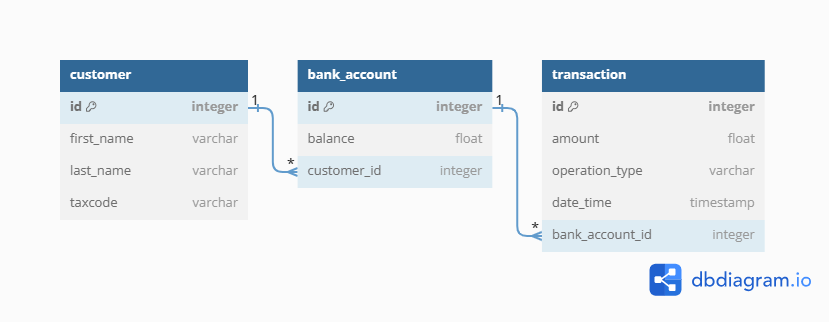
ATE Applicazione conto corrente



# Applicazione conto corrente

Scopo dell’applicazione è di esporre alcuni servizi REST che permettono all’utente, già loggato, di accedere a un proprio conto corrente ed effettuare operazioni di prelievo, versamento, ottenere il saldo disponibile e la lista degli ultimi 5 movimenti.

## Servizi REST esposti

**GET** /customers/bank-accounts/{customerId}

Ottiene la lista dei numeri di conto corrente associati al cliente.  
Ritorna 404 not found se non trova alcun record nella suddetta tabella.

Parametri:

|  |  |  |
| --- | --- | --- |
| pathparam | customerId | (int) codice cliente |

Risposta:

|  |  |  |
| --- | --- | --- |
| 200 | application/json | [  int  ] |
| 404 |  |  |

**POST** /bank-accounts/credit

Effettua un versamento sul conto corrente della somma indicata.

Parametri:

|  |  |  |
| --- | --- | --- |
| Json | {  "amount": float,  "bankAccountId": int  } | Amount: (int) somma da versare  BankAccountId: (int) numero conto corrente |

Risposta:

|  |  |  |
| --- | --- | --- |
| 200 |  |  |

**POST** /bank-accounts/debit

Effettua un prelievo sul conto corrente della somma indicata. Se la somma da prelevare è maggiore del saldo, o il conto è in rosso, viene restituito un errore.

Parametri:

|  |  |  |
| --- | --- | --- |
| Json | {  "amount": float,  "bankAccountId": int  } | Amount: (int) somma da prelevare  BankAccountId: (int) numero conto corrente |

Risposta:

|  |  |  |
| --- | --- | --- |
| 200 |  |  |
| 500 | string | "Impossibile prelevare la somma richiesta in quanto supera il saldo disponibile" |

**GET** /bank-accounts/balance

Restituisce il saldo disponibile sul conto corrente indicato.

Parametri:

|  |  |  |
| --- | --- | --- |
| queryparam |  | BankAccountId: (int) numero conto corrente |

Risposta:

|  |  |  |
| --- | --- | --- |
| 200 | string |  |

**GET** /transactions/history/{accountId}

Ottiene la lista degli ultimi 5 movimenti effettuati sul conto corrente indicato.

Parametri:

|  |  |  |
| --- | --- | --- |
| pathparam | accountId | (int) numero conto corrente |

Risposta:

|  |  |  |
| --- | --- | --- |
| 200 | application/json | [  {  "id": int,  "amount": float,  "operationType": string,  "dateTime": dateTime,  "bankAccountId": int  "customerId": int  },  } |

## Tecnologie utilizzate

L’applicazione è stata sviluppata con framework Spring Boot, basata su Java 8.   
Per la connessione al database è stato utilizzato il driver postgres-jdbc.   
È stato utilizzato come Design Pattern JPA, la cui implementazione è stata realizzata con il framework Hibernate.   
Per l’esposizione dei serizi REST API è stato utilizzato il framework Spring Web.   
Infine, per la funzionalità di logging, è stato utilizzato Log4j, v1.2.17.

## Dati e informazioni

L’applicazione lavora su 3 tabelle:

* 1 TABELLA CUSTOMER Tabella che contiene l’anagrafica dei client. Per ogni cliente sono elencati: nome, cognome e codice fiscale.
* 2 TABELLA BANK\_ACCOUNT Tabella che contiene le informazioni sui conti corrente. Contiene le informazioni sul saldo disponibile. Il campo CUSTOMER\_ID lega in relazione 1 a n la tabella CUSTOMER a quella BANK\_ACCOUNT.
* 3 TABELLA TRANSACTION Tabella che contiene la lista di tutti i movimenti effettuati. Per ogni movimento sono elencati tipo di transazione, somma versata/prelevata e data e ora della transazione. Il campo BANK\_ACCOUNT \_ID lega in relazione 1 a n la tabella BANK\_ACCOUNT a quella TRANSACTION.