

AJAX avec RIALTO et PHP

version 0.1

Par : [GHANEM Lhoussin](#)

Sommaire

INTRODUCTION.....	1
I- RIALTO.....	2
II- AJAX :.....	4
III- EXEMPLE :.....	6
CONCLUSION	16

INTRODUCTION

Ça fait longtemps que j'ai voulu apprendre AJAX pour me permettre de beaucoup m'amuser à créer des pages web qui restent chez leurs clients en se connectant de temps en temps à leur serveur, mais à chaque fois que je commence à suivre un tutoriel je me trouve forcé à l'abandonner pour des différentes raisons.

En cherchant à utiliser facilement AJAX je suis arrivé à un tutoriel qui me conseillait d'utiliser le framework prototype. plus tard je découvre RIALTO.

RIALTO pour Rich Internet Application Toolkit est un framework développé en Javascript pour gérer les applications destinées aux clients lourds.

RIALTO propose un ensemble des objets de bon look qu'on peut personnaliser.

Enfin pour savoir nos droits et nos devoirs Rialto est distribué sous licence Apache et est soutenu par Cyril Balit (Improve) et François Lion (IGR).

Visitez le site professionnel de RIALTO

<http://rialto.improve-technologies.com/wiki/rialto>

I- RIALTO

Dans cette partie on va découvrir RIALTO, comment fonctionne-t-il?

Pour commencer :

- Télécharger le framework RIALTO ici :
 - <http://rialto.improve-technologies.com/wiki/rialto/download/>
- Créer un dossier rialtoTuto dans la racine de votre serveur (/var/www/rialtoTuto) ou (/opt/lampp/htdocs/rialtoTuto) ou encore (c:\wamp\www\rialtoTuto\)
- décompresser le fichier zip téléchargé dans ce dossier
- Vous devez avoir le dossier comme suit :
 - chemin/vers/racine/rialtoTuto/rialtoEngine/
- Maintenant RIALTO est installé et vous pouvez commencer à l'utiliser.
- Revenez sur votre dossier: chemin/vers/racine/rialtoTuto/
- Créer un fichier index.html
- Ce fichier contiendra le code :

```

<html>
<head>
<!-- Ici on inclut les fichiers CSS -->
<link rel='STYLESHEET' type='text/css' href='./rialtoEngine/style/rialto.css'/>
<link id = 'standart_behavior' rel='STYLESHEET' type='text/css'
href='./rialtoEngine/style/behavior.css'>
<link rel='STYLESHEET' type='text/css'
href='./rialtoEngine/style/defaultSkin.css' >
<!-- Ici On inclut les fichiers javascript qui nous offrons les objets de RIALTO -->
<script type='text/javascript' src='./rialtoEngine/config.js'></script>
<script>rialtoConfig.pathRialtoE = "./rialtoEngine/";</script>
<script type='text/javascript' src='./rialtoEngine/javascript/rialto.js'></script>
</head>
<!-- Notre page html n'est rien d'autre que javascript
On annonce le début d'un code javascript -->
<script type='text/javascript'>
    /* Un objet SimpleWindow crée une fenêtre qui sera affiché par le
navigateur elle aura un titre, une fenête contenante */
    var ws=new rialto.widget.SimpleWindow({title:"Gestion des
etudiants",position:"absolute",name:"ws",parent:document.body,top:0,left:0,wi
dth:'99%',height:'99%'});
    /* Un objet Frame crée une fenêtre qui sera affiché par le navigateur elle
aura un titre et une fenête contenante qui est le SimpleWindow */
    frame= new rialto.widget.Frame({
    name:'fr',
    top:'20',
    left:'20',
    width:'99%',
    height:'99%',
    title:'Liste des etudiants',
    open:true,
    position:'absolute',
    parent:ws});

```

```
</script>
</body></html>
```

Lancez votre fichier pour découvrir le résultat (<http://localhost/rialtoTuto/>).

Maintenant que vous avez pris une idée sur le fonctionnement de RIALTO et avant de passer aux choses sérieuses, je vous recommande la documentation complète (bien qu'elle est en anglais) de RIALTO pour découvrir les autres objets offerts par ce framework. elle est en téléchargement ici : <http://rialto.improve-technologies.com/rialto/wiki/download/>

II- AJAX :

C'est sûr que ce tutoriel n'est pas destiné à l'explication de cette géante technologie mais on peut pas quand même passer sans en parler.

Cette technologie se base principalement sur un objet du langage JavaScript, XMLHttpRequest.

Cette objet permet de formuler des requêtes HTTP asynchrones, c'est à dire qu'elle seront exécutés après le chargement de la page web. L'utilisateur n'aura pas besoin de recharger la page web toute entière. Le plus bel exemple est le service de messagerie de Google.

Voici un très simple exemple que moi aussi j'ai rencontré sur le web:

Le fichier index.html

```
<html>
<body><script type="text/javascript">
//préparez l'objet XMLHttpRequest pour différents navigateurs :
function initAjax()
{
  var xmlhttp;
  try
  {
    // Firefox, Opera 8.0+, Safari
```

```

    xmlhttp=new XMLHttpRequest();
}
catch (e)
{
    // Internet Explorer
    try
    {
        xmlhttp=new ActiveXObject("Msxml2.XMLHTTP");
    }
    catch (e)
    {
        xmlhttp=new ctiveXObject("Microsoft.XMLHTTP");
    }
}
return xmlhttp ;
}
// La fonction qu'on appellera par un événement :
function ajax()
{
    xmlhttp=initAjax()
    xmlhttp.onreadystatechange=function()
    {
        if(xmlhttp.readyState==4)
        {
            //Ce qu'on va faire avec la réponse de la requête :
            //Ici on l'affecte à un champ de texte et il est touché tout seul
            document.myForm.time.value=xmlhttp.responseText;
        }
    }
    xmlhttp.open("GET","time.php",true);
    xmlhttp.send(null);
}
</script>

```

```
<form name="myForm">
Name: <input type="text" onkeyup="ajax();" name="username"
/>
Time: <input type="text" name="time" />
</form>
</body>
</html>
```

time.php est un fichier qui contient du texte simple

```
<?php
    echo "it is 04:00";
?>
```

Notez bien que cet exemple n'est pas suffisant pour parler de l'AJAX, il lui manque de l'XML, mais le principe reste le même.

III- EXEMPLE :

Un de mes prof le dit toujours: « Rien ne vaut qu'un exemple pour bien comprendre ».

Dans cette partie on va mettre en place une très petite application sur laquelle j'ai travaillé pendant deux jours, c'est ma première application utilisant RIALTO et AJAX et je souhaite la partager avec vous.

Dans cette application on aura besoin d'une base de donnée mysql contenant une table nommée "etudiant" :

```
CREATE DATABASE school;
USE school;
CREATE TABLE etudiant (
id char(20) primary key, nom char(50), prenom char(50), date date
);
```

Notre application va afficher la liste des étudiants dans un tableau (l'objet Grid offert par RIALTO), et va présenter un formulaire pour ajouter de nouveaux étudiants, et un bouton permettant la suppression de l'étudiant sélectionné sur le tableau.

Je suppose que vous êtes d'accord avec le cahier des charges, et je commence...

Vous avez votre base de données mysql, on va créer la page avec RIALTO tout d'abord. On aura besoin de php pour remplir le tableau avec les données de la table etudiant.

Je note que cet exemple n'a aucune relation avec les précédents.

voilà la page index.php

```
<html>
<head>
<title>Liste des auteurs</title>
<link rel='STYLESHEET' type='text/css' href='./rialtoEngine/style/rialto.css'/>
<link id = 'standart_behavior' rel='STYLESHEET' type='text/css'
href='./rialtoEngine/style/behavior.css'>
<link rel='STYLESHEET' type='text/css'
href='./rialtoEngine/style/defaultSkin.css'>
<script type='text/javascript' src='./rialtoEngine/config.js'></script>
<script>rialtoConfig.pathRialtoE = "./rialtoEngine/";</script>
<script type='text/javascript' src='./rialtoEngine/javascript/rialto.js'></script>
<script type='text/javascript' src='ajax.js'></script>
</head>
  <body>
    <script type='text/javascript'>
//On crée une fenêtre
var sWin = new
rialto.widget.SimpleWindow({parent:document.body,title:'Auteurs'});
//On y insère une frame
var frame = new rialto.widget.Frame({ name:'frame',title:'Ajout d'un etudiant',
top:'15',left:'10',width:'98%',height:'460',open:true,dynamic:true,position:'relati
ve',draggable:false,parent:sWin});
//On la divise horizontalement
var splitFC = new
rialto.widget.Splitter({top:0,left:0,height:'100%',width:'100%',prop:0.33,orienta
tion:'h',name:'splitFC',parent:frame,style:'3D',minProp:0,maxProp:1,withImg:tru
```

```

e,reverseClose:true});
//Sur la première division on insère les objets suivants :
//une étiquette (label) avec le texte id
new rialto.widget.Label('lib1',15,10,splitFC.div1,"id",'libelle1');
//Une zone de texte devant cet étiquette
var idEtudiant=new rialto.widget.Text('idEtudiant', 10,180,100,'A', splitFC.div1,
{rows:5} );
//Une étiquette et un champ de texte pour chaque champ dans la table :
new rialto.widget.Label('lib2',45,10,splitFC.div1,"Nom",'libelle1');
var nom=new rialto.widget.Text('nom',40,180,100,'A',splitFC.div1,{rows:5});
new rialto.widget.Label('lib4',75,10,splitFC.div1,"Prenom",'libelle1');
var prenom=new
rialto.widget.Text('prenom',70,180,100,'A',splitFC.div1,{rows:5});
new rialto.widget.Label('lib4',105,10,splitFC.div1,"Date de
Naissance",'libelle1');
var date=new rialto.widget.Text('date',100,180,100,'D',splitFC.div1,{rows:5});

//un bouton pour valider le formulaire une fois remplis
var btInsert=new rialto.widget.Button(130,200,"ajouter", "Ajouter l'etudiant",
splitFC.div1,{enable:true,adaptToText:true,width:88,widthMin:88});
//Un autre diviseur de de la deuxième division
var splitFC2 = new
rialto.widget.Splitter({top:0,left:0,height:'100%',width:'100%',prop:0.85,orienta
tion:'v',name:'splitFC2',parent:splitFC.div2,
style:'3D',minProp:0,maxProp:1,withImg:true,reverseClose:true});
//On a le droit de faire puisque c'est facile et en plus gratuit
//Une deuxième frame dans la division obtenue :
var frame2 = new rialto.widget.Frame({name:'frame2',title:'Manipulation des
donnees', top:'15',left:'10',width:'98%',height:'98%', open:true,dynamic:true,
position:'relative',draggable:false,parent:splitFC2.div2});
//On y insère un bouton permettant la suppression :
var btSupp=new rialto.widget.Button(1,1,"Supprimer","Supprimer

```

```
l'etudiant",frame2,{enable:true,adaptToText:true,width:88,widthMin:88});
```

```
//Ici c'est du php classique adaptez le script à vos paramètres:
```

```
<?php
```

```
    mysql_connect("localhost","root","");
    mysql_select_db("school");
    $res = mysql_query("select * from etudiant");
    $i = 0;
    while($tab=mysql_fetch_array($res))
    {
        $ids[$i] = $tab[0];
        $noms[$i] = $tab[1];
        $prenoms[$i] = $tab[2];
        $dates[$i] = $tab[3];
        $i++;
    }
```

```
?>
```

```
//Enfin voilà comment on déclare et dessine un tableau :
```

```
oTableau = new rialto.widget.Grid
({autoResizableW:true,autoResizableH:false,withContextualMenu:true,asynchLoad:true,top:0,left:0,height:'99%',name:'oTableau',parent:splitFC2.div1,
navigationMode:true,row:15,cellActive:false,sortable:true,multiSelect:false,line
Height:16,boolPrint:true,switchable:true,clickable:true,writable:true,
arrHeader:[{title:"Id",width:50,type:"string"},{title:"Nom",width:150,type:"string"},
{title:"Prenom",width:150,type:"string"},
{title:"Date de
naissance",width:150,type:"date",pattern:"dd/MM/yyyy",maskPattern:"dd/MM/yyyy"}]});
```

```
//Et voilà comment le remplir avec des données de la bd :
```

```

oTableau.fillGrid([
    <?php
        $n = $i;
        for($i=0;$i<$n;$i++){
    ?>
    ["<? echo $ids[$i]; ?>","<?php echo $noms[$i]; ?>","<?php
echo $prenoms[$i]; ?>","<?php echo $dates[$i]; ?>"],
    <?php } ?>
]);

//La pour chaque bouton on appelle une fonction javascript
// Les deux fonctions inserer() et supprimer vont être déclarées dans le fichier
ajax.js
btInsert.onclick = function() { inserer(); }
btSupp.onclick = function() { supprimer(); }
</script>
</body>
</html>
<!-- fin du fichier index.html -->

```

Voyons maintenant comment se présentent les deux fonctions (supprimer et inserer) et dans quels conditions ils vivent, voici le fichier ajax.js :

```

//On récupère l'objet XMLHttpRequest :
function getXhr(){
    var xhr = null;
    if(window.XMLHttpRequest) // Firefox et autres
        xhr = new XMLHttpRequest();
    else if(window.ActiveXObject){ // Internet Explorer
        try {
            xhr = new ActiveXObject("Msxml2.XMLHTTP");

```

```

    }
    catch (e) {
        xhr = new ActiveXObject("Microsoft.XMLHTTP");
    }
}
else { // XMLHttpRequest non supporté par le navigateur
    alert("Votre navigateur ne supporte pas les objets XMLHttpRequest");
    xhr = false;
}
return xhr;
}
//Les deux fontions suivantes vont nous permettre de récupérer un même
fichier xml quelque soit le navigateur
function go(c){
    if(!c.data.replace(/\s/g,""))
        c.parentNode.removeChild(c);
}
function clean(d){
    var bal=d.getElementsByTagName('*');
    for(i=0;i<bal.length;i++){
        a=bal[i].previousSibling;
        if(a && a.nodeType==3)
            go(a);
        b=bal[i].nextSibling;
        if(b && b.nodeType==3)
            go(b);
    }
    return d;
}
//Et ici la fontion qui va permettre l'insertion :
function inserer(){
    var xhr = getXhr();
    // On défini ce qu'on va faire quand on aura la réponse

```

```

xhr.onreadystatechange = function(){
// On ne fait quelque chose que si on a tout reçu et que le serveur est ok
if(xhr.readyState == 4 && xhr.status == 200){
//on récupère la réponse en format XML:
response = clean(xhr.responseXML.documentElement);
var items = response.getElementsByTagName("element");
/* Nombre des elements */
count = items.length;
/* Présentation des données sous forme d'une tableau js */
donnees = new Array();
for(i = 0; i < count; i++)
{
    var identifiant =
items[i].getElementsByTagName("id")[0].firstChild.nodeValue;
    var nom =
items[i].getElementsByTagName("nom")[0].firstChild.nodeValue ;
    var prenom =
items[i].getElementsByTagName("prenom")[0].firstChild.nodeValue;
    var date =
items[i].getElementsByTagName("date")[0].firstChild.nodeValue;
    donnees[i] = new Array(identifiant,nom,prenom,date);
}
//On réinitialise note tableau dessiné
oTableau.initTab();
//On le remplit avec les nouvelles données
oTableau.fillGrid(donnees);
//On y sélectionne la première ligne :
oTableau.clickFirst();
}
}
// Ici on envoie des données en post
xhr.open("POST","data.php",true);
// ne pas oublier ça pour le post

```

```

    xhr.setRequestHeader('Content-Type','application/x-www-form-
urlencoded');
    // ne pas oublier de poster les arguments
    // ici, l'id de l'etudiant
    idEt = idEtudiant.getValue(); // id du nouveau étudiant
    nomEt = nom.getValue(); // nom du nouveau étudiant
    preEt = prenom.getValue(); // prénom du nouveau étudiant
    dtEt = date.getValue(); //Date de naissance du nouveau étudiant
    //On envoie la requête au serveur
    xhr.send("id="+idEt+"&nom="+nomEt+"&prenom="+preEt+"&date="+
dtEt);
}
//Pour la fonction de suppression j'ai préféré de tout reprendre comme pour la
fontion d'insertion même si j'étais sûr qu'il y aura une solution plus fiable
function supprimer(){
    var xhr = getXhr();
// On défini ce qu'on va faire quand on aura la réponse
    xhr.onreadystatechange = function(){
// On ne fait quelque chose que si on a tout reçu et que le serveur est ok
    if(xhr.readyState == 4 && xhr.status == 200){
//on récupère la réponse en format XML:
    response = clean(xhr.responseXML.documentElement);
    var items = response.getElementsByTagName("element");
/* Nombre des elements */
    count = items.length;
/* Présentation des données sous forme d'une tableau js */
    donnees = new Array();
    for(i = 0; i < count; i++)
    {
        var identifiant =
items[i].getElementsByTagName("id")[0].firstChild.nodeValue;
        var nom =
items[i].getElementsByTagName("nom")[0].firstChild.nodeValue ;

```

```

        var prenom =
items[i].getElementsByTagName("prenom")[0].firstChild.nodeValue;
        var date =
items[i].getElementsByTagName("date")[0].firstChild.nodeValue;
        donnees[i] = new Array(identifiant,nom,prenom,date);
    }
    //On réinitialise note tableau dessiné
    oTableau.initTab();
    //On le remplit avec les nouvelles données
    oTableau.fillGrid(donnees);
    //On y sélectionne la première ligne :
    oTableau.clickFirst();
    }
}
// Ici on envoie des données en post
xhr.open("POST","data.php",true);
// ne pas oublier ça pour le post
xhr.setRequestHeader('Content-Type','application/x-www-form-
urlencoded');
// ne pas oublier de poster les arguments
// ici, l'id de l'étudiant
//C'est où commence la différence avec la fonction précédente
//On récupère la cellule du tableau sur la on affiche l'identifiant de
l'étudiant :
    ident = oTableau.getCellText(oTableau.getIndLineClic(),0);
    // On utilise l'objet string offert par RIALTO pour éviter les espaces de
début et de fin de la chaîne.
    idEt = rialto.string.trim(ident);
    // On se sert de Notre objet XHR pour envoyer la requête au serveur
    xhr.send("idEt="+idEt);
}
//fin du fichier ajax.js

```

En commentant ce dernier fichier on a parlé des données sous formats XML mais où sont ils ces données? c'est la réponse du serveur.

Un fichier xml est généré dynamiquement par php de façon à ce qu'il représente les mêmes données que celles qu'on a dans la base des données.

Voilà le fichier php/XML nommé **data.php** :

```
<?php
//Définition de l'entête
header('Content-Type: text/xml');
echo "<?xml version=\"1.0\"?>\n";
//On se connecte à la BD
mysql_connect("localhost","root","");
mysql_select_db("school");

//On insère les données si le bouton d'insertion a été cliqué
if(isset($_POST['id']))
{
    $id = $_POST['id'];
    $nom = $_POST['nom'];
    $prenom = $_POST['prenom'];
    $date = $_POST['date'];
    $req="insert into etudiant values($id,'$nom','$prenom','$date)";
    mysql_query($req)or die("Un problème a eu place pendant
l'insertion de l'etudiant");
}
//On supprime l'étudiant concerné si le bouton de suppression a été
liqué
if(isset($_POST['idEt']))
{
    $id = $_POST['idEt'];
    $req="delete from etudiant where id='$id'";
    mysql_query($req)or die("Un problème a eu place pendant la
suppression de l'etudiant");
}
```

```

//Dans le reste on cherche à créer un fichier XML contenant nos données
echo "<etudiant>";
$res = mysql_query("select * from etudiant");
while($tab=mysql_fetch_array($res))
{
    $id = $tab[0];
    $nom = $tab[1];
    $prenom = $tab[2];
    $date = $tab[3];
    echo "\t<element>\n";
    echo "\t\t<id> ".$id." </id>\n";
    echo "\t\t<nom> ".$nom." </nom>\n";
    echo "\t\t<prenom> ".$prenom." </prenom>\n";
    echo "\t\t<date> ".$date." </date>\n";
    echo "\t</element>\n";
}
echo "</etudiant>";
?>
//Fin du fichier data.php

```

CONCLUSION

RIALTO nous offre un objet à utiliser pour faire de l'AJAX mais je n'ai pas pu deviné comment il marche et sa documentation ne m'a pas suffit pour comprendre son fonctionnement, c'est **ajaxRequest**, à vous de le découvrir.

Tout ce que je souhaite maintenant c'est que ce tutoriel va aider quelqu'un. faites un feedback dans le cas où vous appréciez ce petit travail, qui est fait proprement pour vous et vous en avez tout les droits offerts par la GPL.

lhoussin2005@gmail.com