

Programmazione Java per Dispositivi Mobili

ESERCIZIO 2: Lista della Spesa

Autore: Danilo Croce - Pierpaolo Loreti

Si realizzi la seguente interfaccia REST per manipolare un elenco di elementi (*item*) mediante metodi HTTP.

L'API ha i seguenti endpoint:

- `GET /items`: Recupera l'elenco totale degli elementi.
- `POST /items`: Aggiunge un nuovo elemento all'elenco.
- `GET /items?id=ID`: Recupera i dettagli di un elemento specifico.
- `PUT /items`: Aggiorna un elemento specifico.
- `DELETE /items?id=ID`: Elimina un elemento specifico.

L'endpoint `GET /items` restituisce l'elenco degli oggetti *Item* disponibili. Se la richiesta ha successo, verrà restituito un array JSON di oggetti *Item* con lo stato di risposta 200. Il metodo restituisce la lista degli *Item* in formato JSON.

L'endpoint `GET /items?id=ID` consente di recuperare un oggetto *Item* esistente identificato tramite l'*ID* fornito nella richiesta. L'*ID* dell'oggetto *Item* da recuperare deve essere fornito come parametro nella richiesta. Se l'oggetto *Item* esiste, verrà restituito insieme allo stato di risposta 200. In caso contrario, se l'oggetto *Item* non esiste e verrà restituito uno stato di risposta 404. Ciò significa che l'*ID* fornito nella richiesta non corrisponde a nessun oggetto *Item* esistente. Il metodo restituisce l'*Item* in formato JSON.

L'endpoint `POST /items` consente di aggiungere un nuovo oggetto *Item*. La richiesta deve includere un oggetto *Item* valido nel corpo della richiesta (ad esempio usando un parametro chiamato `json` il cui valore è l'*item* in formato JSON) e deve avere l'intestazione `Content-Type: application/json`. In input, l'oggetto deve avere TUTTI i campi inizializzati e il valore `ID=-1` poichè esso verrà assegnato automaticamente dal sistema una volta inserito nella collezione. Se la richiesta ha successo, verrà restituito l'oggetto *Item* appena creato (con un ID assegnato opportunamente dal sistema) con lo stato di risposta 201. In caso di errore, verrà restituito lo stato di risposta 422 (`Unprocessable Content`). Ciò può accadere se la richiesta non è valida, ad esempio se è stato fornito un formato non valido.

L'endpoint `PUT /items` consente di aggiornare un oggetto *Item* esistente. La richiesta deve includere un oggetto *Item* valido nel corpo della richiesta e l'intestazione `Content-Type: application/json`. In particolare, l'*ID* dell'oggetto *Item* inviato deve essere lo stesso di uno oggetto già presente nella collezione, affinché il sistema sostituisca l'oggetto corretto. Se la richiesta ha successo, verrà restituito lo stato di risposta 200. Se l'oggetto *Item* da aggiornare non esiste nella collezione, verrà restituito uno stato di risposta

404. Nel caso in cui è stato fornito in unput un formato non valido, verrà restituito lo stato di risposta 422.

L'endpoint `DELETE /items?id=ID` consente di eliminare un oggetto `Item` esistente identificato tramite l'`ID` fornito nella richiesta. L'`ID` dell'oggetto `Item` da eliminare deve essere fornito come parametro nella richiesta. Se l'oggetto `Item` esiste e viene eliminato con successo, verrà restituito uno stato di risposta 204 con un corpo vuoto. In caso contrario, se l'oggetto `Item` non esiste, verrà restituito uno stato di risposta 404. Ciò significa che l'`ID` fornito nella richiesta non corrisponde a nessun oggetto `Item` esistente.

Inoltre, è possibile che si verifichino errori interni al server. In questo caso, verrà restituito uno stato di risposta 500.

L'oggetto `Item` è un oggetto JSON che ha tre proprietà: `id`, `desc` e `quant`. La proprietà `id` è di tipo intero e rappresenta l'identificatore dell'oggetto `Item`. La proprietà `desc` è una stringa che rappresenta la descrizione dell'oggetto `Item` ed è *obbligatoria* per la creazione di un nuovo oggetto. La proprietà `quant` è una stringa che rappresenta la quantità dell'oggetto `Item` ed è *facoltativa*. La proprietà `id` ha un valore di default di -1, che viene utilizzato quando un nuovo oggetto `Item` viene creato. La proprietà `quant` ha un valore di default vuoto "".

L'oggetto `Item` deve essere incluso come corpo della richiesta (ad esempio fornito come parametro chiamato `json`) nelle operazioni `POST` e `PUT`. Nell'operazione `GET`, l'oggetto `Item` è restituito se l'`ID` dell'oggetto esiste.