Cette méthode permet d’effectuer une requête AJAX en maîtrisant, grâce aux nombreuses options disponibles, les différents paramètres et étapes de celle-ci.

ajax(options)

Réalise une requête HTTP asynchrone (AJAX).

$.ajax({

url: "test.htm",

success: function(data ) {

$("#resultat").html(data);

$().log("Terminé");

},

});

La méthode renvoie un objet XMLHttpRequest.

Passons en revue les nombreuses options disponibles.

* url (obligatoire): une chaîne de caractères contenant l’adresse de la requête.
* type (optionnel) : une chaîne de caractères qui définit la méthode HTTP à utiliser pour la requête (GET ou POST). La valeur par défaut est GET. D’autres méthodes d’envoi HTTP peuvent être utilisées, comme PUT ou DELETE, mais celles-ci ne sont pas supportées par tous les navigateurs.
* dataType (optionnel) : une chaîne de caractères qui spécifie le format des données qui seront renvoyées par le serveur (xml, html, json ou script). Si rien n’est spécifié, jQuery utilisera le type MIME pour déterminer le format adéquat soit responseXML ou ResponseText. Les types disponibles sont :
  + "xml" : retourne un document XML qui pourra être traité par jQuery.
  + "html" : retourne du code Html au format texte.
  + "script" : évalue la réponse en JavaScript et retourne cette dernière au format texte.
  + "json" : évalue la réponse en JSON et retourne un objet JavaScript.
* ifModified (optionnel) : une valeur booléenne qui indique que le serveur doit vérifier si les données retournées sont différentes de la dernière requête avant de renvoyer le fichier avec succès. Par défaut, cette option vaut false.
* timeout (optionnel) : nombre de millisecondes après lequel la requête est considérée comme non réussie.
* global (optionnel) : une valeur booléenne qui permet le déclenchement du gestionnaire d’évènements global d’AJAX. Par défaut, la valeur est true. Avec une valeur false, les déclenchements d’évènements de type ajaxStart() ou ajaxStop() sont ignorés.
* beforeSend (optionnel) : une fonction qui doit être exécutée avant l’envoi de la requête. Ceci permet de modifier l’objet XMLHttpRequest avant qu’il soit envoyé pour spécifier, par exemple, des en-têtes HTTP personnalisées.
* error (optionnel) : une fonction qui doit être appelée en cas d’échec de la requête. La fonction dispose de trois arguments : l’objet XMLHttpRequest, une chaîne de caractères décrivant le type d’erreur rencontré et un objet d’exception, dans le cas où ce dernier a été généré.
* success (optionnel) : fonction à appeler si la requête s’exécute avec succès. Un seul argument est passé en paramètre soit les données retournées par le serveur.
* complete (optionnel) : fonction à exécuter lorsque la requête se termine. La fonction dispose de deux arguments : l’objet XMLHttpRequest et une chaîne de caractères décrivant le type de succès de la requête.
* data (optionnel) : données à envoyer au serveur. L’objet doit être formé de paires de la forme clé/valeur. Les données sont converties en chaîne de caractères (si elles ne le sont pas déjà). Voir l’option processData ci-après pour empêcher ce processus automatique.
* processData (optionnel) : valeur booléenne qui indique si les données de l’option data doivent être converties en chaîne de caractères. La valeur par défaut est true. Pour empêcher la conversion, passez cette option à false.
* contentType (optionnel) : chaîne de caractères contenant le MIME des données lorsque des données sont envoyées au serveur. Par défaut, le MIME application/x-www-form-urlencoded est retenu.
* async (optionnel) : une valeur booléenne qui indique si la requête doit s’effectuer de façon asynchrone ou synchrone. La valeur par défaut pour une requête AJAX est bien entendu true.

D’autres options sont encore disponibles mais d’un usage moins fréquent. Énumérons simplement :

* cache (optionnel) : une valeur booléenne qui, lorsqu’elle est mise sur false, empêche la page chargée d’être mise dans le cache du navigateur.
* password (optionnel) : dans le cas où l’accès HTTP de la requête nécessite un mot de passe.
* username (optionnel) : dans le cas où l’accès HTTP de la requête nécessite un nom d’utilisateur (username).
* scriptCharset (optionnel) : force les requêtes du type script à être interprétées avec un charset particulier.
* xhr (optionnel) permet de créer l’ActiveXObject (Internet Explorer) ou le XMLHttpRequest (pour les autres).