



Javascript **jQuery**



Impariamo JQuery

Giulio Angiani
I.F.O.A. - Reggio Emilia



AJAX and JSON



Cos'è JSON

JSON è un acronimo che sta per

Java**S**cript **O**bject **N**otation

ovvero il modo con cui si implementa il concetto di Classe e Oggetto in Javascript

ESEMPIO:

Un esempio di formato JSON è il seguente:

```
{  
  "nome": "Giulio",  
  "cognome": "Angiani",  
  "eta": 47,  
  "linguaggi" : ["java", "php", "c++"]  
}
```

JSON



PERCHÈ JSON?

- per avere un formato semplice e chiaro per scambiare dati fra fonti diverse
- per avere dati auto definiti
- perchè indipendente da mittente e ricevente



Usare JSON

Usare il formato dati JSON permette ad un servizio che fornisce dati in rete (ad esempio un web service) di rispondere a richieste client inviando le informazioni senza preoccuparsi di come esse verranno **renderizzate** (mostrate all'utente)



Quindi lo stesso server può interfacciarsi con:

- un sito web
- una app mobile
- un altro software

senza modificare nulla nel modo di inviare dati

NOTA: è **necessario** definire l'interfaccia di richiesta al server ovvero il modo con il quale ogni **client** fa richiesta al server per ricevere le informazioni



Usare JSON in una pagina WEB

Abbiamo visto come via AJAX è possibile modificare **parti** di una pagina web invece di dover sempre ricaricare tutta la pagina

nello snippet seguente un esempio di chiamata ajax ad una pagina php, chiamata che permette al client di ricevere una risposta in formato html da sostituire nel contenuto di una div

JS

```
$(document).ready(function() {  
  
    $("#button").on("click", function(){  
        $.ajax({  
            type: "GET",                // il method  
            url: "loadDateTime.php",    // la pagina  
            data: {},                  // eventuali parametri  
            dataType: "json",          // necessario a JS se ricevo i dati in formato JSON  
            success: function(risposta) {  
                $("#container").html(risposta.testo); // risposta è un oggetto JSON non un testo!  
            },  
            error: function(){  
                alert("Ooops! Qualcosa è andato storto...");  
            }  
        });  
    });  
});
```



Usare JSON in una pagina WEB

La parte HTML deve contenere l'elemento sul quale è scatenato l'evento e quella sulla quale va applicata la modifica

Nel nostro caso

```
<body style='padding:30px'>
  <input type='button' class='btn-success rounded' id='button' value="Vedi l'ora esatta">

  <div id="container"></div>
</body>
```

HTML

e la parte PHP per la restituzione dei dati in formato JSON

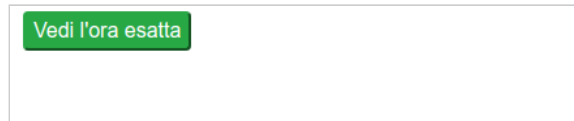
```
<?php
    $risposta = Array(
        "esito" => "OK",
        "testo" => "Sono le ore " . strftime("%H:%M:%S del %Y-%m-%d")
    );
    // la funzione predefinita "json_encode" traduce un array associativo in una stringa in formato JSON
    $stringajson = json_encode($risposta);
    exit($stringajson);
    // la risposta del server non è più una pagina HTML ma un insieme di dati
    // potrà essere fruita da qualsiasi tipo di client
?>
```

PHP



Usare JSON in una pagina WEB

Il risultato che otteniamo è che cliccando sul bottone verde...



otteniamo questo...



Usare JSON in una pagina WEB - 2

Carichiamo i dati di una tabella via JSON

Vogliamo ottenere una tabella cliccando sul bottone verde:

da così

Visualizza la lista degli articoli presenti

a così

Visualizza la lista degli articoli presenti

M001	Monitor HP 19"	Hardware	199.99
M002	Monitor HP 21"	Hardware	299.99
WC03	Webcam Trust 1024x768 Full HD	Hardware	59.99
W010	Windows 10 Pro Student Edition	Software	39.99
MSO1	Microsoft Office 365 Student Edition	Software	49.99



Usare JSON in una pagina WEB - 2

Prepariamo l'html per mostrare i dati...

HTML

```
<body style='padding:30px'>
  <input type='button' class='btn-success rounded' id='button' value="Visualizza la lista degli articoli presenti">

  <div id="container"></div>
</body>
```

e la parte PHP per la restituzione dei dati in formato JSON

PHP

```
<?php
include("config/connessione.php");
include("tools.php");

$filtro_categoria = ""; // se ricevo via GET il parametro categoria faccio il filtro nella query
if (isset($_GET["categoria"])) {
    $filtro_categoria = " where categoria = '{$_GET["categoria"]}' ";
}
// OK con la connessione
$sql = "SELECT * FROM items $filtro_categoria";
$resultset = $conn->query($sql);
//var_dump($resultset);
$righe = mysqli_fetch_all($resultset, MYSQLI_ASSOC);
exit(json_encode($righe));
?>
```



Usare JSON in una pagina WEB - 2

Il codice JavaScript per la gestione asincrona del risultato JSON è

JAVASCRIPT

```
$("#button").on("click", function(){
    $.ajax({
        type: "GET",                // il method
        url: "../negozio/json_items.php",    // la pagina
        data: {},                  // eventuali parametri
        dataType: "json",          // eventuali parametri
        success: function(risposta) {
            // costruisco l'html nel client e non nel server!
            var tabella = "<table class='table table-striped table-bordered w-100 m-auto' id='articoli'>";
            for (i in risposta) {
                obj = risposta[i];
                tabella += "<tr><td>"+obj.codice+"</td><td>"+obj.descrizione+"</td>";
                tabella += "<td>"+obj.categoria+"</td><td>"+obj.prezzo+"</td></tr>";
            }
            tabella += "</table>";
            $("#container").html(tabella); // risposta è un oggetto JSON non un testo!
        },
        error: function(){
            alert("Ooops! Qualcosa è andato storto...");
        }
    });
});
```

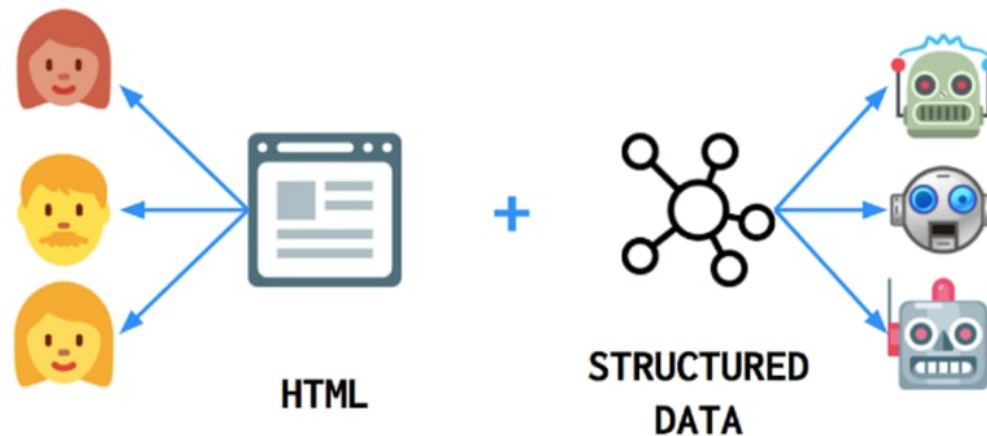


Ma perché? Qual è lo scopo vero?

HTML è interpretabile solamente da un browser!

JSON è un formato di scambio dati che può essere fruito da qualsiasi client

- web page
- web app
- mobile app (sia Android che iPhone)
- programma generico



Ma perché? Qual è lo scopo vero? - 2

Di seguito un esempio di codice java che interagisce con un servizio che espone dati in formato JSON e ne fa una stampa a video

JAVA

```
HttpClient client = HttpClient.newHttpClient();
String url = "http://localhost/ifo/json/loadDateTime.php";
HttpRequest request = HttpRequest.newBuilder()
    .uri(URI.create(url)) // inicializza la chiamata sulla url giusta
    .GET()                // metodo di chiamata
    .build();             // completa la chiamata
HttpResponse<String> response = client.send(request, HttpResponse.BodyHandlers.ofString());

String json = response.body(); // la risposta del server è sempre la stessa! Una stringa!
JSONObject obj = new JSONObject(response.body()); // viene trasformata in oggetto JSON
String esito = (String)obj.get("esito"); // recupero il dato 'esito' chiamandolo per chiave
String dataora = (String)obj.get("testo"); // recupero il dato 'testo' chiamandolo per chiave
System.out.println("Esito dell'operazione : " + esito);
System.out.println("Data e Ora corrente : " +dataora);
```

OUTPUT

```
Esito dell'operazione : OK
Data e Ora corrente : Sono le ore 17:13:49 del 2020-09-14
```



Documentare le API

Quando si espone un servizio JSON è fondamentale **documentare** il servizio stesso

Per **documentazione** si intende spiegare nel dettaglio a chi scrive come chiamare il servizio:

Dettagliatamente:

- l'url da chiamare
- il metodo http da usare (GET, POST, e altri...)
- eventuali parametri da utilizzare e perché
- il formato della restituzione dei dati (in linguaggio naturale)

Nel nostro primo esempio il servizio andrebbe documentato come segue:

- **URL** : `http://localhost/ifoa/json/loadDateTime.php`
- **Metodo** : GET
- **Parametri** : nessuno
- **Formato** : Oggetto JSON contenente due chiavi
 - La chiave **esito** può avere valore "OK" o "KO" (Se vale "OK" la comunicazione è andata a buon fine, altrimenti si sono verificati dei problemi)
 - La chiave **testo** contiene la stringa nel formato "Sono le ore HH:MM:SS del YYYY-MM-DD" nel caso esito sia OK, contiene il testo dell'errore nel caso l'esito sia KO



Documentare le API - Esercizio

Scrivere la documentazione per l'esempio n. 2 la cui risposta è indicata di seguito

JSON

```
[  
  {"id": "1", "codice": "M001", "descrizione": "Monitor HP 19", "prezzo": "199.99", "categoria": "Hardware"},  
  {"id": "2", "codice": "M002", "descrizione": "Monitor HP 21", "prezzo": "299.99", "categoria": "Hardware"},  
  {"id": "3", "codice": "WC03", "descrizione": "Webcam Trust 1024x768 Full HD", "prezzo": "59.99", "categoria": "Hardware"},  
  {"id": "4", "codice": "W010", "descrizione": "Windows 10 Pro Student Edition", "prezzo": "39.99", "categoria": "Software"},  
  {"id": "5", "codice": "MS01", "descrizione": "Microsoft Office 365 Student Edition", "prezzo": "49.99", "categoria": "Software"}  
]
```

suggerimento: controlla il codice php per capire cosa si può ricevere....

Tempo: 10 minuti



Documentare le API - Soluzione

- **URL :** `http://localhost/ifo/negozio/json_items.php`
- **Metodo :** GET
- **Parametri :**
 - **categoria** (facoltativo) se valorizzato permette di filtrare i dati della categoria indicata
- **Formato :** Oggetto JSON contenente una lista di oggetti (la lista potrebbe anche essere vuota)
 - Ogni oggetto contiene 5 chiavi:
 - La chiave **id** è un numero intero e indica l'identificativo interno dell'articolo selezionato
 - La chiave **codice** è una stringa e indica il codice dell'articolo
 - La chiave **descrizione** è una stringa e indica la descrizione dell'articolo
 - La chiave **prezzo** è un numero double e indica il prezzo dell'articolo
 - La chiave **categoria** è una stringa e indica la categoria dell'articolo



Esercizi

1) Realizzare e documentare un servizio web che ricevendo in ingresso due parametri con valori interi, restituisca la somma in formato JSON

2) Realizzare e documentare un servizio web che ricevendo in ingresso due parametri "username" e "password" restituisca in formato JSON l'esito del riconoscimento dell'utente

3) Realizzare e documentare un servizio web che ricevendo in ingresso due parametri "username" e "password" restituisca in formato JSON i dati del contocorrente dell'utente riconosciuto e la lista dei movimenti effettuati sul conto¹

4) Scrivere un programma Java che chiede all'utente username e password e esegue l'operazione di riconoscimento tramite il servizio implementato al punto 2²



1) nota: fare riferimento all'esercizio conticorrenti

2) suggerimento: inviare i dati via querystring e non via POST



Esercizi - Soluzioni

Esercizio 1

- Realizzare e documentare un servizio web che ricevendo in ingresso due parametri con valori interi, restituisca la somma in formato JSON

Una possibile soluzione è la seguente:

documentazione

- **URL :** <SOME_URL>/somma.php
- **Metodo :** GET
- **Parametri :**
 - **a** intero
 - **b** intero
- **Formato risposta :**
 - Oggetto JSON con una sola chiave **risultato** associata a valore intero che rappresenta la somma dei due numeri passati in input
 - Esempio d'uso
 - chiamata <SOME_URL>/somma.php?a=2&b=3
 - risposta { "risultato": 5 }



Esercizi - Soluzioni

Esercizio 1

- Realizzare e documentare un servizio web che ricevendo in ingresso due parametri con valori interi, restituisca la somma in formato JSON

Una possibile implementazione del servizio è la seguente:

```
<?php
    $risposta = Array(
        "risultato" => $_GET["a"] + $_GET["b"]
    );
    exit(json_encode($risposta));
?>
```

PHP



Esercizi - Soluzioni

Esercizio 2

2) Realizzare e documentare un servizio web che ricevendo in ingresso due parametri "username" e "password" restituisca in formato JSON l'esito del riconoscimento dell'utente

Una possibile soluzione è la seguente:

documentazione

- **URL :** <SOME_URL>/check.php
- **Metodo :** POST
- **Parametri :**
 - **usr** string
 - **pwd** string
- **Formato risposta :**
 - Oggetto JSON con una sola chiave **risultato** associata al valore "OK" se il riconoscimento è andato a buon fine, "KO" altrimenti
 - Esempio d'uso
 - chiamata ##<SOME_URL>/check.php
 - parametri pwd="giulio" pwd="ifoa"
 - risposta { "risultato": "OK" }



Esercizi - Soluzioni

Esercizio 2

2) Realizzare e documentare un servizio web che ricevendo in ingresso due parametri "username" e "password" restituisca in formato JSON l'esito del riconoscimento dell'utente

Una possibile implementazione del servizio è la seguente:

PHP

```
// recupero dati da POST
$usr = $_POST["usr"];
$pwd = $_POST["pwd"];
// query su DB
$sql = "SELECT * FROM studenti WHERE username = '$usr' and password = '$pwd' ";
$resultset = $conn->query($sql);
$righe = mysqli_fetch_all($resultset, MYSQLI_ASSOC);
if (count($righe) > 0) {
    exit(json_encode(Array("risultato" => "OK")));
}
else {
    exit(json_encode(Array("risultato" => "KO")));
}
```



Esercizi - Soluzioni

Esercizio 3

- Realizzare e documentare un servizio web che ricevendo in ingresso due parametri "username" e "password" restituisca in formato JSON i dati del contocorrente dell'utente riconosciuto e la lista dei movimenti effettuati sul conto

Una possibile soluzione è la seguente:

documentazione

- **URL :** <SOME_URL>/movimenti.php
- **Metodo :** POST
- **Parametri :**
 - **usr** string
 - **pwd** string
- **Formato risposta :**
 - oggetto JSON con due chiavi
 - **esito** (string) "OK" se utente riconosciuto, "KO" altrimenti
 - **count** (intero) che indica il numero di movimenti restituiti, -1 se esito="KO"
 - **movimenti** lista di oggetti JSON dove ogni oggetto racchiude le informazioni relative ad un movimento del conto, [] se esito="KO"



Esercizi - Soluzioni

Esercizio 3

- Esempio d'uso
- chiamata `##<SOME_URL>/movimenti.php`
- parametri `pwd="valerossi" pwd="quarantasei"`
- risposta
- ```
{ "esito": "OK", "count": 4, "movimenti": [{ "id": "1", "numeroconto": "12345", "data": "2020-03-05",
"valoremovimento": "5000.00" }, { "id": "2", "numeroconto": "12345", "data": "2020-03-06",
"valoremovimento": "-2500.00" }, { "id": "3", "numeroconto": "12345", "data": "2020-04-02",
"valoremovimento": "-1000.00" }, { "id": "4", "numeroconto": "12345", "data": "2020-04-20",
"valoremovimento": "2000.00" }] }
```



# Esercizi - Soluzioni

## Esercizio 3

Una possibile implementazione del servizio è la seguente:

PHP

```
<?php
 session_start();
 include("config/connessione.php");

 // recupero dati da POST
 $usr = $_POST["usr"]; $pwd = $_POST["pwd"];

 // query su DB
 $sql = "SELECT * FROM conticorrenti WHERE username = '$usr' and password = '$pwd' ";
 $resultset = $conn->query($sql);
 $righe = mysqli_fetch_all($resultset, MYSQLI_ASSOC);
 if (count($righe) == 0) { // KO LOGIN
 exit(json_encode(Array("esito"=>"KO", "count"=>-1, "movimenti"=>Array())));
 }
 // OK LOGIN
 $numeroconto = $righe[0]["numeroconto"];
 $sql = "SELECT * FROM movimenti WHERE numeroconto = '$numeroconto'";
 $resultset = $conn->query($sql);
 $righe = mysqli_fetch_all($resultset, MYSQLI_ASSOC);
 exit(json_encode(Array("esito"=>"OK", "count"=>count($righe), "movimenti"=>$righe)));
?>
```



# Esercizi - Soluzioni

## Esercizio 4

Scrivere un programma Java che chiede all'utente username e password e esegue l'operazione di riconoscimento tramite il servizio implementato al punto 2<sup>2</sup>

Una possibile implementazione del servizio è la seguente:

JAVA

```
HttpRequest request = HttpRequest.newBuilder()
 .uri(URI.create("http://localhost/ifoajson/check.php?usr=giuang&pwd=ifoajson"))
 .GET() // GET is default
 .build();

HttpResponse<String> response = client.send(request, HttpResponse.BodyHandlers.ofString());

// parse JSON string
String json = response.body();
JSONObject obj = new JSONObject(json);
if (obj.get("risultato").equals("OK"))
 System.out.println("OK riconosciuto");
else
 System.out.println("Username o password errata");
```







Giulio Angiani  
I.F.O.A. - Reggio Emilia