

# 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 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 )
```

```
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 risposte(  
  idRisposta int not null PRIMARY KEY auto_increment,  
  rifRichiesta int not null,  
  risposta longtext not null )
```

```
idRisposta int not null PRIMARY KEY auto_increment,  
rifUtente int not null,  
rifRichiesta int 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 risposte  
ADD CONSTRAINT associazione2  
FOREIGN KEY (rifUtente) REFERENCES utenti(idUtente)
```

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

```
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)

SELECT idUtente, Username, oraInizioConsegna, oraFineConsegna, durataRichiesta FROM richiesteCompletate WHERE idUtente=[idUtente]

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	↔int
Username	Query	Username dell'utente del quale si vuole ottenere la lista di richieste completate	No	↔string
oraInizioConsegna	Query	Ora di inizio della disponibilità a ricevere la consegna della spesa	No	↔time
oraFineConsegna	Query	Ora di fine della disponibilità di ricevere la consegna della spesa	No	↔time
durataRichiesta	Query	Durata della richiesta	No	↔time
Responses				
Code	Description	Schema		
200	Elenco delle richieste completate da un determinato utente	<pre> &lt;xs:complexType name="return"&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="richieste" minOccurs="0" maxOccurs="unbounded"&gt;       &lt;xs:complexType&gt;         &lt;xs:sequence&gt;           &lt;xs:element name="utente"&gt;             &lt;xs:complexType&gt;               &lt;xs:sequence&gt;                 &lt;xs:element name="idUtente" type="xs:int"/&gt;                 &lt;xs:element name="Username" type="xs:string"/&gt;                 &lt;xs:element name="nome" type="xs:string"/&gt;                 &lt;xs:element name="cognome" type="xs:string"/&gt;                 &lt;xs:element name="codFiscale" type="xs:string"/&gt;                 &lt;xs:element name="regione" type="xs:string"/&gt;                 &lt;xs:element name="via" type="xs:string"/&gt;                 &lt;xs:element name="nCivico" type="xs:int"/&gt;               &lt;/xs:sequence&gt;             &lt;/xs:complexType&gt;           &lt;/xs:element&gt;           &lt;xs:element name=" oraInizioConsegna" 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>		

		<pre> &lt;xs:element name=" oraFineConsegna" type="xs:string"/&gt; &lt;xs:element name=" durataRichiesta" 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	

2)

SELECT \* FROM prodotto

GET /gestioneSpesa/prodotto/				
Summary				
Elenco dei prodotti				
Description				
Permette di ottenere un elenco di prodotti in base a determinati parametri				
Parameters				
Name	Located In	Description	Required	Schema
Genere	Query	Genere del prodotto	No	⇔ string
Etichetta	Query	Etichetta del prodotto	No	⇔string
Costo	Query	Costo del prodotto	No	⇔ double
Nome	Query	Nome del prodotto	No	⇔string
Marca	Query	Marca del prodotto	No	⇔ string
Descrizione	Query	Descrizione del prodotto	No	⇔ string
Responses				
Code	Description	Schema		
<b>200</b>	Elenco dei prodotti	<pre> &lt;xs:complexType name="return"&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="prodotto" minOccurs="0" maxOccurs="unbounded"&gt; </pre>		

		<pre> &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>
<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	

### 3)

INSERT INTO prodotto (genere, etichetta, costo, nome, marca, descrizione) VALUES  
(\$\_POST[genereProdotto], \$\_POST[etichettaProdotto], \$\_POST[costoProdotto], \$\_POST[nomeProdotto],  
\$\_POST[marcaProdotto], \$\_POST[descrizioneProdotto])

POST /gestioneSpesa/prodotto/				
Summary				
Inserimento prodotto				
Description				
Permette di aggiungere un prodotto al database				
Parameters				
Name	Located In	Description	Required	Schema
Genere	Body	Genere del prodotto	Yes	⇒ string
Etichetta	Body	Etichetta del prodotto	Yes	⇒ string
Costo	Body	Costo del prodotto	Yes	⇒ double

Nome	Body	Nome del prodotto	Yes	⇒string
Marca	Body	Marca del prodotto	Yes	⇒ string
Descrizione	Body	Descrizione del prodotto	Yes	⇒ 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			

4)

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]

PUT /gestioneSpesa/prodotto/				
<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	⇒ int
Genere	Body	Genere del prodotto	No	⇒ string
Etichetta	Body	Etichetta del prodotto	No	⇒string
Costo	Body	Costo del prodotto	No	⇒ double
Nome	Body	Nome del prodotto	No	⇒string
Marca	Body	Marca del prodotto	No	⇒ string
Descrizione	Body	Descrizione del prodotto	No	⇒ 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	

5)

DELETE FROM prodotto WHERE Genere=\$\_POST[genere] AND Nome=\$\_POST [nome] AND Marca=\$\_POST[marca]

DELETE /gestioneSpesa/prodotto/				
<b>Summary</b>				
Eliminazione di un prodotto				
<b>Description</b>				
Permette di eliminare un dato prodotto				
<b>Parameters</b>				
<b>Name</b>	<b>Located In</b>	<b>Description</b>	<b>Required</b>	<b>Schema</b>
Genere	Query	Genere del prodotto	Yes	⇒ string
Etichetta	Query	Etichetta del prodotto	No	⇒string
Costo	Query	Costo del prodotto	No	⇒ double
Nome	Query	Nome del prodotto	Yes	⇒string
Marca	Query	Marca del prodotto	Yes	⇒ string
Descrizione	Query	Descrizione del prodotto	No	⇒ 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	