> Ce dernier exemple me parait fonctionnelle.

```

$(document).ready(function() {
$("#bouton").click(function(event){
$.getJSON('http://localhost:8042/patients/', function(test) {
    $('#patient').html('<p> Patient: ' + test.patients + '</p>');
});
});
});

```

<http://stackoverflow.com/questions/9922101/get-json-data-from-external-url-and-display-it-in-a-div-as-plain-text> Lien intéressant avec exemple (Fonctionnelle)

*/

Après quelques recherches et essaie j'ai pu réussir, voici le code de 4vet.js :

```

// Test connexion
document.write("Test");

// Cette fonction permet l'affichage des patients qu'on pourra adapter pour l'affichage des
tags vétérinaires : http://www.tutorialspoint.com/jquery/ajax-jquery-getjson.htm
$(document).ready(function() {
$("#bouton").click(function(event){
$.getJSON('http://localhost:8042/patients/', function(test) {
    $('#patient').html('<p> Patient: ' + test + '</p>');
});
});
});

```

Voici le code de 4vet.html :

```

<!doctype html>

<html lang="us">
<head>

```

```

    <meta charset="utf-8" />
    <title>Orthanc 4vet</title>
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>
    <script src="4vet.js"></script>
</head>

<body>

    <h1> Bienvenue sur Orthanc 4vet </h1>
    <p>
        Todo : dans cette page, en javascript (avec ajax & jquery mobile), interroger l'api
rest de orthanc :
        <br/><b>afficher la liste des patients</b>
    </p>

    <p>
        Ensuite, quand l'étape précédnete marche :
    <br/>
        *Coder en C++ pour exécuter une requête sur notre base de données orthanc4vet
(c'est donc une 2eme bdd en plus de celle d'orthanc)
    <br/>
        *Exposer le résultat de cette requête sous la forme d'une nouvelle url dans notre api
rest (l'api rest d'Orthanc 4vet va compléter celle d'orthanc)

    </p>


    <!-- Permet l'apparition du bouton qui est utilisé par le script 4vet.js -->
    <h1>Patients:</h1>
    <p>Click on the button to load patients:</p>
    <div id="patient" style="background-color:#cc0;">
    </div>
    <input type="button" id="bouton" value="Load Data" />

</body>

```