

# Quelques méthodes détaillées

## 1. Méthode getAllResponseHeaders

Pour ce qui suit, il faut bien distinguer un en-tête HTTP d'un en-tête de fichier HTML. Les balises comprises dans un en-tête HTML, soit entre les balises `<head> . . . </head>`, envoient des informations concernant la page HTML, tandis que les en-têtes HTTP envoient des informations relatives au fichier.

L'en-tête HTTP est un court message que le serveur web envoie au navigateur juste avant de lui transmettre le document lui-même. Ce message sert par exemple à donner la taille du document ou à indiquer qu'il a disparu (code 404). Il peut aussi servir à préciser l'encodage du fichier grâce à la ligne `Content-Type`.

Il faut aussi noter qu'avec une requête HTTP, le serveur renvoie les en-têtes de la ressource ou du fichier, pas le fichier lui-même. Cela signifie que des renseignements sur un document, comme `Content-type` ou `Last-modified`, sont connus sans avoir à télécharger le document lui-même.

La méthode `getAllResponseHeaders()` renvoie tous les en-têtes HTTP de la réponse sous la forme d'une chaîne de caractères.

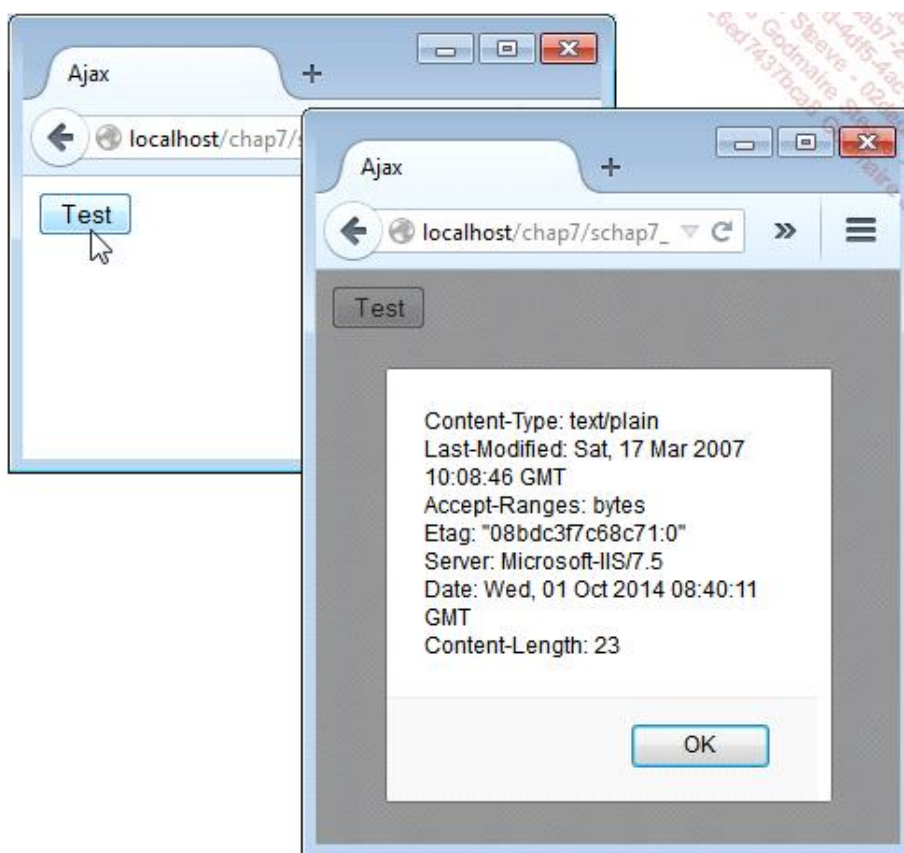
### Exemple

Au clic sur un bouton, affichons, dans une boîte d'alerte, tous les en-têtes du fichier `test.txt`.

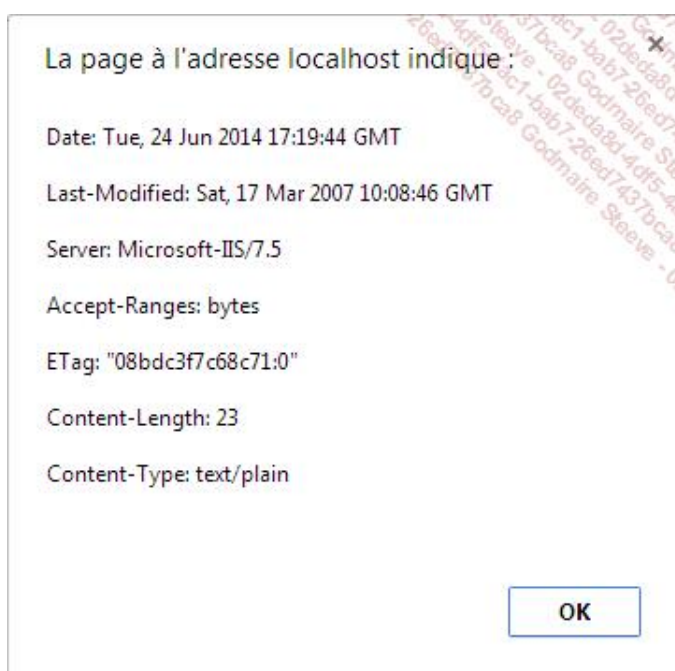
```
<!DOCTYPE html>
<html lang="fr">
<head>
<title>XMLHttpRequest</title>
<meta charset="UTF-8">
<script>
function getxhr(){
var xhr = null;
if(window.XMLHttpRequest){
var xhr = new XMLHttpRequest();
}
else if(window.ActiveXObject){
var xhr = new ActiveXObject("Microsoft.XMLHTTP");
}
else {
alert("Votre navigateur n'est pas compatible avec AJAX...");
}
xhr.onreadystatechange = function(){
if(xhr.readyState == 4 && xhr.status == 200) {
alert(xhr.getAllResponseHeaders());
}
}
xhr.open("GET","test.txt",true);
xhr.send(null);
}
</script>
</head>
<body>
<form>
<input type="button" value="Test" onclick="getxhr()">
```

```
</form>
</body>
</html>
```

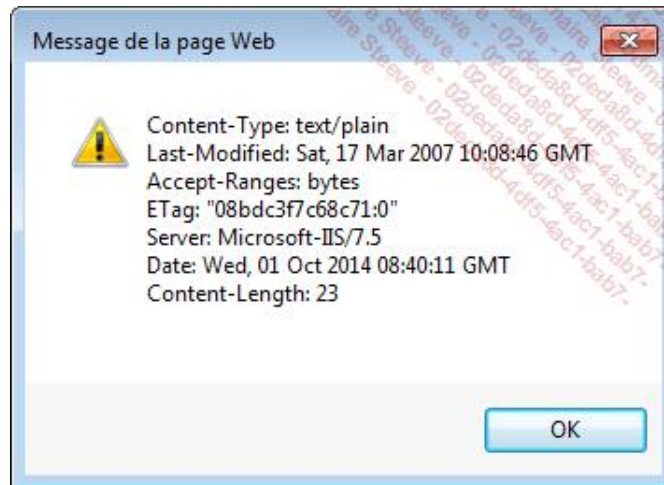
Sous Firefox :



Sous Google Chrome :



Sous Internet Explorer :



## 2. Méthode `getResponseHeader`

La méthode `getResponseHeader()` renvoie la valeur d'un seul champ de l'en-tête HTTP sous forme d'une chaîne de caractères.

Elle prend la forme de `getResponseHeader("nom du paramètre d'en-tête HTTP")`.

Soit `xhr.getResponseHeader("Content-Type")`, qui ne retourne que les renseignements concernant le type de contenu (*Content-Type*).

### Exemple

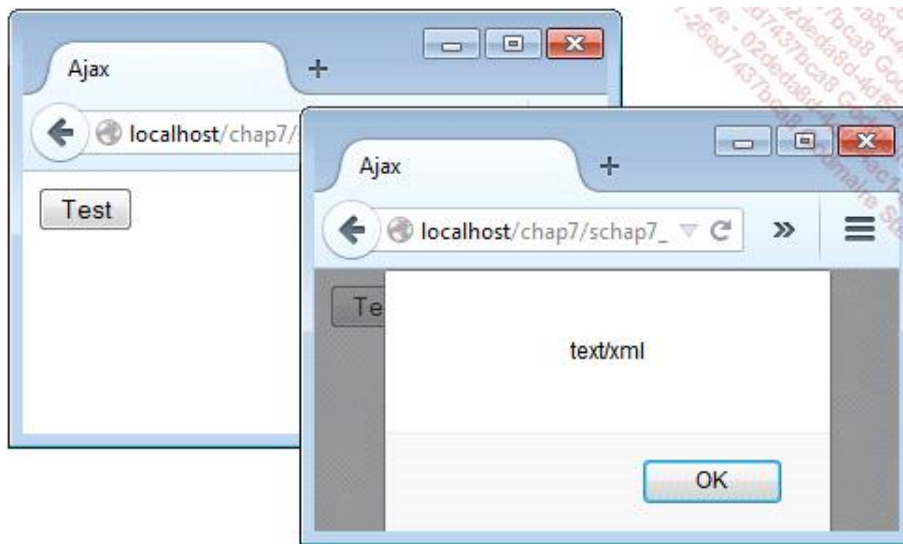
Au clic sur un bouton, affichons, dans une boîte d'alerte, l'en-tête de contenu de fichier (*Content-Type*) du fichier `test.xml` :

```
<!DOCTYPE html>
<html lang="fr">
<head>
<title>XMLHttpRequest</title>
<meta charset="UTF-8">
<script>
function getxhr(){
var xhr = null;
if(window.XMLHttpRequest){
var xhr = new XMLHttpRequest();
}
else if(window.ActiveXObject){
var xhr = new ActiveXObject("Microsoft.XMLHTTP");
}
else {
alert("Votre navigateur n'est pas compatible avec AJAX...");
}
xhr.onreadystatechange = function(){
if(xhr.readyState == 4 && xhr.status == 200) {
```

```

alert(xhr.getResponseHeader("Content-type"));
}
}
xhr.open("GET","test.xml",true);
xhr.send(null);
}
</script>
</head>
<body>
<form>
<input type="button" value="Test" onclick="getxhr()">
</form>
</body>
</html>

```



### 3. Méthode `setRequestHeader`

Cette méthode assigne une valeur à un champ d'en-tête HTTP qui est envoyée lors de la requête.

Elle prend la forme de `setRequestHeader("nom du paramètre d'en-tête", "valeur associée au paramètre")`.

Elle est spécialement utilisée lors d'un envoi avec la méthode POST pour spécifier l'encodage :

```
xhr.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
```

### 4. Méthode `overrideMimeType`

Cette méthode force un document à être traité selon un type de contenu particulier (*MIME Type*). On appelle souvent cette méthode lors de l'utilisation de `responseXML` sur un serveur transmettant du XML avec un en-tête *Content-Type* incorrect.

```
xhr.req.overrideMimeType('text/xml');
```