

# Gestione di google maps tramite Ajax

Luca Meoli

5° BII

23/03/2018

## **TABELLA DELLE REVISIONI**

<i><b>Data</b></i>	<i><b>Versione</b></i>	<i><b>Modifiche</b></i>
14/02/18	Versione n°1	Creazione della documentazione
23/03/18	Versione n°2	Correzione ortografica, correzione del template, aggiunta dei passaggi di lavoro

# **INDICE**

## *1. Documentazione*

### *1.1. Ajax*

### *1.2. Javascript*

### *1.3. Passaggi del codice*

#### *1.3.1. Creazione della mappa*

#### *1.3.2. Acquisizione dei dati dall'XML con Ajax*

#### *1.3.3. Marker*

### *1.4. Google Maps API*

### *1.5. Fonti*

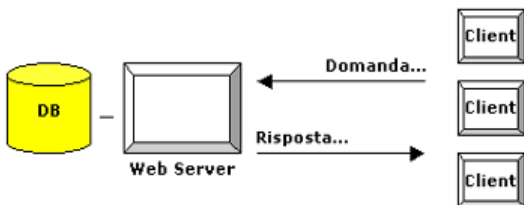
## **1. Documentazione**

### **1.1. Ajax**

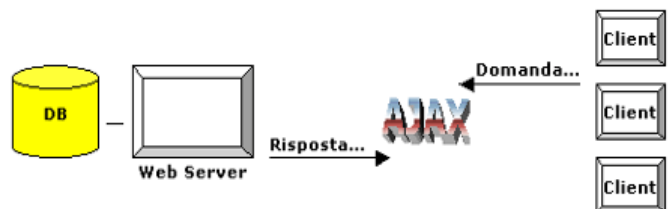
*Ajax è sistema di sviluppo di applicazioni Web lato server che usufruisce delle caratteristiche di Javascript lato client ed XML per rendere più veloci i processi di richiesta e l'esposizione dei dati. Ajax è uno standard, per cui si tratta di un sistema compatibile con tutti i browser Web (cross-browser) in cui l'oggetto XMLHttpRequest può assumere non solo comportamenti differenti ma in alcuni casi anche nomi differenti. Il funzionamento di Ajax è molto semplice ed è simile all'invio di dati attraverso un modulo HTML tramite i metodi GET e/o POST. La sua differenza sta nel fatto che i dati sono già caricati in memoria in elementi come array Javascript o nell'XML, fisico o*

*generato “al volo”, per cui i tempi di attesa sono praticamente inesistenti.*

Invio e restituzione dei dati remoti tramite HTTP



Invio e restituzione dei dati remoti tramite Ajax



## **1.2 Javascript**

*Javascript è un linguaggio di scripting orientato agli oggetti e agli eventi. Il suo utilizzo viene richiesto nella programmazione Web lato client per la creazione, in siti web e applicazioni web, di effetti dinamici interattivi tramite funzioni di script invocate da eventi innescati a loro volta in vari modi dall'utente sulla pagina web in uso. In questo caso abbiamo utilizzato Javascript per creare una Google map che ricava i suoi dati da un file XML.*

## **1.3. Passaggi del codice**

### 1.3.1. Creazione della mappa

*Questa parte di codice (è un esempio), che ci permette di creare la mappa, la possiamo trovare nella guida Google Maps. Nel nostro progetto la mappa è stata creata in modo da visualizzare sulla mappa stessa tutti i marker.*

*Codice:*

```
<html>
<head>
<script async defer
src="https://maps.googleapis.com/maps/api/js?
key=YOUR\_API\_KEY&callback=initMap">
</script>
<style>
#map {
  height: 400px;
  width: 100%;
  }
</style>
</head>
<body>
<h3>My Google Maps Demo</h3>
<div id="map"></div>
<script>
function initMap() {
var uluru = {lat: -25.363, lng: 131.044};
```

```
var map = new google.maps.Map(document.getElementById('map'), {
  zoom: 4,
  center: uluru
});
}
</script>
<script async defer
  src="https://maps.googleapis.com/maps/api/js?
key=YOUR_API_KEY&callback=initMap">
  </script>
</body>
</html>
```

### **1.3.2. Acquisizione dei dati dall'XML con Ajax**

*Con Ajax nel nostro progetto riusciamo a ricavare latitudine e longitudine dei marker. Questo esempio di codice permette di ricavare dati dall'XML.*

*Codice:*

```
function loadDoc() {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      myFunction(this);
    }
  };
  xhttp.open("GET", "cd_catalog.xml", true);
  xhttp.send();
}
```

```

}
function myFunction(xml) {
  var i;
  var xmlDoc = xml.responseXML;
  var table="<tr><th>Title</th><th>Artist</th></tr>";
  var x = xmlDoc.getElementsByTagName("CD");
  for (i = 0; i <x.length; i++) {
    table += "<tr><td>" +
      x[i].getElementsByTagName("TITLE")
[0].childNodes[0].nodeValue +
      "</td><td>" +
      x[i].getElementsByTagName("ARTIST")
[0].childNodes[0].nodeValue +
      "</td></tr>";
  }
  document.getElementById("demo").innerHTML = table;
}

```

### **1.3.3. Marker**

*Tramite questa parte di codice avviene la creazione dei marker. Quando viene inserito un nuovo marker i bound della mappa vengono aggiornati così da poter visualizzare tutti i marker. Codice:*

```

var position = new google.maps.LatLng(lat, lng);
bounds.extend(position);

marker = new google.maps.Marker({

position: position,

map: map,

title: 'Marker'

```

```
});  
  
google.maps.event.addListener(marker, 'click', (function(marker, i) {  
  
    return function() {  
  
        infoWindow.setContent(description);  
  
        infoWindow.open(map, marker);  
  
    }  
  
})(marker, i));  
  
map.fitBounds(bounds);  
  
});
```

## **1.4. Google Maps API**

*Grazie alle API di Google Maps abbiamo gli strumenti necessari per creare e gestire le Google Maps. Per utilizzarle è stato necessario accedere ai servizi Google e richiedere una API key apposita per il nostro progetto.*

## **1.5. Fonti**

*Per il funzionamento di Ajax:*

[https://www.mrwebmaster.it/ajax/funzionamento-ajax\\_9959.html](https://www.mrwebmaster.it/ajax/funzionamento-ajax_9959.html)

*Per Javascript:*

[https://it.wikipedia.org/wiki/JavaScript#Usi\\_frequenti\\_nei\\_web\\_browser](https://it.wikipedia.org/wiki/JavaScript#Usi_frequenti_nei_web_browser)