

# Istruzioni progetto SD

## Autori:

Gabriele Sormani 869217

Emanuele Masiero 872695

## Introduzione:

Partendo da un computer con solo un sistema operativo installato:

Sistema Operativo utilizzato: Windows 10.

Architettura CPU: x86

Per eseguire il progetto è necessario installare Java sul proprio PC.

Nel progetto sono state usate le seguenti dipendenze:

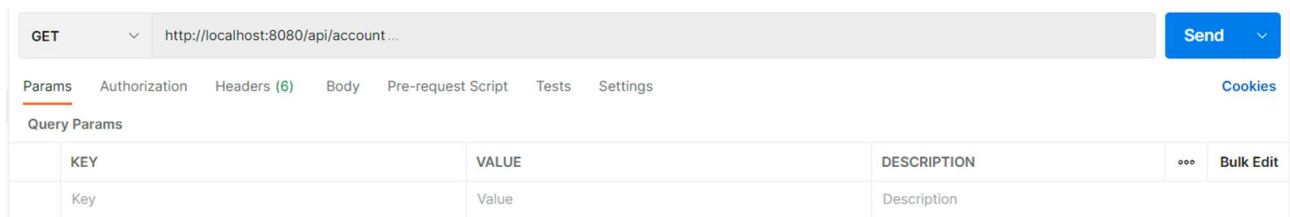
- SQLite per creare il database è possibile posizionare l'eseguibile di SQLite all'interno della cartella del progetto per manipolare manualmente (da linea di comando) il database.
- Libreria jdbc per interfacciarsi con il database, la libreria è stata inclusa nel file pom.xml
- Sono state utilizzate le dipendenze Spring Web anche queste incluse nel file pom.xml
- È stato usato il framework Spring MVC per realizzare gli endpoint
- All'interno dei file html sono stati importati i css forniti da bootstrap

## Istruzione per testare:

Per testare gli endpoint abbiamo utilizzato Postman utile per specificare le varie richieste con i loro parametri o body. Solamente degli endpoint `/api/transfer` e `/api/account/{account id}` (GET) sarà disponibile l'interfaccia grafica per i test (come da richiesta).

### `/api/account`

GET:



The screenshot shows a Postman interface for a GET request to `http://localhost:8080/api/account...`. The 'Params' tab is selected, showing a table for Query Params with columns KEY, VALUE, and DESCRIPTION. The table has one row with 'Key' and 'Value'.

KEY	VALUE	DESCRIPTION
Key	Value	Description

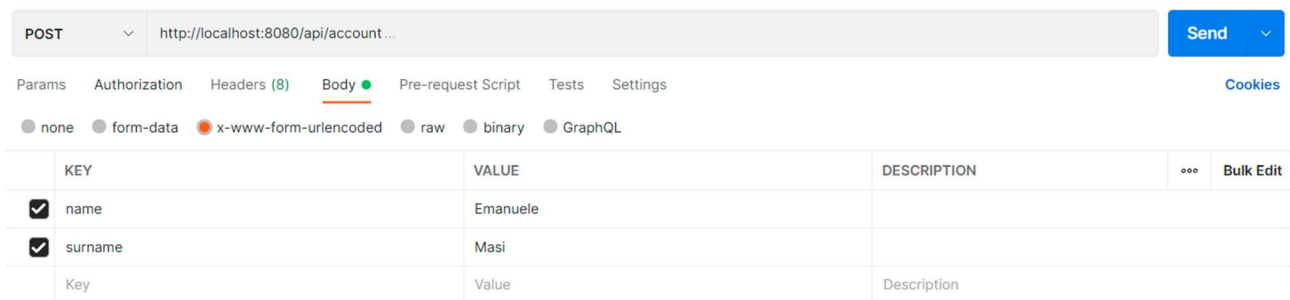
Ritorna l'elenco di tutti gli account con relative transazioni in formato JSON

URL: <http://localhost:8080/api/account>

Metodo: GET

Parametri: nessuno

POST:



The screenshot shows a Postman interface for a POST request to `http://localhost:8080/api/account...`. The 'Body' tab is selected, showing a table for form data with columns KEY, VALUE, and DESCRIPTION. The table has two rows: 'name' with value 'Emanuele' and 'surname' with value 'Masi'.

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> name	Emanuele	
<input checked="" type="checkbox"/> surname	Masi	
Key	Value	Description

Ritorna l'id dell'account creato in formato JSON

URL: <http://localhost:8080/api/account>

Metodo: POST

Parametri: all'interno del body della POST in formato x-www-form-urlencoded vanno inserite le coppie name:valore, surname:valore

## DELETE:

The screenshot shows a REST client interface with the following details:

- Method:** DELETE
- URL:** `http://localhost:8080/api/account?ID=c952d2e49cdb24067c95`
- Params:** Tab selected, showing Query Params.
- Query Params Table:**

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	ID	c952d2e49cdb24067c95			
	Key	Value	Description		

Elimina un account già creato

URL: <http://localhost:8080/api/account>

Metodo: DELETE

Parametri: va passato come parametro nell'URL l'id dell'account da eliminare

Nota: non elimina le transazioni di quell'account per tenere traccia della contabilità di tutte le transazioni anche di vecchi account eliminati

</api/account/{id account}>

## GET:

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** `http://localhost:8080/api/account/50c41a391cbe2ace67ae`
- Params:** Tab selected, showing Query Params.
- Query Params Table:**

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
	Key	Value	Description		

Ritorna i dati dell'account con l'id specificato come parametro dell'URL, in formato JSON e aggiunge un header con chiave X-Sistema-Bancario con valore nome;cognome

URL: <http://localhost:8080/api/account/{idAccount}>

Metodo: GET

Parametri: nessuno

## POST:

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** `http://localhost:8080/api/account/3279b21e3af23f8c82ca`
- Body:** Tab selected, showing raw body.
- Body Content:** `amount: 10`

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	amount	10			

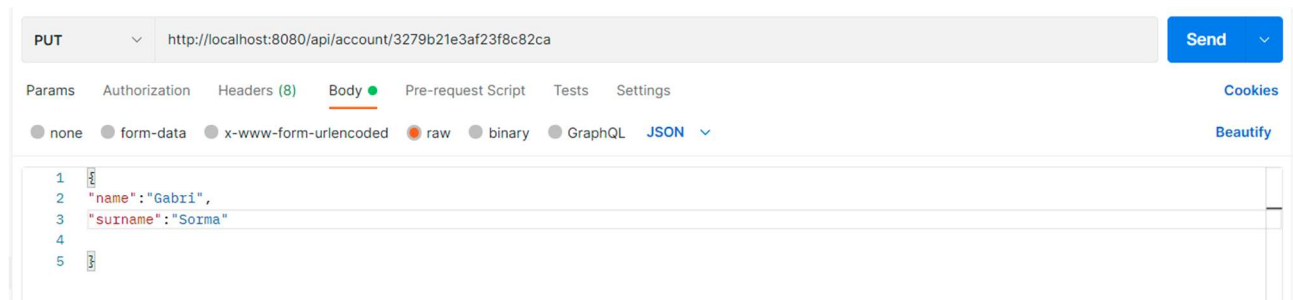
Effettua un versamento o un prelievo verso l'account specificato

URL: <http://localhost:8080/api/account/{idAccount}>

Metodo: POST

Parametri: nel body della POST in formato x-www-form-urlencoded va specificato la coppia amount:valore

PUT:



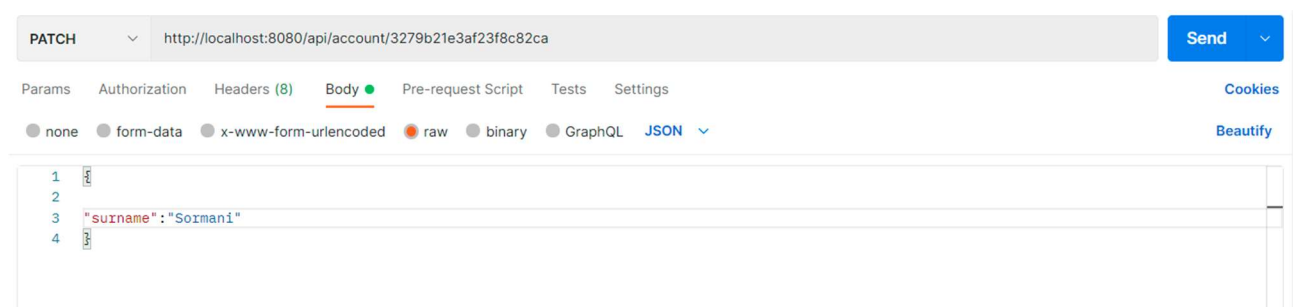
Modifica nome e cognome di un account

URL: <http://localhost:8080/api/account/{idAccount}>

Metodo: PUT

Parametri: nel body in formato JSON vanno passati il nuovo nome e cognome come in foto (chiavi name o surname)

PATCH:



Modifica il nome o il cognome di un account

URL: <http://localhost:8080/api/account/{idAccount}>

Metodo: PATCH

Parametri: nel body in formato JSON va passato il nuovo cognome o il nuovo nome come in foto (chiavi name o surname)

## HEAD:

The screenshot shows a REST client interface with the following details:

- Method:** HEAD
- URL:** `http://localhost:8080/api/account/3279b21e3af23f8c82ca`
- Send Button:** A blue button labeled "Send".
- Tabs:** Params, Authorization, Headers (6), Body, Pre-request Script, Tests, Settings. The "Params" tab is selected.
- Query Params Table:**

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Aggiunge un header nella risposta con chiave X-Sistema-Bancario con valore nome;cognome

URL: <http://localhost:8080/api/account/{idAccount}>

Metodo: HEAD

Parametri: nessuno

[/api/transfer](#)

## POST:

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** `http://localhost:8080/api/transfer`
- Send Button:** A blue button labeled "Send".
- Body Type:** x-www-form-urlencoded (selected)
- Body Table:**

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> from	3279b21e3af23f8c82ca			
<input checked="" type="checkbox"/> to	50c41a391cbe2ace67ae			
<input checked="" type="checkbox"/> amount	50			

Crea una transazione da un account a un altro oppure a sé stesso

URL: `http://localhost:8080/api/transfer`

Metodo: POST

Parametri: nel body della post in formato x-www-form-urlencoded vanno passate le coppie  
from:valore, to:valore, amount:valore

/api/divert

POST:

POST ⌵ http://localhost:8080/api/divert Send ⌵

Params Authorization Headers (8) **Body** ● Pre-request Script Tests Settings Cookies

☐ none ☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	transfertID	7ac7c02c-8c6d-4cf5-befd-141013f4d205			
	Key	Value	Description		

Inverte una transazione esistente creando una nuova transazione opposta

URL: <http://localhost:8080/api/divert>

Metodo: POST

Parametri: nel body in formato x-www-form-urlencoded va specificata la coppia transferID:valore

## Interfaccia Grafica

Gli endpoint /api/transfer e /api/account/{account id} (GET) dispongono di un lato frontend realizzato tramite pagine html e supportato da javascript.

La pagina principale raggiungibile da URL <http://localhost:8080/> contiene la pagina base '/' come da richieste, la pagina '/transfer' è raggiungibile tramite <http://localhost:8080/transfer>, all'interno delle pagine è presente un menu di navigazione che permette di spostarsi tra le due pagine senza riscrivere l'URL ogni volta.

Per il testing di queste pagine è sufficiente attenersi alle indicazioni fornite all'interno dei campi da completare della pagina web e provare a visualizzare risultati o eventuali errori.