

Chiamate ajax con jQuery

[Pagina principale di Javascript](#)

Cos'è AJAX

AJAX sta per Asynchronous JavaScript and XML. AJAX è una tecnica di sviluppo web utilizzata per creare applicazioni web interattive. Consente di inviare e ricevere dati dal server senza dover ricaricare completamente la pagina web. Invece di caricare una pagina intera ogni volta che si desidera visualizzare nuovi dati, AJAX consente agli sviluppatori di caricare solo le porzioni di pagina che sono necessarie per aggiornare il contenuto desiderato.

Le applicazioni web che utilizzano AJAX possono inviare richieste HTTP asincrone al server web e ricevere risposte senza interrompere o ricaricare la pagina. Questo può migliorare l'esperienza utente rendendo le applicazioni web più veloci e fluide, poiché gli utenti non devono attendere il ricaricamento completo della pagina ogni volta che interagiscono con essa.

Nonostante il nome, AJAX non è limitato all'utilizzo di XML come formato per scambiare dati con il server. Spesso, i dati vengono scambiati in formati più leggeri e flessibili come JSON (JavaScript Object Notation).

Per quanto ci riguarda utilizzeremo solamentene jQuery per grstire chiamate e risposte fatte con AJAX.

Si usa jQuery è nescessario includerlo usando il tag script:

includere jQuery

```
<script src="https://code.jquery.com/jquery-3.7.1.min.js"></script>
```

Il metodo get

Il metodo **get()** di jQuery viene utilizzato per effettuare una chiamata ajax con il metodo get

La sua sintassi è:

metodo get

```
$.get(stringa_url,parametri,function(data){  
    //nel parametro data ci sono il risultato dell'esecuzione  
    //del file chiamato con i parametri passati  
})
```

- **stringa_url**: è una stringa che rappresenta la pagina che riceverà i dati
- **parametri**: sono i parametri che vengono passati, con il metodo get, alla pagina in stringa_url, servono a produrre un risultato. I parametri possono avere forma di stringa o avere la codifica json

- **function(data)**: è la funzione di callback, server a gestire il risultato della pagina chiamata, che è contenuto nella variabile locale **data**. Il contenuto di data è in formato HTML e spesso si utilizza per modificare la pagina chiamante.

esempio metodo get

```
//chiamiamo la pagina pag.php, passando il parametro num=7
$.get('ajaxResp.php','num=7',function(data){
    $('#res').html(data)
})
```

E' anche possibile passare più parametri:

esempio metodo get con più parametri

```
//chiamiamo la pagina pag.php, passando il parametro num=7
$.get('ajaxResp.php','num=7&num2=4&num3=7',function(data){
    $('#res').html(data)
})
```

I parametri possono essere passati anche usando JSON (è un formato di scambio dati):

metodo get con JSON

```
//chiamiamo la pagina pag.php, passando il parametro num=7
$.get('ajaxResp.php',{num:7,num2:2,num3:7},function(data){
    $('#res').html(data)
})
```

Il codice JSON richiede che i dati vengano passati in coppie chiave:valore, se si deve passare più coppie vanno passate da virgola. Il valore posto a sinistra è quello che bisogna usare nel metodo JSON, anche se sembra più complesso, semplifica l'invio dei dati quando alla sinistra dei : c'è una variabile.

Metodo post

Il metodo **post()** è del tutto analogo al get, l'unica cosa diversa è che i parametri verranno passati utilizzando il metodo post. La sintassi jQuery è praticamente la stessa.

metodo post

```
$.post(stringa_url,parametri,function(data){
    //nel parametro data ci sono il risultato dell'esecuzione
    //del file chiamato con i parametri passati
})
```

- **stringa_url**: è una stringa che rappresenta la pagina che riceverà i dati
- **parametri**: sono i parametri che vengono passati, con il metodo get, alla pagina in stringa_url, servono a produrre un risultato. I parametri possono avere forma di stringa o avere la codifica json

- **function(data)**: è la funzione di callback, server a gestire il risultato della pagina chiamata, che è contenuto nella variabile locale **data**. Il contenuto di data è in formato HTML e spesso si utilizza per modificare la pagina chiamante.

esempio metodo post

```
//chiamiamo la pagina pag.php, passando il parametro num=7
$.post('ajaxResp.php', 'num=7&num2=4&num3=7', function(data){
    $('#res').html(data)
})
```

I parametri possono essere passati anche usando JSON:

metodo post con JSON

```
//chiamiamo la pagina pag.php, passando il parametro num=7
$.post('ajaxResp.php', {num:7,num2:2,num3:7}, function(data){
    $('#res').html(data)
})
```

Il metodo serialize()

Il metodo `.serialize()` è utilizzato per serializzare i valori di un modulo HTML in una stringa di query string. Questo è utile quando si desidera inviare i dati di un modulo tramite AJAX o quando si desidera semplicemente ottenere una rappresentazione dei dati del modulo per ulteriori elaborazioni o debugging.

Ecco un esempio di come utilizzarlo:

Supponiamo di avere un semplice modulo HTML come questo:

from HTML

```
<form id="myForm">
  <input type="text" name="username" value="john">
  <input type="password" name="password" value="password123">
  <input type="checkbox" name="rememberMe" checked>
  <input type="submit" value="Submit">
</form>
```

Quando il modulo viene inviato, possiamo utilizzare jQuery per serializzare i suoi valori:

Codice jQuery

```
$(document).ready(function() {
    $('#myForm').submit(function(event) {
        event.preventDefault(); // Evita il comportamento predefinito di inviare
        // il modulo
        var formData = $(this).serialize();
        console.log(formData); // Stampa la stringa serializzata nel console
    });
});
```

```
});  
});
```

In questo esempio, `$(this).serialize()` viene utilizzato per serializzare i dati del modulo quando viene inviato.

La stringa serializzata rappresenterà tutti i valori dei campi del modulo nel formato `chiave=valore&chiave=valore`.

From:

<https://www.wiki.webtocom.com/> - **WTC wiki**

Permanent link:

https://www.wiki.webtocom.com/doku.php?id=page:chiamate_ajax_jquery

Last update: **2024/04/09 10:40**

