

MySQL

Esercizio 1

Implementare un servizio di API, che permette di accedere alle informazioni contenute nel dataset popolazione e famiglie, strutture familiari, famiglie, ampiezza delle famiglie, presente sul portale <http://dati.istat.it>. In particolare, l'API deve permettere la consultazione per anno e per numero di componenti della famiglia, il valore in migliaia.

Soluzione

Strutturiamo l'esercizio come un progetto, che poi utilizzeremo anche negli esercizi successivi. Nella htdocs di XAMPP creiamo una cartella progetto, dentro cui creiamo la cartella api, che conterrà tutte le api php che andremo a sviluppare

Creiamo ora un file config.php che posizioniamo nella cartella api. Questo file contiene i parametri di configurazione della connessione al database e aprirà la connessione al database. Il file config.php dovrà essere incluso in tutti gli script php che effettueranno l'accesso al database. Apriamo il file config.php ed effettuiamo la connessione al database:

```
$database = "LPW";
$username = "root";
$password = "";
$servername = "localhost";
$conn = mysqli_connect($servername,$username,$password,$database) or
die("Errore connessione ".mysqli_connect_error($conn));
```

La connessione al database è ora aperta ed è memorizzata nella variabile \$conn. Ora possiamo passare all'implementazione vera e propria della nostra API. Nella cartella API creiamo un file ampiezza.php. Come prima cosa, includiamo il file config.php che ci permette di aprire la connessione al database:

```
<?php
include('config.php');
?>
```

Una volta aperta la connessione, ricordiamoci subito di chiuderla in fondo, altrimenti poi potremmo dimenticarcelo:

```
<?php
include('config.php');
// corpo di ampiezza.php
mysqli_close($conn);
?>
```

Ora, dobbiamo sostituire il commento // corpo di ampiezza.php con il corpo vero e proprio della api. Ricordiamoci di lasciare la closeDB in fondo allo script. Supponiamo che nel nostro database LPW, abbiamo memorizzato la tabella con il nome AmpiezzaFamiglie. Dalla tabella dobbiamo estrarre per ogni anno e per numero di componenti il valore in migliaia. La query che dobbiamo fare è la seguente:

```
SELECT ComponentiFamiglia,Valore,Periodo FROM AmpiezzaFamiglie WHERE
Misura = 'Valori in migliaia'
```

Possiamo passare questa query alla funzione `mysqli_query` e caricare il risultato tramite la funzione `mysqli_fetch_assoc`.

```
$sql = SELECT ComponentiFamiglia,Valore,Periodo FROM AmpiezzaFamiglie
WHERE Misura = 'Valori in migliaia';
$resultato = mysqli_query($conn,$sql);
if (!$resultato) die("Errore query: ".mysqli_error($conn));

while($row=mysqli_fetch_assoc($resultato))
{
    // corpo while
}
```

Il nostro obiettivo è ora mostrare a video il risultato della query. Prima del ciclo while creiamo una variabile `$return` di tipo array, che conterrà il risultato da mostrare a video. Dentro questa variabile mettiamo le righe che carichiamo tramite la `mysqli_fetch_assoc`. Inseriamo anche una stampa del risultato:

```
$resultato = mysqli_query($conn,$sql);
if (!$resultato) die("Errore query: ".mysqli_error($conn));

$return = array();
while($row=mysqli_fetch_assoc($resultato))
{
    $return[] = $row;
}
var_dump($return);
```

Guardando la stampa dell'array, si può notare che il periodo e il valore sono stampati come stringhe. Questo non va bene, perché in realtà si tratta di decimali. Occorre quindi sistemare il codice in modo da convertire le stringhe in float. Per fare questo, possiamo semplicemente fare una conversione da stringa a float in corrispondenza di questi valori.

```
$resultato = mysqli_query($conn,$sql);
if (!$resultato) die("Errore query: ".mysqli_error($conn));

$return = array();
while($row=mysqli_fetch_assoc($resultato))
{
```

```

    $row['Valore'] = floatval($row['Valore']);
    $row['Periodo'] = floatval($row['Periodo']);
    $return[] = $row;
}

```

Per rendere i dati accessibili a tutti i tipi di client, la nostra API dovrebbe fornire i risultati in JSON. JSON (JavaScript Object Notation) è un formato utilizzato per lo scambio dei dati in applicazioni client-server. Il JSON è un formato che serve per codificare un array in una stringa. In JSON un array numerico è rappresentato come un elenco di elementi separati da virgola e racchiusi tra parentesi quadre. Ad esempio:

```
[ 'cane', 'gatto', 'topo' ]
```

Un array associativo invece è rappresentato da coppie chiave valore separate dalla virgola e racchiuse tra parentesi graffe. Ogni coppia chiave valore è rappresentata nel seguente modo:

chiave : valore

Un esempio di array associativo è il seguente:

```
{ 'animale' : 'cane', 'fiore' : 'rosa', 'frutto' : 'mela' }
```

In PHP, affinché una API ritorni un array in JSON è necessario eseguire due passi:

1. includere all'inizio dell'API la chiamata ad una funzione che setta il formato del risultato in json

```
header('Content-Type: application/json');
```

2. codificare l'array risultato in json attraverso la funzione json_encode e stampare il risultato:

```
echo json_encode($return);
```

Nel nostro caso, togliamo la stampa fatta con la var_dump e inseriamo il codice appena definito, facendo attenzione ad inserire la chiamata alla header all'inizio del documento e la json_encode al posto della var_dump. Il risultato è il seguente:

```

<?php
include('config.php');
header('Content-Type: application/json');

$sql = "SELECT ComponentiFamiglia,Valore,Periodo FROM AmpiezzaFamiglie
WHERE Misura = 'Valori in migliaia'";

$resultato = mysqli_query($conn,$sql);
if (!$resultato) die("Errore query: ".mysqli_error($conn));

$return = array();
while($row=mysqli_fetch_assoc($resultato))

```

```
{
    $row['Valore'] = floatval($row['Valore']);
    $row['Periodo'] = floatval($row['Periodo']);
    $return[] = $row;
}

echo json_encode($return);
mysqli_close($conn);

?>
```

Visualizzando il risultato dal browser, notiamo che non è molto chiaro. Per visualizzare meglio un json, possiamo installare un plugin per il nostro browser. Il plugin si chiama JSON View.

Per installare il plugin su Google Chrome, andiamo nel menu Impostazioni, selezioniamo estensioni, poi prova altre estensioni. Nella barra di ricerca digitiamo jsonview e poi installiamolo.

Per installare il plugin su Safari, selezionare il menu Safari, poi estensioni, cercare JSON ed installare json-lite.

Per installare il plugin su Firefox, dal menu in alto a destra selezionare componenti aggiuntivi, quindi estensioni e cercare jsonview.

Esercizio 2

Modificare il file `ampiezza.php` in modo da ricevere in ingresso tramite GET un parametro `misura`. In base al valore del campo `misura`, il file restituisce un risultato diverso:

- se `misura = caratteristiche`, lo script restituisce i valori per 100 famiglie con le stesse caratteristiche
- se `misura = migliaia` lo script restituisce i valori in migliaia