

# Documentazione API

## UTENTE

### POST /api/login

### Body della richiesta:

```
JSON
{
  "username": "string (obbligatorio)",
  "password": "string (obbligatorio)"
}
```

### Esempio richiesta API.js:

```
javascript
fetch('/api/login', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json',
  },
  body: JSON.stringify({
    username: 'your_username',
    password: 'your_password',
  }),
})
.then(response => response.json())
.then(data => console.log(data))
.catch(error => console.error('Error:', error));
```

### Status code 200:

```
JSON
{
  "success": true,
  "user": {
    "username": "utente_esistente",
    "email": "utente@example.com",
```

```
"indirizzo": "Via Milano 45",  
"role": "admin"  
}  
}
```

### Status code 401 (Utente non trovato):

```
JSON  
{  
  "success": false,  
  "message": "Utente non trovato"  
}
```

### Status code 401 (Password errata):

```
JSON  
{  
  "success": false,  
  "message": "Password errata"  
}
```

### Status code 400:

```
JSON  
{  
  "success": false,  
  "message": "Username, password ed email sono obbligatori"  
}
```

### Status code 500:

```
JSON  
{  
  "success": false,  
  "message": "Errore interno del server"  
}
```

## POST /api/register

### Body della richiesta:

```
JSON  
{  
  "username": "string (obbligatorio)",  
  "password": "string (obbligatorio)",  
  "email": "string (obbligatorio)",  
}
```

```
"indirizzo": "string (opzionale)",  
"ruolo": "string (opzionale, default: 'user')"  
}
```

### Esempio richiesta API.js:

```
javascript  
fetch("/api/register", {  
  method: "POST",  
  headers: {  
    "Content-Type": "application/json",  
  },  
  body: JSON.stringify({  
    username: "new_user",  
    password: "secure_password",  
    email: "new.user@example.com",  
    indirizzo: "Via Example 123",  
    ruolo: "customer",  
  }),  
})  
.then(response => response.json())  
.then(data => console.log(data))  
.catch(error => console.error("Error:", error));
```

### Status code 201:

```
JSON  
{  
  "success": true,  
  "user": {  
    "username": "nuovo_utente",  
    "email": "nuovo.utente@example.com",  
    "indirizzo": "Via Indirizzo 123",  
    "role": "customer"  
  },  
  "message": "Registrazione avvenuta con successo"  
}
```

### Status code 409:

```
JSON  
{  
  "success": false,
```

```
"message": "Un utente con questo username è già esistente"
}
```

### Status code 400:

```
JSON
{
  "success": false,
  "message": "Username, password ed email sono obbligatori"
}
```

### Status code 500:

```
JSON
{
  "success": false,
  "message": "Errore interno del server"
}
```

## GET /api/user/:username

### Esempio:

```
/api/user/john_doe?username=john_doe
```

### Parametri query:

```
JSON
{
  "username": "string (obbligatorio)"
}
```

### Esempio richiesta API.js:

```
javascript
fetch("/api/user/john_doe?username=john_doe")
  .then(response => response.json())
  .then(data => console.log(data))
  .catch(error => console.error("Error:", error));
```

### Status code 200:

```
JSON
{
  "success": true,
  "user": {
    "username": "john_doe",
```

```
"email": "john@example.com",  
"telefono": "+39 123 456 789",  
"indirizzo": "Via Roma 123",  
"role": "customer"  
}  
}
```

### Status code 401:

```
JSON  
{  
  "success": false,  
  "message": "Utente non trovato"  
}
```

### Status code 400:

```
JSON  
{  
  "success": false,  
  "message": "Sono presenti caratteri speciali"  
}
```

### Status code 500:

```
JSON  
{  
  "success": false,  
  "message": "Errore interno del server"  
}
```

## POST /api/user/update

### Body della richiesta:

```
JSON  
{  
  "id": "integer (obbligatorio)",  
  "username": "string (opzionale)",  
  "password": "string (opzionale)",  
  "telefono": "string (opzionale)",  
  "indirizzo": "string (opzionale)"  
}
```

### Esempio richiesta API.js:

```
javascript
fetch("/api/user/update", {
  method: "POST",
  headers: {
    "Content-Type": "application/json",
  },
  body: JSON.stringify({
    id: 123,
    username: "updated_user",
    telefono: "+39 987 654 321",
    indirizzo: "Via Nuova 456",
  }),
})
.then(response => response.json())
.then(data => console.log(data))
.catch(error => console.error("Error:", error));
```

### Status code 200:

```
JSON
{
  "success": true,
  "user": {
    "id": 123,
    "username": "utente_aggiornato",
    "email": "utente@example.com",
    "telefono": "+39 987 654 321",
    "indirizzo": "Via Nuova 456"
  }
}
```

### Status code 404:

```
JSON
{
  "success": false,
  "message": "Utente non trovato"
}
```

### Status code 500:

```
JSON
{
```

```
"success": false,  
"message": "Errore interno del server"  
}
```

# PRODOTTO

## GET /api/products

### Esempio:

```
/api/products?
```

```
categoria=abbigliamento&categoria=gioielli&prezzo_min=50&disponibilita=100&limit=10
```

### Parametri url:

JSON

```
{  
  "categoria": "array (opzionale, es. [\ 'arredamento\ ', \ 'gioielli\ '])",  
  "prezzo_min": "number (opzionale, es. 50, 100.50)",  
  "prezzo_max": "number (opzionale, es. 200, 75.99)",  
  "disponibilita": "integer (opzionale, es. 10, 50 - mostra prodotti con almeno X unità  
disponibili)",  
  "limit": "integer (opzionale, es. 10, 20 - limita il numero di risultati)"  
}
```

### Esempio richiesta API.js:

```
javascript
```

```
fetch("/api/products?
```

```
categoria=abbigliamento&prezzo_min=50&disponibilita=100&limit=10")
```

```
.then(response => response.json())
```

```
.then(data => console.log(data))
```

```
.catch(error => console.error("Error:", error));
```

### Status code 200:

JSON

```
{  
  "success": true,  
  "products": [  
    { "id": 101, "nome": "Prodotto A", "prezzo": 799.99 },  
    { "id": 102, "nome": "Prodotto B", "prezzo": 89.00 },  
    { "id": 103, "nome": "Prodotto C", "prezzo": 249.50 }
```

```
]
}
```

### Status code 404:

```
JSON
{
  "success": false,
  "message": "Prodotti non trovati per i filtri specificati"
}
```

### Status code 500:

```
JSON
{
  "success": false,
  "message": "Errore interno del server"
}
```

## GET /api/products/:id

### Esempio:

```
/api/products/123
```

### Esempio richiesta API.js:

```
javascript
fetch("/api/products/123")
  .then(response => response.json())
  .then(data => console.log(data))
  .catch(error => console.error("Error:", error));
```

### Status code 200:

```
JSON
{
  "success": true,
  "product": {
    "id": 123,
    "nome": "Maglietta in Cotone",
    "prezzo": 25.99,
    "descrizione": "Una maglietta comoda e traspirante, 100% cotone biologico.",
    "disponibilita": 250,
    "categoria": "abbigliamento"
```



```
}
```

```
}
```

### Status code 404:

JSON

```
{
```

```
"success": false,
```

```
"message": "Prodotto non trovato"
```

```
}
```

### Status code 400:

JSON

```
{
```

```
"success": false,
```

```
"message": "ID prodotto non valido"
```

```
}
```

### Status code 500:

JSON

```
{
```

```
"success": false,
```

```
"message": "Errore interno del server"
```

```
}
```

## POST /api/product/add

### Body della richiesta:

JSON

```
{
```

```
"nome": "string (obbligatorio)",
```

```
"categoria": "string (obbligatorio)",
```

```
"descrizione": "string (obbligatorio)",
```

```
"prezzo": "number (obbligatorio)",
```

```
"disponibilita": "integer (obbligatorio)",
```

```
"idVenditore": "integer (obbligatorio)"
```

```
}
```

### Esempio richiesta API.js:

javascript

```
fetch("/api/product/add", {
```

```
  method: "POST",
```

```
headers: {  
  "Content-Type": "application/json",  
},  
body: JSON.stringify({  
  nome: "Nuovo Prodotto",  
  categoria: "elettronica",  
  descrizione: "Descrizione del nuovo prodotto",  
  prezzo: 199.99,  
  disponibilita: 50,  
  idVenditore: 789,  
}),  
})  
.then(response => response.json())  
.then(data => console.log(data))  
.catch(error => console.error("Error:", error));
```

### Status code 200:

```
JSON  
{  
  "success": true,  
  "product": {  
    "id": 456,  
    "nome": "Nuovo Prodotto",  
    "categoria": "elettronica",  
    "descrizione": "Descrizione del nuovo prodotto",  
    "prezzo": 199.99,  
    "disponibilita": 50,  
    "idVenditore": 789  
  }  
}
```

### Status code 409:

```
JSON  
{  
  "success": false,  
  "message": "Prodotto già esistente"  
}
```

### Status code 500:

```
JSON
```

```
{  
  "success": false,  
  "message": "Errore interno del server"  
}
```

## POST /api/product/delete

### Body della richiesta:

```
JSON  
{  
  "id": "integer (obbligatorio)"  
}
```

### Esempio richiesta API.js:

```
javascript  
fetch("/api/product/delete", {  
  method: "DELETE",  
  headers: {  
    "Content-Type": "application/json",  
  },  
  body: JSON.stringify({  
    id: 123,  
  }),  
})  
.then(response => response.json())  
.then(data => console.log(data))  
.catch(error => console.error("Error:", error));
```

### Status code 200:

```
JSON  
{  
  "success": true  
}
```

### Status code 404:

```
JSON  
{  
  "success": false,  
  "message": "Prodotto non trovato"  
}
```

### Status code 500:

```
JSON
{
  "success": false,
  "message": "Errore interno del server"
}
```

## POST /api/product/update

### Body della richiesta:

```
JSON
{
  "id": "integer (obbligatorio)",
  "descrizione": "string (opzionale)",
  "prezzo": "number (opzionale)",
  "disponibilita": "integer (opzionale)"
}
```

### Esempio richiesta API.js:

```
javascript
fetch("/api/product/update", {
  method: "POST",
  headers: {
    "Content-Type": "application/json",
  },
  body: JSON.stringify({
    id: 123,
    descrizione: "Nuova descrizione del prodotto",
    prezzo: 149.99,
    disponibilita: 75,
  }),
})
.then(response => response.json())
.then(data => console.log(data))
.catch(error => console.error("Error:", error));
```

### Status code 200:

```
JSON
{
  "success": true,
```

```
"product": {  
  "id": 123,  
  "nome": "Prodotto Aggiornato",  
  "categoria": "elettronica",  
  "descrizione": "Nuova descrizione",  
  "prezzo": 149.99,  
  "disponibilita": 75  
}
```

### Status code 404:

```
JSON  
{  
  "success": false,  
  "message": "Prodotto non trovato"  
}
```

### Status code 500:

```
JSON  
{  
  "success": false,  
  "message": "Errore interno del server"  
}
```

## IMMAGINI PRODOTTO

### GET /api/product/:id/image

### Esempio:

```
/api/product/123/image?max=10
```

### Parametri query:

```
JSON  
{  
  "max": "integer (opzionale, default: 0 - numero massimo di immagini da restituire)"  
}
```

### Esempio richiesta API.js:

```
javascript
```

```
fetch("/api/product/123/image?max=10")
.then(response => response.json())
.then(data => console.log(data))
.catch(error => console.error("Error:", error));
```

### Status code 200:

```
JSON
{
  "success": true,
  "images": [
    {
      "id": 1,
      "product_id": 123,
      "url": "https://example.com/image1.jpg",
      "alt_text": "Immagine principale del prodotto"
    },
    {
      "id": 2,
      "product_id": 123,
      "url": "https://example.com/image2.jpg",
      "alt_text": "Vista laterale del prodotto"
    }
  ]
}
```

### Status code 404:

```
JSON
{
  "success": false,
  "message": "Prodotto non trovato"
}
```

### Status code 500:

```
JSON
{
  "success": false,
  "message": "Errore interno del server"
}
```

## POST /api/product/:id/image/add

### Body della richiesta:

```
JSON
{
  "images": [
    {
      "url": "string (obbligatorio)",
      "alt_text": "string (opzionale)"
    },
    {
      "url": "string (obbligatorio)",
      "alt_text": "string (opzionale)"
    }
  ]
}
```

### Esempio richiesta API.js:

```
javascript
fetch("/api/product/123/image/add", {
  method: "POST",
  headers: {
    "Content-Type": "application/json",
  },
  body: JSON.stringify({
    images: [
      {
        url: "https://example.com/new_image1.jpg",
        alt_text: "Nuova immagine 1",
      },
      {
        url: "https://example.com/new_image2.jpg",
        alt_text: "Nuova immagine 2",
      },
    ],
  }),
})
.then(response => response.json())
.then(data => console.log(data))
.catch(error => console.error("Error:", error));
```

### Status code 200:

JSON

```
{  
  "success": true,  
  "data": [  
    {  
      "id": 10,  
      "product_id": 123,  
      "url": "https://example.com/new_image1.jpg",  
      "alt_text": "Nuova immagine 1"  
    },  
    {  
      "id": 11,  
      "product_id": 123,  
      "url": "https://example.com/new_image2.jpg",  
      "alt_text": "Nuova immagine 2"  
    }  
  ]  
}
```

### Status code 400:

JSON

```
{  
  "success": false,  
  "message": "Lista di immagini non fornita o vuota nel corpo della richiesta."  
}
```

### Status code 400 (nessuna immagine valida):

JSON

```
{  
  "success": false,  
  "message": "Nessuna immagine valida fornita."  
}
```

### Status code 500:

JSON

```
{  
  "success": false,  
  "message": "Errore interno del server: nessuna riga inserita."  
}
```



# DELETE /api/product/image/:id/delete

### Esempio:

```
DELETE /api/product/image/456/delete
```

### Esempio richiesta API.js:

```
javascript
fetch("/api/product/image/456/delete", {
  method: "DELETE",
  headers: {
    "Content-Type": "application/json",
  },
})
.then(response => response.json())
.then(data => console.log(data))
.catch(error => console.error("Error:", error));
```

### Status code 200:

```
JSON
{
  "success": true
}
```

### Status code 404:

```
JSON
{
  "success": false,
  "message": "Immagine non trovata"
}
```

### Status code 500:

```
JSON
{
  "success": false,
  "message": "Errore interno del server"
}
```

# ORDINI

## GET /api/orders

### Parametri query:

```
JSON
{
  "id": "integer (obbligatorio - ID utente)",
  "ruolo": "string (obbligatorio - ruolo dell'utente: 'customer', 'admin', 'seller')"
```

### Esempio:

```
/api/orders?id=123&ruolo=customer
```

### Esempio richiesta API.js:

```
javascript
fetch("/api/orders?id=123&ruolo=customer")
.then(response => response.json())
.then(data => console.log(data))
.catch(error => console.error("Error:", error));
```

### Status code 200:

```
JSON
{
  "success": true,
  "orders": [
    {
      "id": 1001,
      "user_id": 123,
      "data_ordine": "2024-01-15T10:30:00Z",
      "stato": "confermato",
      "totale": 149.99,
      "prodotti": [
        {
          "id": 456,
          "nome": "Prodotto A",
          "quantita": 2,
          "prezzo_unitario": 74.99
        }
      ]
    }
  ]
}
```

```
]
}
]
}
```

### Status code 404:

```
JSON
{
  "success": false,
  "message": "Ordini non trovati"
}
```

### Status code 500:

```
JSON
{
  "success": false,
  "message": "Errore interno del server"
}
```

# PAGAMENTI

## POST /api/checkout

### Body della richiesta:

```
JSON
{
  "carrello": "array (obbligatorio - array di oggetti prodotto, es. [{ productId: 1,
quantity: 2 }])",
  "cliente_id": "integer (obbligatorio - ID del cliente)"
}
```

### Esempio richiesta API.js:

```
javascript
fetch("/api/checkout", {
  method: "POST",
  headers: {
    "Content-Type": "application/json",
  },
  body: JSON.stringify({
```

```
    carrello: [  
      { productId: 123, quantity: 1 },  
      { productId: 456, quantity: 2 },  
    ],  
    cliente_id: 789,  
  }},  
})  
.then(response => response.json())  
.then(data => console.log(data))  
.catch(error => console.error("Error:", error));
```

### Status code 200:

```
JSON  
{  
  "success": true,  
  "data": {  
    "sessionId": "cs_test_a123b456c789",  
    "url": "https://checkout.stripe.com/pay/cs_test_a123b456c789"  
  }  
}
```

### Status code 500:

```
JSON  
{  
  "success": false,  
  "message": "Errore nel checkout"  
}
```

## GET /api/checkout/status/:sessionId

### Esempio:

```
/api/checkout/status/cs_test_a123b456c789
```

### Esempio richiesta API.js:

```
javascript  
fetch("/api/checkout/status/cs_test_a123b456c789")  
.then(response => response.json())  
.then(data => console.log(data))  
.catch(error => console.error("Error:", error));
```

### Status code 200:

JSON

```
{  
  "success": true,  
  "data": {  
    "id": "cs_test_a123b456c789",  
    "status": "complete",  
    "amount_total": 10000,  
    "currency": "eur"  
  }  
}
```

### Status code 404:

JSON

```
{  
  "success": false,  
  "message": "La sessione di checkount non è stata trovata"  
}
```

### Status code 500:

JSON

```
{  
  "success": false,  
  "message": "Errore nel recupero dello stato della sessione"  
}
```

# PAGAMENTI

## POST /api/checkout

### Body della richiesta:

JSON

```
{  
  "carrello": "array (obbligatorio - array di oggetti prodotto, es. [{ productId: 1,  
quantity: 2 }])",  
  "cliente_id": "integer (obbligatorio - ID del cliente)"  
}
```

### Esempio richiesta API.js:

```
javascript
fetch("/api/checkout", {
  method: "POST",
  headers: {
    "Content-Type": "application/json",
  },
  body: JSON.stringify({
    carrello: [
      { productId: 123, quantity: 1 },
      { productId: 456, quantity: 2 },
    ],
    cliente_id: 789,
  }),
})
.then(response => response.json())
.then(data => console.log(data))
.catch(error => console.error("Error:", error));
```

### Status code 200:

```
JSON
{
  "success": true,
  "data": {
    "sessionId": "cs_test_a123b456c789",
    "url": "https://checkout.stripe.com/pay/cs_test_a123b456c789"
  }
}
```

### Status code 500:

```
JSON
{
  "success": false,
  "message": "Errore nel checkout"
}
```

## GET /api/checkout/status/:sessionId

### Esempio:

```
/api/checkout/status/cs_test_a123b456c789
```

### Esempio richiesta API.js:

```
javascript
fetch("/api/checkout/status/cs_test_a123b456c789")
.then(response => response.json())
.then(data => console.log(data))
.catch(error => console.error("Error:", error));
```

### Status code 200:

```
JSON
{
  "success": true,
  "data": {
    "id": "cs_test_a123b456c789",
    "status": "complete",
    "amount_total": 10000,
    "currency": "eur"
  }
}
```

### Status code 404:

```
JSON
{
  "success": false,
  "message": "La sessione di checkount non è stata trovata"
}
```

### Status code 500:

```
JSON
{
  "success": false,
  "message": "Errore nel recupero dello stato della sessione"
}
```