

Indirizzo **INFORMATICA**

Crud Server

Un CRUD server è un server che, **indipendentemente dalla collezione richiesta dal client**, esegue una azione diversa a seconda del tipo di chiamata sulla base del seguente schema in cui, in tutte le chiamate, id viene passato al server sempre come parte della risorsa:







GET / restituisce l'elenco completo dei record.
GET /id restituisce il record avente l'id indicato
GET / + {filter} restituisce tutti i record che soddisfano al filtro indicato
Questi metodi restituiscono tutti i campi ma il client leggerà solo **_id e name**
POST inserisce nella collezione il record ricevuto come parametro
DELETE /id elimina il record avente l'id indicato
DELETE / + {filter} elimina tutti i record che soddisfano al filtro indicato
PATCH /id {action} aggiorna il record avente l'id indicato sulla base del json ricevuto come parametro.
Esegue cioè una **action** di tipo **\$set**
PUT /id {action} dovrebbe sostituire il record avente l'id con il json ricevuto come parametro.
In realtà esegue un **update** del record eseguendo una **action generica**
PUT / + {filter:{}, action:{}} esegue la action indicata sul filter indicato



Collection: **unicorns** Documents: **20** Add

unicorns

vampires

_id	name	
61a9441a50bd7e08afc1bc7c	Dracula	 
61a9441a50bd7e08afc1bc7d	Jurek	 
61a9441a50bd7e08afc1bc7e	Nosferatu	 

_id: 61a9441a50bd7e08afc1bc7d
name: Jurek
naz: Cecoslovacchia

1) All'avvio il client richiede al server l'elenco delle collezioni contenute all'interno del database di lavoro e le visualizza tramite appositi Option Button nel frame di sinistra.

2) Elenco dei Record: In corrispondenza del **click su una collezione**, l'applicazione deve:

- Aggiornare le due label superiori con il nome della collezione corrente ed il numero di documenti contenuti
- Visualizzare all'interno della tabella centrale l'elenco di tutti i record e tutti i campi restituiti dal server. In corrispondenza di questa richiesta **il server invia solo i campi _id e name**. Si assume che il campo name sia sempre esistente. I tre pulsantini sono visualizzati all'interno di una unica cella con tre tag DIV automaticamente gestiti da CSS
- Se la collezione richiesta è unicorns, visualizzare la sezione relativa ai filtri (inizialmente nascosta)

ISTITUTO TECNICO INDUSTRIALE STATALE "G. VALLAURI"

▲ Via S. Michele, 68 – 12045 Fossano

Indirizzo **INFORMATICA**

- 3) **Filtri** *Nel solo caso della tabella unicorns*, visualizzare anche la sezione relativa ai Filtri di ricerca. In corrispondenza del click sul pulsante FIND inviare una richiesta per il servizio **/unicorns** passando come parametro GET aggiuntivo un filtro del tipo: {"hair":"blonde", "gender":"m"}. In caso di assenza dei parametri la query restituirà l'intero recordset
- 4) **Dettagli**: In corrispondenza del **click su una voce dell'elenco** (primi due campi) l'applicazione visualizza nel riquadro di destra tutti i dettagli relativi alla voce selezionata, riportando le chiavi in neretto. Lato server esegue un **FINDONE** sull'ID che restituisce NON un vettore ma un semplice json
- 5) **Aggiunta di un nuovo record** In corrispondenza del click sul pulsante **ADD**, l'applicazione aggiunge nel riquadro di destra una textarea con in coda un pulsante INVIA. In corrispondenza del click sul pulsante INVIA l'applicazione controlla la validità del json inserito e, in caso affermativo, invia la richiesta di inserimento al server. In corrispondenza dell'OK aggiorna la tabella centrale

Pulsantini a fianco dei record

- 6) In corrispondenza del **click sul pulsante Elimina**, dopo apposita richiesta di conferma, l'applicazione invia al server la richiesta di eliminazione del record selezionato. In corrispondenza dell'OK aggiorna la tabella centrale
- 7) In corrispondenza del **click sul pulsante Aggiorna** l'applicazione aggiunge nel riquadro di destra una textarea con in coda un pulsante INVIA. Inoltre ricopia all'interno della textArea tutti i dettagli del record corrente in modo che l'utente li possa modificare (**escluso l'ID**). In corrispondenza del click sul pulsante INVIA l'applicazione controlla la validità del json inserito e, in caso affermativo, invia la richiesta di aggiornamento al server. In corrispondenza dell'OK aggiorna la tabella centrale
- 8) In corrispondenza del **click sul pulsante Sostituisci** vengono eseguite le stesse azioni del punto precedente. La differenza consiste soltanto nel fatto che eventuali campi cancellati in fase di editing:
- Nel caso dell'update mantengono il loro vecchio valore
 - Nel caso del replace vengono rimossi

In realtà il comportamento viene modificato come indicato all'inizio