

Manuale d'uso Web Service REST

Introduzione:

Il progetto del Web Service REST è stato realizzato su richiesta dei docenti di Informatica durante l'anno scolastico (in data 13 marzo 2024). Sviluppato come endpoint REST in PHP per le richieste HTTP di base (GET, POST, PUT e DELETE).

Utilizzo e funzionamento API:

Il risultato della query effettuata sul server web verrà trasformato nel formato JSON o XML, a seconda del valore di `HTTP_ACCEPT` impostato nell'intestazione della richiesta inviata dal client. Se non specificato, verrà utilizzato JSON come formato predefinito.

Le richieste alle API devono essere effettuate tramite l'URL per la richiesta GET. Per POST, PUT e DELETE, invece, viene utilizzato il corpo della richiesta. Le informazioni necessarie per la gestione del database devono essere inviate nel corpo della richiesta per ogni tipo di operazione (eccetto GET):

- Per GET, verranno restituiti i dati nel formato selezionato. Se l'URL contiene solo la posizione del file richiesto, verranno restituite tutte le entità di tutte le tabelle del database. Se viene specificata anche la tabella, verrà visualizzata l'intera tabella. Se viene fornito un numero (considerato come ID), verrà restituito solo quell'elemento, in caso sul database non esista una tale tabella o un tale elemento di tabella verrà restituito il messaggio "nessun riscontro".
- Per POST, sono necessari tutti i campi dell'entità e il nome della tabella stessa. In caso contrario, se i campi sono compilati in modo errato o con valori non validi, la richiesta non verrà soddisfatta. Verranno effettuati controlli sulla completezza delle informazioni e, se presente un ID, non verrà inserito poiché la chiave primaria è auto-incrementante e si imposta automaticamente (è ignorato il campo ID in modo che ci sia o meno, il programma possa continuare senza dare errori).
- Per PUT, si applicano gli stessi controlli di POST. In questo caso, è necessario specificare solo la tabella su cui effettuare l'azione, l'ID e almeno un altro campo per evitare richieste vuote e inutili (per evitare che il server rimanga intasato).
- Per DELETE, si applicano gli stessi controlli di POST e PUT. DELETE richiede solo due campi: il nome della tabella e l'ID dell'elemento da eliminare dal database. Gli altri campi verranno ignorati poiché non richiesti.

Esito dell'utilizzo delle API:

Nel caso in cui la richiesta manchi di parametri o contenga informazioni errate, verrà restituita una richiesta fallita con un codice di errore appartenente alla famiglia 400. In caso di successo invece si avrà 200 come codice di risposta.

Esempio di contenuto del corpo della richiesta POST in formato json:

```
{"nome_tabella":"computer", "id":"2", "marca":"Dell", "modello":"Ultrabook XPS 13 9320", "numero_di_serie":"7654321", "tipo":"server", "anno_acquisto":"2022-09-20 00:00:00", "specifiche_tecniche":"molto interessante"}
```

Interfaccia client in C# (in sviluppo):

Il progetto prevede anche l'utilizzo da parte di un web client in C# delle API citate in precedenza (GET, POST, PUT e DELETE), ancora in fase di sviluppo. Attraverso un'interfaccia grafica, tutte le tabelle del database verranno caricate dinamicamente in una combobox. Quando viene selezionata una tabella, verrà visualizzata su una listview. In questa fase del progetto, verrà utilizzata un'API per eseguire una richiesta GET.

È possibile passare il cursore sulla listview e selezionare l'elemento da gestire. Verranno create delle textbox che si adattano agli attributi della tabella per una migliore visualizzazione e, eventualmente, modifica (anche l'ID, ovvero la chiave primaria, verrà mostrato ma non sarà modificabile o cancellabile). Sono presenti 3 pulsanti: Modifica (PUT), Crea (POST), Cancella (DELETE), che eseguiranno le rispettive API e azioni.

Note e avvertenze finali:

Il sistema è ancora soggetto a bug e problematiche di vario tipo, è inoltre modellato per funzionare con il database di informazioni denominato "allaboutpc", in caso si voglia usarne un altro si è pregati di usare il sistema con cautela, in quanto potrebbe non essere calibrato a sufficienza per un altro database nonostante gli svariati controlli.