**REST API**

**REST** ( **Re**presentational **S**tate **T**ransfer) è un tipo di architettura software del web, capace di trasmettere dati per mezzo del protocollo **HTTP**.

Permette la creazione di **Web API**, che sono interfacce di programmazione per comunicare facilmente e in modo standard, soprattutto in modalità **client-server**.

Vedremo un esempio di architettura REST client/server.

Nell’esempio si tratterà un anagrafico online, i dati sono memorizzati in un database e gestiti tramite servizi REST. Il client richiede un servizio al server che lo fornisce. Le richieste e le risposte che il client e il server si scambiano viaggiano tramite il protocollo HTTP.

Si usa il linguaggio di programmazione web PHP e il DBMS MySQL.

Si realizzeranno le quattro operazioni chiamate CRUD (Create, Read, Update, Delete):

* creare un nuovo utente da aggiungere all’anagrafico (*Create*);
* leggere un utente dall’anagrafico (*Read*);
* aggiornare le informazioni di un utente dell’anagrafico (*Update*);
* eliminare un utente dall’anagrafico (*Delete*);

associate ai metodi HTTP **GET, POST, PUT e DELETE**.

**Metodo http Operazione CRUD Descrizione**

POST Create Crea una nuova risorsa

GET Read Ottiene una risorsa esistente

PUT Update Aggiorna una risorsa

DELETE Delete Elimina una risorsa

Passiamo ora a MySQL per la creazione del DataBase “anagrafico” che conterrà una sola tabella “utente”:

CREATE TABLE `utente` (

`iD` int primary key,

`Nome` varchar(50) NOT NULL,

`eta` int

);

Si inseriscono dei dati:

INSERT INTO utente VALUES (1, "Mario", 18);

INSERT INTO utente VALUES (2, "Ada", 17);

Per l’esempio sono presenti dei file posti in una cartella chiamata “ServerREST”, il file “database.php” fornisce i comandi per collegarsi al database “anagrafico”, in pratica contiene una classe “Database”, con un metodo pubblico “getConnection()”.

Il file “Utente.php” è una classe che descrive la struttura della tabella e contiene i metodi con le operazioni che si vogliono fornire come servizio, read, create, update e delete.

If file “read.php” restituisce tutti i dati presenti nel database in formato json utilizzando il metodo “read” della classe “Utente”.

**Per attivare il servizio** aprire la pagina http://localhost/ServerREST/read.php che restituisce i dati letti in formato json, oppure si attiva una chiamata HTTP GET in un qualsiasi linguaggio di programmazione.

In alternativa si possono usare il plug-in **Postman** del browser Chrome o il plug-in **RESTClient** di Firefox inserendo sempre l’indirizzo [*http://localhost/ServerREST/read.php*](http://localhost/ServerREST/read.php)

Per testare il codice, con RESTClient o Postman, si deve:

1. impostare il metodo http(Get o Post o Put o Delete)
2. immettere l'URL della richiesta
3. selezionare il body per immettere i dati dell’utente
4. selezionare *raw*
5. inserire il JSON.

Il JSON nel caso di POST (inserimento) sarà (per esempio):

{

"iD" : 10,

"Nome" : "Luigi",

"Eta" : 25

}

La richiesta al server può essere fatta in vari modi:

http://localhost/ServerREST/read.php/utente/1 per richiedere i dati dell’utente con ID 1;

http://localhost/ServerREST/read.php*?sort=eta&order=asc* per richiedere i dati ordinati per età e in modo ascendente, per richiedere i parametri da PHP:

*$var= $\_SERVER['QUERY\_STRING']; //$var =* utente*?sort=eta&order=asc*

*$var1= $\_GET['sort']; //$var1 = eta*

In PHP esiste una variabile nell'array globale $\_SERVER, che consente di determinare quale metodo è stato utilizzato per fare una richiesta, esempio d’uso:

*if ($\_SERVER[‘REQUEST\_METHOD’]==’PUT’) ….. // o ‘DELETE’ o ‘POST’ o ‘GET’*

Per le chiamate REST da PHP si può usare la libreria ***curl***.