

AJAX

jQuery is a fast, small, and feature-rich JavaScript library.

« write less, do more »

AJAX permet d'émettre des requêtes asynchrones et de ne modifier qu'une partie du DOM.

Références :

- <http://api.jquery.com/category/ajax/>
- jquery_ajax.odt

\$.get()

`$.get()` est une version simplifiée de `$.ajax()`.

```
var jqXHR = $.get(  
    "ressource"  
    [, paramètres  
    , [type] ] ]  
);  
jqXHR.done(function(data [,statut, xhr]) {  
    // code  
});  
jqXHR.fail(function(xhr, statut, erreur) {  
    // code  
});
```

\$.get()

Les paramètres de `done()` sont les données, le statut de la requête, un objet XHR.

Objet/String

Description

`data`

Les données reçues

`statut`

`"success".`

`xhr`

`xhr.status : 200, 404, ...`

`xhr.statusText : OK, ...`

Les paramètres de `fail(xhr, statut, erreur)`.

Objet/String

Description

`xhr`

`xhr.status : 200, 404, ...`

`xhr.statusText : Not Found, ...`

`statut`

`"timeout", "error", "abort", and "parsererror".`

`erreur`

`HTTP Status (Not Found, Internal Server Error, ...)`

\$.get()

```
var jqXHR = $.get(
    "ressource"
    [, paramètres
    , [type] ] ]
);

jqXHR.done(function(data [,statut, xhr]) {
    // code
});

jqXHR.fail(function(xhr, statut, erreur) {
    // code
});
```

\$.post()

```
var jqXHR = $.post(  
  "ressource"  
  [, { attribut1:valeur1 [, attribut2:valeur2] }  
  , [type] ]  
  );
```

```
jqXHR.done(function(data, statut, xhr) {  
  // code  
});
```

```
jqXHR.fail(function(xhr, statut, erreur) {  
  // code  
});
```

\$.post()

```
function insert() {
    var lsCp = $("#itCP").val();
    var lsNomVille = $("#itNomVille").val();
    var lsIdPays = $("#itIdPays").val();

    var jqXHR = $.post(
        "../php/VillesInsert.php",
        {cp: lsCp, nom_ville: lsNomVille, id_pays: lsIdPays},
        "text"
    );

    jqXHR.done(function(data) {
        $("#lblMessage").html(data);
    });

    jqXHR.fail(function(xhr, statut, erreur) {
        var sTexte = xhr.status + ":" + xhr.statusText;
        $("#lblMessage").html(sTexte);
    });
} /// insert
```

\$.getJSON()

```
function init() {  
  
    var jqXHR = $.getJSON("../ressources/json/villes.json");  
  
    jqXHR.done(function(data) {  
        var lsResultat = "";  
  
        // Recuperation des elements un par un avec each()  
        $.each(data, function(cle, valeur) {  
            // cle est l'indice dans le tableau  
            // valeur est un objet JSON  
            lsResultat += "[" + cle + "] " + valeur.codePostal + " : " + valeur.Ville  
+ ", " + valeur["habitants"] + " habitants<br/>";  
        });  
  
        // Recuperation des elements un par un avec un for-)  
        for (var i = 0; i < data.length; i++) {  
            lsResultat += data[i].ville;  
            lsResultat += " : ";  
            lsResultat += data[i].codePostal;  
            lsResultat += "<br>";  
        }  
        $("#pResultat").html(lsResultat);  
    });  
} /// init
```

Compléments

- Pour jsonp cf jquery_ajax.odt (section 1.8)
- Pour \$.getScript() cf jquery_ajax.odt (section 1.7)
- Pour le cross domain cf jquery_ajax.odt (section 1.10)