

# ANALISI API REST PROGETTO LABORATORIO

## TABELLE

```
CREATE TABLE utenti(  
  idUtente int not null PRIMARY KEY auto_increment,  
  username varchar(30)not null unique,  
  nome varchar(30)not null,  
  cognome varchar(30)not null,  
  password varchar(32)not null,  
  codiceFiscale varchar(16)not null,  
  regione varchar(30)not null,  
  via varchar(30)not null,  
  nCivico varchar(5)not null )
```

```
CREATE TABLE richieste(  
  idRichiesta int not null PRIMARY KEY auto_increment,  
  rifUtente int not null,  
  oraInizioConsegna time not null,  
  oraFineConsegna time not null,  
  durataRichiesta time not null )
```

```
CREATE TABLE liste(  
  idLista int not null PRIMARY KEY auto_increment,  
  rifRichiesta int not null,  
  rifProdotto int not null,  
  Q.Tà int(10)not null )
```

ALTER TABLE richieste

ADD CONSTRAINT associazione1

FOREIGN KEY (rifUtente) REFERENCES utenti(idUtente)

ALTER TABLE liste

ADD CONSTRAINT associazione3

FOREIGN KEY (rifRichiesta) REFERENCES richieste(IdRichiesta)

ALTER TABLE liste

ADD CONSTRAINT associazione3

FOREIGN KEY (rifProdotto) REFERENCES prodotti(IdProdotto)

1)

GET /gestioneSpesa/ richiesteCompletate /{idUtente}/				
Summary				
Elenco delle richieste completate				
Description				
Permette di avere l'elenco delle varie richieste soddisfatte/completate di un determinato utente				
Parameters				
Name	Located In	Description	Required	Schema
idUtente	Path	ID dell'utente del quale si vuole ottenere la lista di richieste completate	Yes	Xs:int
Responses				
Code	Description	Schema		
200	Elenco delle richieste completate da un determinato utente	<xs:complexType name="return"> <xs:sequence> <xs:element name="richiesteCompletate" minOccurs="0" maxOccurs="unbounded"> <xs:complexType> <xs:sequence> <xs:element name="idRichiestaCompletata" type="xs:string"/> </xs:sequence> </xs:complexType> </xs:sequence> </xs:sequence> </xs:complexType>		

		<pre> &lt;xs:element name="rifRichiesta" type="xs:string"/&gt; &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; </pre>
<b>403</b>	Non autorizzato	
<b>406</b>	Parametri di input non validi	
<b>415</b>	valore del content-type non è quello atteso	
<b>Default</b>	Errore non previsto	

### Esempi output proposto all'utente dal consumer:

//Select non utilizzata per proporre un output all'utente ma volta ad ottenere l'id identificativo della/delle richieste completate dall'utente

### Esempi query SQL:

```

SELECT idRichiestaCompletata, rifRichiesta
WHERE rifUtente=[idUtente]

```

### Esempi create table SQL:

```

CREATE TABLE richiesteCompletate (
    CREATE TABLE richiesteCompletate (
        idRisposta int not null PRIMARY KEY auto_increment,
        rifUtente int not null,
        rifRichiesta int not null )
)

```

```

ALTER TABLE risposte
ADD CONSTRAINT associazione2
FOREIGN KEY (rifUtente) REFERENCES utenti(idUtente)

```

```

ALTER TABLE risposte
ADD CONSTRAINT associazione3
FOREIGN KEY (rifRichiesta) REFERENCES richieste(IdRichiesta)

```

2)

GET /gestioneSpesa/getProdotto/				
<b>Summary</b>				
Elenco dei prodotti				
<b>Description</b>				
Permette di ottenere un elenco di prodotti in base a determinati parametri				
<b>Parameters</b>				
Name	Located In	Description	Required	Schema
Genere	Query	Genere del prodotto	No	Xs:string
Etichetta	Query	Etichetta del prodotto	No	Xs:string
Costo	Query	Costo del prodotto	No	Xs:double
Nome	Query	Nome del prodotto	No	Xs:string
Marca	Query	Marca del prodotto	No	Xs:string
Descrizione	Query	Descrizione del prodotto	No	Xs:string
<b>Responses</b>				
Code	Description	Schema		
200	Elenco dei prodotti	<pre> &lt;xs:complexType name="return"&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="prodotto" minOccurs="0" maxOccurs="unbounded"&gt;       &lt;xs:complexType&gt;         &lt;xs:sequence&gt;           &lt;xs:element name="idProdotto" type="xs:int"/&gt;           &lt;xs:element name="Genere" type="xs:string"/&gt;           &lt;xs:element name="Etichetta" type="xs:string"/&gt;           &lt;xs:element name="Costo" type="xs:double"/&gt;           &lt;xs:element name="Nome" type="xs:string"/&gt;           &lt;xs:element name="Marca" type="xs:string"/&gt;           &lt;xs:element name="Descrizione" type="xs:string"/&gt;         &lt;/xs:sequence&gt;       &lt;/xs:complexType&gt;     &lt;/xs:element&gt;   &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; </pre>		
403	Non autorizzato			
406	Parametri di input non validi			
415	valore del content-type non è quello atteso			

Default	Errore non previsto				
Esempi output proposto all'utente dal consumer:					
Genere	Etichetta	Costo	Nome	Marca	Descrizione
Alimentare	Aaaaa	2,50	Pizza margherita	Bbbb	Una pizza margherita senza glutine
Esempi query SQL:					
SELECT idProdotto,Genere,Etichetta,Costo,Nome,Marca,Descrizione FROM prodotti					
Esempi create table SQL:					
CREATE TABLE prodotti( idProdotto int not null PRIMARY KEY auto_increment, genere varchar(30) not null, etichetta longtext not null, costo double(6,2)not null, nome varchar(50)not null, marca varchar(50)not null, descrizione longtext not null )					

3)

<b>POST /gestioneSpesa/postProdotto/</b>				
<b>Summary</b>				
Inserimento prodotto				
<b>Description</b>				
Permette di aggiungere un prodotto al database				
<b>Parameters</b>				
<b>Name</b>	<b>Located In</b>	<b>Description</b>	<b>Required</b>	<b>Schema</b>
Genere	Body	Genere del prodotto	Yes	Xs:string
Etichetta	Body	Etichetta del prodotto	Yes	Xs:string
Costo	Body	Costo del prodotto	Yes	Xs:double
Nome	Body	Nome del prodotto	Yes	Xs:string
Marca	Body	Marca del prodotto	Yes	Xs:string
Descrizione	Body	Descrizione del prodotto	Yes	Xs:string
<b>Responses</b>				
<b>Code</b>	<b>Description</b>	<b>Schema</b>		

<b>200</b>	Operazione eseguita correttamente	<xs:element name="return" type="xs:boolean">
<b>403</b>	Non autorizzato	
<b>406</b>	Parametri di input non validi	
<b>415</b>	valore del content-type non è quello atteso	
<b>Default</b>	Errore non previsto	
<b>Esempi output proposto all'utente dal consumer:</b>		
<ul style="list-style-type: none"> <li>➤ "Operazione andata a buon fine"</li> <li>➤ "Errore, operazione non completata"</li> </ul>		
<b>Esempi query SQL:</b>		
INSERT INTO prodotto (genere, etichetta, costo, nome, marca, descrizione) VALUES (\$_POST[genereProdotto], \$_POST[etichettaProdotto], \$_POST[costoProdotto], \$_POST[nomeProdotto], \$_POST[marcaProdotto], \$_POST[descrizioneProdotto])		
<b>Esempi create table SQL:</b>		
CREATE TABLE prodotti( idProdotto int not null PRIMARY KEY auto_increment, genere varchar(30) not null, etichetta longtext not null, costo double(6,2)not null, nome varchar(50)not null, marca varchar(50)not null, descrizione longtext not null ) )		

4)

PUT /gestioneSpesa/putProdotto/				
<b>Summary</b>				
Modifica un prodotto				
<b>Description</b>				
Permette la modifica di un prodotto specificando l'id del prodotto da modificare				
<b>Parameters</b>				
<b>Name</b>	<b>Located In</b>	<b>Description</b>	<b>Required</b>	<b>Schema</b>
idProdotto	Body	ID del prodotto	Yes	Xs:int
Genere	Body	Genere del prodotto	Yes	Xs:string

Etichetta	Body	Etichetta del prodotto	Yes	Xs:string
Costo	Body	Costo del prodotto	Yes	Xs:double
Nome	Body	Nome del prodotto	Yes	Xs:string
Marca	Body	Marca del prodotto	Yes	Xs:string
Descrizione	Body	Descrizione del prodotto	Yes	Xs:string

## Responses

Code	Description	Schema
<b>200</b>	Operazione eseguita correttamente	<xs:element name="return" type="xs:boolean">
<b>403</b>	Non autorizzato	
<b>406</b>	Parametri di input non validi	
<b>415</b>	valore del content-type non è quello atteso	
<b>Default</b>	Errore non previsto	

Esempi output proposto all'utente dal consumer:

- "Operazione andata a buon fine"
- "Errore, operazione non completata"

## Esempi query SQL:

```
UPDATE prodotto SET genere=$_POST[genereProdotto], etichetta=$_POST[etichettaProdotto],
costo=$_POST[costoProdotto], nome=$_POST[nomeProdotto], marca=$_POST[marcaProdotto],
descrizione=$_POST[descrizioneProdotto] WHERE idProdotto=$_POST[idProdotto]
```

## Esempi create table SQL:

```
CREATE TABLE prodotti(
    idProdotto int not null PRIMARY KEY auto_increment,
    genere varchar(30) not null,
    etichetta longtext not null,
    costo double(6,2)not null,
    nome varchar(50)not null,
    marca varchar(50)not null,
    descrizione longtext not null )
)
```

5)

DELETE /gestioneSpesa/deleteProdotto/

### Summary

Eliminazione di un prodotto

### Description

Permette di eliminare un dato prodotto

### Parameters

Name	Located In	Description	Required	Schema
idProdotto	Query	idProdotto	Yes	Xs:int

### Responses

Code	Description	Schema
200	Operazione eseguita correttamente	<xs:element name="return" type="xs:boolean">
403	Non autorizzato	
406	Parametri di input non validi	
415	valore del content-type non è quello atteso	
Default	Errore non previsto	

### Esempi output proposto all'utente dal consumer:

- "Operazione andata a buon fine"
- "Errore, operazione non completata"

### Esempi query SQL:

```
DELETE FROM prodotto WHERE idProdotto=[idProdotto]
```

### Esempi create table SQL:

```
CREATE TABLE prodotti(  
    idProdotto int not null PRIMARY KEY auto_increment,  
    genere varchar(30) not null,  
    etichetta longtext not null,  
    costo double(6,2)not null,  
    nome varchar(50)not null,  
    marca varchar(50)not null,  
    descrizione longtext not null )  
)
```



