

Documentazione API Esame Sessione

4

Documentazione Ufficiale per l'utilizzo dell'API

Sommario

Sommario	1
Introduzione	3
Esempio Richiesta	3
Indicazioni Generali	5
Dominio	5
Vincoli	5
Formato	6
Utenti	6
Richieste	7
Autenticazione	8
Registrazione	8
Login	9
Logout	10
Forza Logout	10
Categoria	12
Elenco Categorie	12
Singola Categoria	13
Aggiunta Categoria	14
Modifica Categoria	15
Eliminazione Categoria	16
Film	18
Elenco Film	18
Singolo Film	20
Aggiunta Film	22
Modifica Film	24
Eliminazione Film	26
Recupero Film	27
Eliminazione Definitiva Film	28
Serie TV	30
Elenco Serie	30
Singola Serie	32
Aggiunta Serie	35
Modifica Serie	36
Eliminazione Serie	39
Recupero Serie	39

Eliminazione Definitiva Serie	41
Episodio	42
Elenco Episodi	42
Singolo Episodio	44
Aggiunta Episodio	46
Modifica Episodio	47
Eliminazione Episodio	49
Recupero Episodio	50
Eliminazione Definitiva Serie	51
Utenti	53
Elenco Utenti	53
Singolo Utente	54
Aggiunta Utente	55
Modifica Utente	56
Eliminazione Utente	57
Recupero Utente	58
Eliminazione Definitiva Utente	59
Ban Utente	60
Sospensione Utente	60
Blocco Utente	61
Attivazione Utente	62
Profilo	64
Accesso Profilo	64
Modifica Profilo	65
Eliminazione Profilo	66
Pagamenti	68
Accesso Credito	68
Aggiunta Credito	68
Rimozione Credito	69
Note Finali	71

Introduzione

Questa è la **documentazione ufficiale** della mia **API Codex**, di tipo **RESTful**. Qui è possibile trovare tutte le informazioni riguardo all'**autenticazione del client**, alle **risorse accessibili per ogni tipologia di utente** e le **possibili azioni che possono intraprendere**.

Per **ogni richiesta** verranno elaborati tutti le **informazioni utili** come metodo e ritorno, **elencate e riassunte nel seguente schema**:

Esempio Richiesta

Utenti con Privilegio: categoria_1, categoria_2

Operazione: il **tipo di operazione** che viene eseguita (es: creazione di una risorsa)

Formato Richiesta: il formato del payload della richiesta

Formato Risposta: il formato del ritorno

Descrizione: breve descrizione della richiesta e il suo impiego, più altre peculiarità

URL: `https://esempio.it/api/(versione api)/esempio/{[parametro]}?[chiave1]=[valore1]`

Metodo HTTP: GET

Payload:

```
{  
  
  "[chiave]": (valore)  
  
}
```

Parametri:

`[parametro] (tipo parametro): [descrizione parametro]`

Query String:

`chiave1 (string): [valore della chiave], es. tipo=attivo`

Valori accettati (se limitati): `valore1, valore2, valore3`

Errori Possibili

`XXX [Nome dell'Errore 1] – [Descrizione dell'Errore 1]`

`YYY [Nome dell'Errore 2] – [Descrizione dell'Errore 2]`

Stato della risposta:

`ZZZ [Nome dello Stato] – [Descrizione dello Stato]`

Risposta:

```
{  
  "message": (string),  
  "data": {  
    "id": (int),  
    "titolo": (string),
```

```
"contenuto": (string)
}
}
```

Di seguito ci saranno delle **indicazioni generali**, seguite da tutte le possibili azioni possibili con l'API.

Indicazioni Generali

Dominio

Il **dominio** utilizzato in questa documentazione è **di esempio** e **non** riflette un reale **sito esistente**.

Il suddetto dominio è il seguente:

www.esempio.it

Spesso, per motivi di spazio e sinteticità, verrà **abbreviato** in :

esempio.it

Ogni **sottodominio** che segue invece **riflette la reale pagina di destinazione sul sito effettivo**.

Vincoli

L'API, come già affermato in precedenza, è di **tipo RESTful**. Ciò significa che è definita dai **seguenti vincoli**:

- **Architettura Client/Server**

L'API è basata sulla **comunicazione tra Client attraverso Server** che **elaborano le richieste e restituiscono le risposte**.

- **Stateless**

Il **Server non memorizza informazioni** sullo **stato del Client** (non prevede sessioni) e le **richieste** devono essere **complete** per essere eseguite.

- **Cacheable**

Supporta il **Caching** per migliorare le prestazioni.

- **Codice On-Demand**

Il **Server può inviare al Client del codice eseguibile su richiesta** di quest'ultimo

- **Layered**

Supporta **diversi livelli di comunicazione** per **proteggere i dati** scambiati

- **Interfaccia Uniforme**

L'API ha un'**interfaccia uniforme** per **disaccoppiare** il Server da un determinato Client.

Questo vincolo prevede a sua volta **4 caratteristiche**:

- **Identificazione di Risorse** nelle **richieste**
- **Manipolazione di Risorse** attraverso la **rappresentazione**
- **Messaggi auto-descrittivi**
- **IperMedia** come motore dello **stato dell'applicazione**

Formato

All'interno di ogni risorsa, il **formato di richiesta e risposta principale** utilizzato è il **JSON**, ma è tuttavia **specificato il formato per ogni richiesta e risposta** nel caso in cui un altro formato sia utilizzato.

In ogni caso, **JSON** è da assumersi come il **formato di riferimento per questa API**, e per questo **da utilizzare anche in mancanza di documentazione adeguata o specificazione del formato** per l'impiego di suddetta API.

Utenti

L'API e il sito prevedono **tre tipologie di autenticazioni client**. Di seguito sono **elencati e brevemente descritti**, incluso una **panoramica generale dei loro privilegi**:

- **Ospite/Guest**

Un qualsiasi **client non autenticato**. È dotato di pochi privilegi, e il principale è **effettuare la registrazione** (oppure fare il login se già dispone di un account)

- **Utente/Cliente/User**

L'**Utente** (oppure **Cliente** per una **distinzione più netta** dagli altri) è l'**autenticazione principale** a cui sono **dedicati i servizi**. Essa potrà generalmente **accedere e visualizzare le risorse senza modificarle in alcun modo**. .

- **Admin**

È l'autenticazione dedicata ai **gestori del sito**. Essi infatti avranno i privilegi necessari per **visualizzare, modificare, aggiungere ed eliminare risorse, espandendosi** quindi generalmente dalle **capacità dei Clienti**.

Ogni utente, eccetto gli ospiti, sono autenticati da un Token Bearer passato negli header della richiesta ottenuto in fase di Login o Registrazione.

In caso di Token mancante, malformato o errato, la richiesta di accesso a Endpoint protetti da verrà rifiutata.

Richieste

In questa sezione saranno presentate tutte le **possibili richieste** eseguibili tramite l'API con le **condizione elencate e riassunte sopra**.

Le **Richieste** sono **suddivise in categorie**:

- **Autenticazione**
- **Accesso Risorse (categorie, film, serie TV)**
- **Aggiunta Risorse**
- **Modifica Risorse**
- **Eliminazione Risorse**
- **Profilo**
- **Pagamenti**

Ogni **richiesta** avrà un **parametro query per la lingua** al fondo dell'URL, in una forma simile a questa:

`esempio.it/api/v1/risorsa/[...]/risorsaN?chiave=valore&[...]`

Poiché presente in ogni richiesta, nelle **tabelle sottostanti** essa **non verrà indicata esplicitamente, ma lasciata sottointesa**.

In **assenza di un valore specificato**, la **lingua preferita indicata nel profilo dell'utente verrà utilizzata**.

Autenticazione

Registrazione

Utenti con Privilegio: Ospiti

Operazione: creazione di una risorsa

Formato Richiesta: JSON

Formato Risposta: JSON

Descrizione:

La richiesta permette il **setup di un nuovo account** e di conseguenza l'**accesso al sito e ai suoi servizi**.

I **dati inseriti** nel modulo di registrazione sono **inviati** al server tramite **payload della richiesta**.

I dati raccolti saranno quelli **anagrafici, amministrativi** e del **profilo**

URL: <https://esempio.it/api/v1/register>

Metodo HTTP: POST

Payload:

```
{  
  "name": (string),  
  "surname": (string),  
  "email": (string),  
  "birthdate": (date),  
  "gender": ("male"|"female"|"other"),  
  "username": (string),  
  "password": (string),  
  "country_id": (int),  
  "italian_municipality_id": (int|null),  
  "cap": (string),  
  "street_address": (string),  
  "house_number": (string),  
  "locality": (string|null),  
  "additional_info": (string|null),  
  "credit": (numeric > 0)  
}
```

Parametri:

Query String:

Errori Possibili

422 Unprocessable Entity – Errore nella validazione di uno dei campi

403 Forbidden – L'utente non è autorizzato ad eseguire l'endpoint

500 Internal Server Error – Errore del Server

Stato della risposta:

201 Created – Risorsa Creata con Successo

Risposta:

```
{
  "message": (token)
}
```

Login

Utenti con Privilegio: Ospiti, Utenti/Clienti, Admin

Operazione: accesso di una risorsa e creazione token bearer

Formato Richiesta: JSON

Formato Risposta: JSON

Descrizione:

La richiesta permette l'**accesso** al proprio **account**.

I **dati inseriti** sono **username e password**, inviati al server tramite **payload della richiesta**.

Vengono **restituiti**, oltre ai **dati principali del profilo**, il **token di autenticazione** da utilizzare per usufruire del servizio e il suo **tempo di scadenza**.

URL: <https://esempio.it/api/v1/login>

Metodo HTTP: POST

Payload:

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

Parametri:

Query String:

Errori Possibili

422 Unprocessable Entity – Errore nella validazione di uno dei campi

403 Forbidden – L'utente non è autorizzato ad eseguire l'endpoint

500 Internal Server Error – Errore del Server

Stato della risposta:

201 Created – Risorsa Creata con Successo

Risposta:

```
{
```

```
"message": (token)
}
}
```

Logout

Utenti con Privilegio: Utenti/Clienti, Admin

Operazione: chiusura di una risorsa ed eliminazione token bearer

Formato Richiesta: nessun payload

Formato Risposta: JSON

Descrizione:

La richiesta permette il **logout** dal **proprio account** e la **chiusura della sessione**

URL: <https://esempio.it/api/v1/login>

Metodo HTTP: POST

Payload:

Parametri:

Query String:

Errori Possibili

403 Forbidden – L'utente non è autorizzato ad eseguire l'endpoint

500 Internal Server Error – Errore del Server

Stato della risposta:

204 No Content– Logout Eseguito con successo

Risposta:

Forza Logout

Utenti con Privilegio: Admin

Operazione: chiusura di una risorsa ed eliminazione token bearer

Formato Richiesta: nessun payload

Formato Risposta: JSON

Descrizione:

La richiesta forza il **logout dell'Utente specificato** e la **chiusura della sessione**

URL: <https://esempio.it/api/v1/users/{user}/force-logout>

Metodo HTTP: POST

Payload:

Parametri:

user (intero): l'ID dell'utente da eliminare

Query String:**Errori Possibili**

404 Not Found - L'Utente specificato non è stato trovato oppure non è acceduto

403 Forbidden – L'utente non è autorizzato ad eseguire l'endpoint

500 Internal Server Error – Errore del Server

Stato della risposta:

204 No Content– Logout dell'Utente Eseguito con successo

Risposta:

Categoria

Elenco Categorie

Utenti con Privilegio: Utenti/Clienti, Admin

Operazione: accesso a risorse multiple

Formato Richiesta: nessun payload

Formato Risposta: JSON

Descrizione:

Richiede dal server l'elenco delle categorie dei media salvati.

Stampa un messaggio informativo in caso non ci siano categorie salvate

URL: <https://esempio.it/api/v1/categories>

Metodo HTTP: GET

Payload:

Parametri:

Query String:

Errori Possibili

403 Forbidden – L'utente non è autorizzato ad eseguire l'endpoint

500 Internal Server Error – Errore del Server

Stato della risposta:

200 Ok– Risorse Ricevute con successo

Risposta:

Admin:

```
{
  "message": "Categories Successfully Retrieved"
  "data": [
    {
      "id": (int),
      "name": (string),
      "label": (string),
      "description": (string|null),
      "created_at": (timestamp|null),
      "updated_at": (timestamp|null)
    }
    ... (per ogni categoria)
  ],
}
```

Cliente:

```
{
  "message": "Categories Successfully Retrieved"
  "data": [
    {
      "label": (string),
      "description": (string|null),
    }
    ... (per ogni categoria)
  ],
}
```

Se non ci sono Categorie:

```
{
  "message": "Categories Successfully Retrieved"
  "data": [
    "There are no Categories Present at the moment"
  ],
}
```

Singola Categoria

Utenti con Privilegio: Utenti/Clienti, Admin

Operazione: accesso a una risorsa

Formato Richiesta: nessun payload

Formato Risposta: JSON

Descrizione:

Richiede dal server i dati della categoria dei media indicata.

URL: <https://esempio.it/api/v1/categories/{category}>

Metodo HTTP: GET

Payload:**Parametri**

category (string|int): l'id o il name della categoria

Query String:**Errori Possibili**

403 Forbidden – L'utente non è autorizzato ad eseguire l'endpoint

404 Not Found– La categoria indicata non è stata trovata

500 Internal Server Error – Errore del Server

Stato della risposta:

200 Ok– Risorsa Ricevuta con successo

Risposta:**Admin:**

```
{
  "message": "Categories Successfully Retrieved"
  "data":
    {
      "id": (int),
      "name": (string),
      "label": (string),
      "description": (string|null),
      "created_at": (timestamp|null),
      "updated_at": (timestamp|null)
    }
}
```

Cliente:

```
{
  "message": "Categories Successfully Retrieved"
  "data":{
    "name": (string),
    "label": (string),
    "description": (string|null),
  }
}
```

Aggiunta Categoria

Utenti con Privilegio: Admin

Operazione: creazione risorsa

Formato Richiesta: JSON

Formato Risposta: JSON

Descrizione:

Aggiunge una categoria con i dati passati.
Si ritornano i dati della categoria creata.

URL: <https://esempio.it/api/v1/categories>

Metodo HTTP: POST

Payload:

```
{
  "name": (string),
  "label": (string),
  "description": (string|null)
```

```
}
```

Parametri

Query String:

Errori Possibili

403 Forbidden – L'utente non è autorizzato ad eseguire l'endpoint

422 Unprocessable Entity – Errore nella validazione di uno dei campi

500 Internal Server Error – Errore del Server

Stato della risposta:

201 Created – Risorsa Creata con successo

Risposta:

Admin:

```
{
  "message": "Category Successfully Created"
  "data":
    {
      "id": (int),
      "name": (string),
      "label": (string),
      "description": (string|null),
      "created_at": (timestamp|null),
      "updated_at": (timestamp|null)
    }
}
```

Modifica Categoria

Utenti con Privilegio: Admin

Operazione: modifica risorsa

Formato Richiesta: JSON

Formato Risposta: JSON

Descrizione:

Modifica la categoria indicata con i valori passati nel payload.

Ogni campo del payload è opzionale, e se omesso, la risorsa manterrà il valore precedente per quel campo.

Si ritornano i dati della categoria modificata.

URL: <https://esempio.it/api/v1/categories/{category}>

Metodo HTTP: PUT

Payload:

```
{
  "id": (int),
```

```
    "name": (string),
    "label": (string|null),
}
```

Parametri:

category (string|int): l'id o il name della categoria

Query String:**Errori Possibili**

403 Forbidden – L'utente non è autorizzato ad eseguire l'endpoint

404 Not Found – La categoria indicata non è stata trovata

422 Unprocessable Entity – Errore nella validazione di uno dei campi

500 Internal Server Error – Errore del Server

Stato della risposta:

200 Ok – Risorsa Modificata con successo

Risposta:**Admin:**

```
{
  "message": "Category Successfully Modified"
  "data":
    {
      "id": (int),
      "name": (string),
      "label": (string),
      "description": (string|null),
      "created_at": (timestamp|null),
      "updated_at": (timestamp|null)
    }
}
```

Eliminazione Categoria

Utenti con Privilegio: Admin

Operazione: eliminazione risorsa

Formato Richiesta: nessun payload

Formato Risposta: nessuna risposta

Descrizione:

Elimina la categoria indicata.

URL: <https://esempio.it/api/v1/categories/{category}>

Metodo HTTP: DELETE

Payload:

Parametri:

category (string|int): l'id o il name della categoria

Query String:

Errori Possibili

403 Forbidden – L'utente non è autorizzato ad eseguire l'endpoint

404 Not Found– La categoria indicata non è stata trovata

500 Internal Server Error – Errore del Server

Stato della risposta:

204 No Content – Risorsa Eliminata con successo

Risposta:

Admin:

Film

Elenco Film

Utenti con Privilegio: Utenti/Clienti, Admin

Operazione: accesso a risorse multiple

Formato Richiesta: nessun payload

Formato Risposta: JSON

Descrizione:

Richiede dal server l'elenco dei Film con possibilità di filtraggio per nome, categoria e anno.

Stampa un messaggio informativo in caso non ci siano film salvati (se presenti, riferito a film con i filtri specificati)

URL: [https://esempio.it/api/v1/films?name=\(string\)&category=\(string\)&year=\(int\)](https://esempio.it/api/v1/films?name=(string)&category=(string)&year=(int))

Metodo HTTP: GET

Payload:

Parametri:

Query String:

name (string): il nome, o la parte di nome, da filtrare

category (string): il nome della categoria da ricercare

year (int): l'anno del film da ricercare

Errori Possibili

403 Forbidden – L'utente non è autorizzato ad eseguire l'endpoint

500 Internal Server Error – Errore del Server

Stato della risposta:

200 Ok– Risorse Ricevute con successo

Risposta:

Admin:

```
{
  "message": "Films Successfully Retrieved"
  "data": [
    {
      "id": (int)
      "name": (string),
      "categories": [
        (string),
        (string),
        ... (per ogni categoria)
      ]
    }
  ]
}
```

```

        "description": (string|null),
        "length": (time),
        "directors": (string),
        "actors": (string),
        "year": (int)
        "created_at": (timestamp|null),
        "updated_at": (timestamp|null),
        "deleted_at": (timestamp|null)
        "files_meta": [
            {
                "id": (int),
                "filename": (string),
                "label": (string),
                "visibility": (string),
                "role": (string),
                "size": (string),
                "mime_type": (string),
                "extension": (string),
                "hash": (string),
                "created_at": (timestamp|null),
                "updated_at": (timestamp|null),
                "deleted_at": (timestamp|null),
                "url": (string, presente se "visibility": "public")
            },
            ... per ogni file associato
        ]
    }
    ... (per ogni film)
],

```

Cliente:

```

{
    "message": "Films Successfully Retrieved"
    "data": [
        {
            "name": (string),
            "categories": [
                (string),
                (string),
                ...
            ]
            "description": (string),
            "length": (time)",
            "directors": (string),
            "actors": (string),
            "year": (int),
            "files_meta": [

```

```

    {
        "label": (string),
        "role": (string),
        "size": (string),
        "mime_type": (string),
        "extension": (string),
        "url": (string)
    },
    ... per ogni file associato pubblico (visibility: public)
]

},
... (per ogni film)
]

```

Se non ci sono Film (con i dati filtri se presenti):

```

{
  "message": "Films Successfully Retrieved"
  "data": "There are no Films Present at the moment"
}

```

Singolo Film

Utenti con Privilegio: Utenti/Clienti, Admin

Operazione: accesso a risorsa

Formato Richiesta: nessun payload

Formato Risposta: JSON

Descrizione:

Richiede dal server le informazioni relative al film specificato

URL: <https://esempio.it/api/v1/films/{filmId}>

Metodo HTTP: GET

Payload:

Parametri:

filmId (int): l'ID del film richiesto

Query String:

Errori Possibili

403 Forbidden – L'utente non è autorizzato ad eseguire l'endpoint

404 Not Found– Il film indicato non è stato trovato

500 Internal Server Error – Errore del Server

Stato della risposta:

200 Ok– Risorsa Ricevuta con successo

Risposta:

Admin:

```
{
  "message": "Film Successfully Retrieved"
  "data": {
    "id": (int)
    "name": (string),
    "categories": [
      (string),
      (string),
      ...
    ]
    "description": (string|null),
    "length": (time),
    "directors": (string),
    "actors": (string),
    "year": (int),
    "created_at": (timestamp|null),
    "updated_at": (timestamp|null),
    "deleted_at": (timestamp|null)
    "files_meta": [
      {
        "id": (int),
        "filename": (string),
        "label": (string),
        "visibility": (string),
        "role": (string),
        "size": (string),
        "mime_type": (string),
        "extension": (string),
        "hash": (string),
        "created_at": (timestamp|null),
        "updated_at": (timestamp|null),
        "deleted_at": (timestamp|null),
        "url": (string, presente se "visibility": "public")
      },
      ... per ogni file associato
    ]
  }
}
```

Cliente:

```
{
  "message": "Film Successfully Retrieved"
  "data": {
    "name": (string),
```

```

    "categories": [
        (string),
        (string),
        ...
    ]
    "description": (string|null),
    "length": (time),
    "directors": (string),
    "actors": (string),
    "year": (int),
    "files_meta": [
        {
            "label": (string),
            "visibility": (string),
            "role": (string),
            "size": (string),
            "mime_type": (string),
            "extension": (string),
            "url": (string)
        },
        ... per ogni file associato pubblico
    ]
}

```

Aggiunta Film

Utenti con Privilegio: Admin

Operazione: creazione di una nuova risorsa

Formato Richiesta: JSON

Formato Risposta: JSON

Descrizione:

Richiede al server di creare un nuovo film, e ritorna le informazioni della risorsa creata

URL: <https://esempio.it/api/v1/films>

Metodo HTTP: POST

Payload:

```

{
    "name": (string),
    "categories": [
        (string),
        (string),
        ... per ogni categoria
    ]
    "description": (string|null),
    "length": (time),
    "directors": (string),

```

```

    "actors": (string),
    "year": (int),
    "files_meta": [
        {
            "role": (string),
            "visibility": (string),
            "label": (string)
        },
        ... per ogni file inviato
    ]
}

```

Parametri

Query String:

Errori Possibili

- 403 Forbidden – L'utente non è autorizzato ad eseguire l'endpoint
- 422 Unprocessable Entity – Errore nella validazione di uno dei campi
- 500 Internal Server Error – Errore del Server

Stato della risposta:

- 201 Created – Risorsa Creata con successo

Risposta:

Admin:

```

{
  "message": "Film Successfully Created"
  "data":
    {
      "id": (int)
      "name": (string),
      "categories": [
        (string),
        (string),
        ... per ogni categoria
      ]
      "description": (string|null),
      "length": (time),
      "directors": (string),
      "actors": (string),
      "year": (int),
      "files_meta": [
        {
          "id": (int),
          "filename": (string),
          "label": (string),

```

```

        "visibility": (string),
        "role": (string),
        "size": (string),
        "mime_type": (string),
        "extension": (string),
        "hash": (string),
        "created_at": (timestamp|null),
        "updated_at": (timestamp|null),
        "deleted_at": (timestamp|null),
        "url": (string, presente se "visibility": "public")
    },
    ... per ogni file associato
]
"created_at": (timestamp|null),
"updated_at": (timestamp|null),"
"deleted_at": (timestamp|null),
}
}

```

Modifica Film

Utenti con Privilegio: Admin

Operazione: modifica risorsa

Formato Richiesta: JSON

Formato Risposta: JSON

Descrizione:

Modifica il film indicato con i valori passati nel payload.

Ogni campo del payload è opzionale, e se omissso, la risorsa manterrà il valore precedente per quel campo.

Si ritornano i dati aggiornati del film.

URL: <https://esempio.it/api/v1/films/{filmId}>

Metodo HTTP: PUT

Payload:

```

{
    "name": (string),
    "categories": [
        (string),
        (string),
        ... per ogni nuova categoria (rimpiazzeranno quelle antecedenti)
    ]
    "description": (string|null),
    "length": (time),
    "directors": (string),
    "actors": (string),
    "year": (int),
}

```



```

    "files_meta": [
        {
            "role": (string),
            "visibility": (string),
            "label": (string)
        },
        ... per ogni file da aggiornare
    ]
}

```

Parametri:

filmId (int): l'ID del film da modificare

Query String:

Errori Possibili

403 Forbidden – L'utente non è autorizzato ad eseguire l'endpoint

404 Not Found– Il film indicato non è stato trovato

422 Unprocessable Entity – Errore nella validazione di uno dei campi

500 Internal Server Error – Errore del Server

Stato della risposta:

200 Ok – Risorsa Modificata con successo

Risposta:

Admin:

```

{
  "message": "Film Successfully Modified"
  "data":
    {
      "id": (int)
      "name": (string),
      "categories": [
        (string),
        (string),
        ... per ogni categoria ora associata
      ]
      "description": (string|null),
      "length": (time),
      "directors": (string),
      "actors": (string),
      "year": (int),
      "files_meta": [
        {
          "id": (int),

```

```

        "filename": (string),
        "label": (string),
        "visibility": (string),
        "role": (string),
        "size": (string),
        "mime_type": (string),
        "extension": (string),
        "hash": (string),
        "created_at": (timestamp|null),
        "updated_at": (timestamp|null),
        "deleted_at": (timestamp|null),
        "url": (string, presente se "visibility": "public")
    },
    ... per ogni file associato
]
"created_at": (timestamp|null),
"updated_at": (timestamp|null),"
"deleted_at": (timestamp|null),
}
}

```

Eliminazione Film

Utenti con Privilegio: Admin

Operazione: eliminazione risorsa

Formato Richiesta: nessun payload

Formato Risposta: nessuna risposta

Descrizione:

Elimina il film indicato (in maniera reversibile).

URL: <https://esempio.it/api/v1/films/{filmId}>

Metodo HTTP: DELETE

Payload:

Parametri:

filmId (int): l'ID del film da eliminare

Query String:

Errori Possibili

403 Forbidden – L'utente non è autorizzato ad eseguire l'endpoint

409 Conflict – Il film indicato è già stato eliminato

404 Not Found– Il film indicato non è stato trovato

500 Internal Server Error – Errore del Server

Stato della risposta:

204 No Content – Risorsa Eliminata con successo

Risposta:

Admin:

Recupero Film

Utenti con Privilegio: Admin

Operazione: recupero risorsa

Formato Richiesta: nessun payload

Formato Risposta: JSON

Descrizione:

Recupera il film indicato e ne ritorna le informazioni.

URL: <https://esempio.it/api/v1/films/{filmId}/restore>

Metodo HTTP: PUT

Payload:

Parametri:

filmId (int): l'ID del film da recuperare

Query String:

Errori Possibili

403 Forbidden – L'utente non è autorizzato ad eseguire l'endpoint

404 Not Found– Il film indicato non è stato trovato

409 Conflict – Il film indicato non è eliminato

500 Internal Server Error – Errore del Server

Stato della risposta:

200 OK – Risorsa Recuperata con successo

Risposta:

Admin:

```
{
  "message": "Film Successfully Restored"
  "data":
    {
      "id": (int)
      "name": (string),
      "categories": [
        (string),
        (string),
        ... per ogni categoria
      ]
      "description": (string|null),
      "length": (time),
      "directors": (string),
```

```

    "actors": (string),
    "year": (int),
    "files_meta": [
        {
            "id": (int),
            "filename": (string),
            "label": (string),
            "visibility": (string),
            "role": (string),
            "size": (string),
            "mime_type": (string),
            "extension": (string),
            "hash": (string),
            "created_at": (timestamp|null),
            "updated_at": (timestamp|null),
            "deleted_at": (timestamp|null),
            "url": (string, presente se "visibility": "public")
        },
        ... per ogni file associato
    ]
    "created_at": (timestamp|null),
    "updated_at": (timestamp|null),
    "deleted_at": (timestamp|null),
}
}

```

Eliminazione Definitiva Film

Utenti con Privilegio: Admin

Operazione: eliminazione risorsa definitiva

Formato Richiesta: nessun payload

Formato Risposta: nessuna risposta

Descrizione:

Cancella il film indicato definitivamente, previa eliminazione "soft" (vedi Eliminazione Film). Tutte le informazioni associate sono cancellate e irrecuperabili

URL: <https://esempio.it/api/v1/films/{filmId}/force>

Metodo HTTP: DELETE

Payload:

Parametri:

filmId (int): l'ID del film da eliminare definitivamente

Query String:

Errori Possibili

403 Forbidden – L'utente non è autorizzato ad eseguire l'endpoint

409 Conflict – Il film indicato non è stato eliminato

404 Not Found– Il film indicato non è stato trovato

500 Internal Server Error – Errore del Server

Stato della risposta:

204 No Content – Risorsa Eliminata Definitivamente con successo

Risposta:

Admin:

Serie TV

Elenco Serie

Utenti con Privilegio: Utenti/Clienti, Admin

Operazione: accesso a risorse multiple

Formato Richiesta: nessun payload

Formato Risposta: JSON

Descrizione:

Richiede dal server l'elenco delle Serie con possibilità di filtraggio per nome, categoria, anno

Stampa un messaggio informativo in caso non ci siano serie salvate (se presenti, riferito a serie con i filtri specificati)

URL: [https://esempio.it/api/v1/tv-series?name=\(string\)&category=\(string\)&year=\(int\)](https://esempio.it/api/v1/tv-series?name=(string)&category=(string)&year=(int))

Metodo HTTP: GET

Payload:

Parametri:

Query String:

name (string): il nome, o la parte di nome, da filtrare

category (string): il nome della categoria da ricercare

year (int): anno in cui la serie era trasmessa

Errori Possibili

403 Forbidden – L'utente non è autorizzato ad eseguire l'endpoint

500 Internal Server Error – Errore del Server

Stato della risposta:

200 Ok– Risorse Ricevute con successo

Risposta:

Admin:

```
{
  "message": "Tv Series Successfully Retrieved"
  "data": [
    {
      "id": (int)
      "name": (string),
      "categories": [
        (string),
        (string),
        ... (per ogni categoria)
      ]
    }
  ]
}
```

```

        "description": (string|null),
        "season_count": (int),
        "episode_count": (int),
        "directors": (string),
        "actors": (string),
        "start_year": (int),
        "end_year": (int),
        "created_at": (timestamp|null),
        "updated_at": (timestamp|null),
        "deleted_at": (timestamp|null)
        "files_meta": [
            {
                "id": (int),
                "filename": (string),
                "label": (string),
                "visibility": (string),
                "role": (string),
                "size": (string),
                "mime_type": (string),
                "extension": (string),
                "hash": (string),
                "created_at": (timestamp|null),
                "updated_at": (timestamp|null),
                "deleted_at": (timestamp|null),
                "url": (string, presente se "visibility": "public")
            },
            ... per ogni file associato
        ]
    }
    ... (per ogni serie)
],

```

Cliente:

```

{
    "message": "TV Series Successfully Retrieved"
    "data": [
        {
            "name": (string),
            "categories": [
                (string),
                (string),
                ...
            ]
            "description": (string),
            "length": (time)",
            "directors": (string),
            "actors": (string),

```

```

        "year": (int)
        "files_meta": [
            {
                "label": (string),
                "role": (string),
                "size": (string),
                "mime_type": (string),
                "extension": (string),
                "url": (string)
            },
            ... per ogni file associato pubblico
        ]
    },
    ... (per ogni serie)
]

```

Se non ci sono Film (con i dati filtri se presenti):

```

{
    "message": "Tv Series Successfully Retrieved"
    "data": "There are no TV Series Present at the moment"
}

```

Singola Serie

Utenti con Privilegio: Utenti/Clienti, Admin

Operazione: accesso a risorsa singola

Formato Richiesta: nessun payload

Formato Risposta: JSON

Descrizione:

Richiede dal server le informazioni relative alla Serie TV specificata

URL: <https://esempio.it/api/v1/tv-series/{tvSeriesId}>

Metodo HTTP: GET

Payload:

Parametri:

tvSeriesId (int): l'ID della serie richiesta

Query String:

Errori Possibili

403 Forbidden – L'utente non è autorizzato ad eseguire l'endpoint

404 Not Found– La series indicata non è stata trovata

500 Internal Server Error – Errore del Server

Stato della risposta:

200 Ok– Risorsa Ricevuta con successo

Risposta:

Admin:

```
{
  "message": "TV Series Successfully Retrieved"
  "data": {
    "id": (int)
    "name": (string),
    "categories": [
      (string),
      (string),
      ...
    ]
    "description": (string|null),
    "season_count": (int),
    "episode_count": (int),
    "directors": (string),
    "actors": (string),
    "start_year": (int),
    "end_year": (int),
    "created_at": (timestamp|null),
    "updated_at": (timestamp|null),
    "deleted_at": (timestamp|null)
    "files_meta": [
      {
        "id": (int),
        "filename": (string),
        "label": (string),
        "visibility": (string),
        "role": (string),
        "size": (string),
        "mime_type": (string),
        "extension": (string),
        "hash": (string),
        "created_at": (timestamp|null),
        "updated_at": (timestamp|null),
        "deleted_at": (timestamp|null),
        "url": (string, presente se "visibility": "public")
      },
      ... per ogni file associato
    ]
  }
}
```

Cliente:

```
{
  "message": "TV Series Successfully Retrieved"
```

```
"data": {  
    "name": (string),  
    "categories": [  
        (string),  
        (string),  
        ...  
    ]  
    "description": (string|null),  
    "season_count": (int),  
    "episode_count": (int),  
    "directors": (string),  
    "actors": (string),  
    "start_year": (int),  
    "end_year": (int),  
    "files_meta": [  
        {  
            "label": (string),  
            "visibility": (string),  
            "role": (string),  
            "size": (string),  
            "mime_type": (string),  
            "extension": (string),  
            "url": (string)  
        },  
        ... per ogni file associato pubblico  
    ]  
}
```

Aggiunta Serie

Utenti con Privilegio: Admin

Operazione: creazione di una nuova risorsa

Formato Richiesta: JSON

Formato Risposta: JSON

Descrizione:

Richiede al server di creare una nuova serie, e ritorna le informazioni della risorsa creata

URL: <https://esempio.it/api/v1/tv-series>

Metodo HTTP: POST

Payload:

```
{
  "name": (string),
  "categories": [
    (string),
    (string),
    ... per ogni categoria
  ]
  "description": (string|null),
  "directors": (string),
  "actors": (string),
  "start_year": (int),
  "end_year": (int),
  "files_meta": [
    {
      "role": (string),
      "visibility": (string),
      "label": (string)
    },
    ... per ogni file inviato
  ]
}
```

Parametri

Query String:

Errori Possibili

403 Forbidden – L'utente non è autorizzato ad eseguire l'endpoint

422 Unprocessable Entity – Errore nella validazione di uno dei campi

500 Internal Server Error – Errore del Server

Stato della risposta:

201 Created – Risorsa Creata con successo

Risposta:

Admin:

```
{
  "message": "TV Series Successfully Created"
  "data":
    {
      "id": (int)
      "name": (string),
      "categories": [
        (string),
        (string),
        ... per ogni categoria
      ]
      "description": (string|null),
      "season_count": (int),
      "episode_count": (int),
      "directors": (string),
      "actors": (string),
      "start_year": (int),
      "end_year": (int),
      "files_meta": [
        {
          "id": (int),
          "filename": (string),
          "label": (string),
          "visibility": (string),
          "role": (string),
          "size": (string),
          "mime_type": (string),
          "extension": (string),
          "hash": (string),
          "created_at": (timestamp|null),
          "updated_at": (timestamp|null),
          "deleted_at": (timestamp|null),
          "url": (string, presente se "visibility": "public")
        },
        ... per ogni file associato
      ]
      "created_at": (timestamp|null),
      "updated_at": (timestamp|null),
      "deleted_at": (timestamp|null),
    }
}
```

Modifica Serie

Utenti con Privilegio: Admin

Operazione: modifica parziale o totale di una risorsa

Formato Richiesta: JSON

Formato Risposta: JSON

Descrizione:

Modifica il film indicato con i valori passati nel payload.

Ogni campo del payload è opzionale, e se omesso, la risorsa manterrà il valore precedente per quel campo.

Si ritornano i dati aggiornati della serie.

URL: `https://esempio.it/api/v1/tv-series/{tvSeriesId}`

Metodo HTTP: PUT

Payload:

```
{
  "name": (string),
  "categories": [
    (string),
    (string),
    ... per ogni nuova categoria (rimpiazzeranno quelle antecedenti)
  ]
  "description": (string|null),
  "directors": (string),
  "actors": (string),
  "start_year": (int),
  "end_year": (int),

  "files_meta": [
    {
      "role": (string),
      "visibility": (string),
      "label": (string)
    },
    ... per ogni file da aggiornare
  ]
}
```

Parametri:

filmId (int): l'ID del film da modificare

Query String:

Errori Possibili

403 Forbidden – L'utente non è autorizzato ad eseguire l'endpoint

404 Not Found– La serie indicata non è stata trovata

422 Unprocessable Entity – Errore nella validazione di uno dei campi

500 Internal Server Error – Errore del Server

Stato della risposta:

200 Ok – Risorsa Modificata con successo

Risposta:

Admin:

```
{
  "message": "TV Series Successfully Modified"
  "data":
    {
      "id": (int)
      "name": (string),
      "categories": [
        (string),
        (string),
        ... per ogni categoria ora associata
      ]
      "description": (string|null),
      "directors": (string),
      "actors": (string),
      "start_year": (int),
      "end_year": (int),
      "files_meta": [
        {
          "id": (int),
          "filename": (string),
          "label": (string),
          "visibility": (string),
          "role": (string),
          "size": (string),
          "mime_type": (string),
          "extension": (string),
          "hash": (string),
          "created_at": (timestamp|null),
          "updated_at": (timestamp|null),
          "deleted_at": (timestamp|null),
          "url": (string, presente se "visibility": "public")
        },
        ... per ogni file associato
      ]
      "created_at": (timestamp|null),
      "updated_at": (timestamp|null),
      "deleted_at": (timestamp|null),
    }
}
```

Eliminazione Serie

Utenti con Privilegio: Admin

Operazione: eliminazione risorsa

Formato Richiesta: nessun payload

Formato Risposta: nessuna risposta

Descrizione:

Elimina la serie indicata (in maniera reversibile).

URL: <https://esempio.it/api/v1/tv-series/{tvSeriesId}>

Metodo HTTP: DELETE

Payload:

Parametri:

tvSeriesId (int): l'ID della serie richiesta

Query String:

Errori Possibili

403 Forbidden – L'utente non è autorizzato ad eseguire l'endpoint

409 Conflict – La serie indicata è già stata eliminata

404 Not Found – La serie indicata non è stata trovata

500 Internal Server Error – Errore del Server

Stato della risposta:

204 No Content – Risorsa Eliminata con successo

Risposta:

Admin:

Recupero Serie

Utenti con Privilegio: Admin

Operazione: recupero risorsa

Formato Richiesta: nessun payload

Formato Risposta: JSON

Descrizione:

Recupera la serie indicata e ne ritorna le informazioni.

URL: <https://esempio.it/api/v1/tv-series/{tvSeriesId}/restore>

Metodo HTTP: PUT

Payload:

Parametri:

tvSeriesId (int): l'ID della serie richiesta

Query String:

Errori Possibili

403 Forbidden – L'utente non è autorizzato ad eseguire l'endpoint

409 Conflict – La serie indicata non è eliminata

404 Not Found – La serie indicata non è stata trovata

500 Internal Server Error – Errore del Server

Stato della risposta:

200 OK – Risorsa Recuperata con successo

Risposta:

Admin:

```
{
  "message": "TV Series Successfully Restored"
  "data": {
    "id": (int)
    "name": (string),
    "categories": [
      (string),
      (string),
      ... per ogni categoria
    ]
    "description": (string|null),
    "season_count": (int),
    "episode_count": (int),
    "directors": (string),
    "actors": (string),
    "start_year": (int),
    "end_year": (int),
    "files_meta": [
      {
        "id": (int),
        "filename": (string),
        "label": (string),
        "visibility": (string),
        "role": (string),
        "size": (string),
        "mime_type": (string),
        "extension": (string),
        "hash": (string),
        "created_at": (timestamp|null),
        "updated_at": (timestamp|null),
        "deleted_at": (timestamp|null),
        "url": (string, presente se "visibility": "public")
      },
    ]
  }
}
```



```
        ... per ogni file associato
    ]
    "created_at": (timestamp|null),
    "updated_at": (timestamp|null),"
    "deleted_at": (timestamp|null),
}
```

Eliminazione Definitiva Serie

Utenti con Privilegio: Admin

Operazione: eliminazione risorsa definitiva

Formato Richiesta: nessun payload

Formato Risposta: nessuna risposta

Descrizione:

Cancella la serie indicata definitivamente, previa eliminazione “soft” (vedi Eliminazione Serie).

Tutte le informazioni associate sono cancellate e irrecuperabili, così come gli episodi ad essa collegati

URL: <https://esempio.it/api/v1/tv-series/{tvSeriesId}/force>

Metodo HTTP: DELETE

Payload:

Parametri:

tvSeriesId (int): l'ID della serie richiesta

Query String:

Errori Possibili

403 Forbidden – L'utente non è autorizzato ad eseguire l'endpoint

409 Conflict – La serie indicata non è eliminata

404 Not Found– La serie indicata non è stata trovata

500 Internal Server Error – Errore del Server

Stato della risposta:

204 No Content – Risorsa Eliminata Definitivamente con successo

Risposta:

Admin:

Episodio

Elenco Episodi

Utenti con Privilegio: User/Cliente, Admin

Operazione: accesso di una risorsa

Formato Richiesta: nessun payload

Formato Risposta: JSON

Descrizione:

Richiede dal server l'elenco degli episodi della serie selezionata con possibilità di filtraggio per stagione.

Stampa un messaggio informativo in caso non ci siano episodi salvati per la serie selezionata (se presenti, riferito a episodi con i filtri specificati)

URL: [https://esempio.it/api/v1/tv-series/{tvSeriesId}/episodes?season=\(int\)](https://esempio.it/api/v1/tv-series/{tvSeriesId}/episodes?season=(int))

Metodo HTTP: GET

Payload:

Parametri:

tvSeriesId (int): l'ID della serie richiesta

Query String:

season (int): il numero della stagione da filtrare

Errori Possibili

403 Forbidden – L'utente non è autorizzato ad eseguire l'endpoint

404 Not Found – La Serie TV indicata non è stata trovata

500 Internal Server Error – Errore del Server

Stato della risposta:

200 Ok– Risorse Ricevute con successo

Risposta:

Admin:

```
{
  "message": "Episodes Successfully Retrieved"
  "data": [
    {
      "id": (int)
      "tv_series_id": (int)
      "name": (string),
      "description": (string|null),
      "season_number": (int),
      "episode_number": (int),
```

```

        "length": (time)
        "year": (int),
        "created_at": (timestamp|null),
        "updated_at": (timestamp|null),
        "deleted_at": (timestamp|null)
        "files_meta": [
            {
                "id": (int),
                "filename": (string),
                "label": (string),
                "visibility": (string),
                "role": (string),
                "size": (string),
                "mime_type": (string),
                "extension": (string),
                "hash": (string),
                "created_at": (timestamp|null),
                "updated_at": (timestamp|null),
                "deleted_at": (timestamp|null),
                "url": (string, presente se "visibility": "public")
            },
            ... per ogni file associato
        ]
    }
    ... (per ogni episodio)
],

```

Cliente:

```

{
    "message": "Episodes Successfully Retrieved"
    "data": [
        {
            "tv_series_id": (int)
            "name": (string),
            "description": (string|null),
            "season_number": (int),
            "episode_number": (int),
            "length": (time)
            "year": (int),
            "files_meta": [
                {
                    "label": (string),
                    "role": (string),
                    "size": (string),
                    "mime_type": (string),
                    "extension": (string),
                    "url": (string)
                }
            ]
        }
    ]
}

```

```

        },
        ... per ogni file associato pubblico
    ]
},
... (per ogni serie)
]

```

Se non ci sono Film (con i dati filtri se presenti):

```

{
  "message": "Episodes Successfully Retrieved"
  "data": "There are no Episodes Present in this TV Series at the moment"
}

```

Singolo Episodio

Utenti con Privilegio: Utente/Cliente, Admin

Operazione: accesso di una risorsa

Formato Richiesta: nessun payload

Formato Risposta: JSON

Descrizione:

Richiede al server di le informazioni di un determinato episodio di una Serie TV data

URL: <https://esempio.it/api/v1/tv-series/{tvSeriesId}/episodes/{episodeId}>

Metodo HTTP: GET

Payload:

Parametri:

tvSeriesId (int): l'ID della serie richiesta

episodeId (int): l'ID dell'episodio richiesto

Query String:

Errori Possibili

403 Forbidden – L'utente non è autorizzato ad eseguire l'endpoint

404 Not Found – La Serie TV oppure l'episodio indicato non è stato trovato

500 Internal Server Error – Errore del Server

Stato della risposta:

200 Ok– Risorsa Ricevuta con successo

Risposta:

Admin:

```

{
  "message": "Episode Successfully Retrieved"
  "data": {

```

```

    "id": (int)
    "tv_series_id": (int)
    "name": (string),
    "description": (string|null),
    "season_number": (int),
    "episode_number": (int),
    "length": (time)
    "year": (int),
    "created_at": (timestamp|null),
    "updated_at": (timestamp|null),
    "deleted_at": (timestamp|null)
    "files_meta": [
        {
            "id": (int),
            "filename": (string),
            "label": (string),
            "visibility": (string),
            "role": (string),
            "size": (string),
            "mime_type": (string),
            "extension": (string),
            "hash": (string),
            "created_at": (timestamp|null),
            "updated_at": (timestamp|null),
            "deleted_at": (timestamp|null),
            "url": (string, presente se "visibility": "public")
        },
        ... per ogni file associato
    ]
}

```

Cliente:

```

{
    "message": "Episode Successfully Retrieved"
    "data": {
        "tv_series_id": (int)
        "name": (string),
        "description": (string|null),
        "season_number": (int),
        "episode_number": (int),
        "length": (time)
        "year": (int),
        "files_meta": [
            {
                "label": (string),
                "visibility": (string),
                "role": (string),
                "size": (string),

```

```

        "mime_type": (string),
        "extension": (string),
        "url": (string)
    },
    ... per ogni file associato pubblico
]
}

```

Aggiunta Episodio

Utenti con Privilegio: Admin

Operazione: creazione di una nuova risorsa

Formato Richiesta: JSON

Formato Risposta: JSON

Descrizione:

Richiede al server di creare un nuovo episodio associato alla serie passata, e ritorna le informazioni della risorsa creata

URL: <https://esempio.it/api/v1/tv-series/{tvSeriesId}/episodes>

Metodo HTTP: POST

Payload:

```

{
    "name": (string),
    "description": (string|null),
    "season_number": (int),
    "episode_number": (int),
    "length": (time)
    "year": (int),
    "files_meta": [
        {
            "role": (string),
            "visibility": (string),
            "label": (string)
        },
        ... per ogni file inviato
    ]
}

```

Parametri

tvSeriesId (int): l'ID della serie richiesta

Query String:

Errori Possibili

403 Forbidden – L'utente non è autorizzato ad eseguire l'endpoint

404 Not Found – La Serie TV Indicata non è stata trovata

422 Unprocessable Entity – Errore nella validazione di uno dei campi

500 Internal Server Error – Errore del Server

Stato della risposta:

201 Created – Risorsa Creata con successo

Risposta:

Admin:

```
{
  "message": "Episode Successfully Created"
  "data": {
    "id": (int)
    "tv_series_id": (int)
    "name": (string),
    "description": (string|null),
    "season_number": (int),
    "episode_number": (int),
    "length": (time)
    "year": (int),
    "created_at": (timestamp|null),
    "updated_at": (timestamp|null),
    "deleted_at": (timestamp|null)
    "files_meta": [
      {
        "id": (int),
        "filename": (string),
        "label": (string),
        "visibility": (string),
        "role": (string),
        "size": (string),
        "mime_type": (string),
        "extension": (string),
        "hash": (string),
        "created_at": (timestamp|null),
        "updated_at": (timestamp|null),
        "deleted_at": (timestamp|null),
        "url": (string, presente se "visibility": "public")
      },
      ... per ogni file associato
    ]
  }
}
```

Modifica Episodio

Utenti con Privilegio: Admin

Operazione: modifica parziale o totale di una risorsa

Formato Richiesta: JSON

Formato Risposta: JSON

Descrizione:

Richiede al server di modificare alcune (o tutte) le informazioni dell'episodio con quelle inviate nel Payload. Solo le informazioni inviate sono modificate

URL: <https://esempio.it/api/v1/tv-series/{tvSeriesId}/episodes/{episodeId}>

Metodo HTTP: PUT

Payload:

```
{
  "name": (string),
  "description": (string|null),
  "season_number": (int),
  "episode_number": (int),
  "length": (time)
  "year": (int),
  "files_meta": [
    {
      "role": (string),
      "visibility": (string),
      "label": (string)
    },
    ... per ogni file inviato
  ]
}
```

Parametri

tvSeriesId (int): l'ID della serie richiesta

episodeId (int): l'ID dell'episodio richiesto

Query String:

Errori Possibili

403 Forbidden – L'utente non è autorizzato ad eseguire l'endpoint

404 Not Found – La Serie TV oppure l'episodio indicato non è stato trovato

422 Unprocessable Entity – Errore nella validazione di uno dei campi

500 Internal Server Error – Errore del Server

Stato della risposta:

200 OK – Risorsa Modificata con successo

Risposta:

Admin:

```
{
  "message": "Episode Successfully Created"
  "data": {
    "id": (int)
  }
}
```



```

    "tv_series_id": (int)
    "name": (string),
    "description": (string|null),
    "season_number": (int),
    "episode_number": (int),
    "length": (time)
    "year": (int),
    "created_at": (timestamp|null),
    "updated_at": (timestamp|null),
    "deleted_at": (timestamp|null)
    "files_meta": [
        {
            "id": (int),
            "filename": (string),
            "label": (string),
            "visibility": (string),
            "role": (string),
            "size": (string),
            "mime_type": (string),
            "extension": (string),
            "hash": (string),
            "created_at": (timestamp|null),
            "updated_at": (timestamp|null),
            "deleted_at": (timestamp|null),
            "url": (string, presente se "visibility": "public")
        },
        ... per ogni file associato
    ]
}

```

Eliminazione Episodio

Utenti con Privilegio: Admin

Operazione: eliminazione risorsa

Formato Richiesta: nessun payload

Formato Risposta: nessuna risposta

Descrizione:

Elimina l'episodio indicato (in maniera reversibile).

URL: <https://esempio.it/api/v1/tv-series/{tvSeriesId}/episodes/{episodeId}>

Metodo HTTP: DELETE

Payload:

Parametri:

tvSeriesId (int): l'ID della serie richiesta

episodeId (int): l'ID dell'episodio richiesto

Query String:

Errori Possibili

- 403 Forbidden – L'utente non è autorizzato ad eseguire l'endpoint
- 409 Conflict – L'episodio è già stato eliminato
- 404 Not Found– La serie oppure l'episodio indicato non è stato trovato
- 500 Internal Server Error – Errore del Server

Stato della risposta:

- 204 No Content – Risorsa Eliminata con successo

Risposta:

Admin:

Recupero Episodio

Utenti con Privilegio: Admin

Operazione: recupero risorsa

Formato Richiesta: nessun payload

Formato Risposta: JSON

Descrizione:

Recupera l'episodio indicato e ne ritorna le informazioni.

URL: <https://esempio.it/api/v1/tv-series/{tvSeriesId}/episodes/{episodeId}/restore>

Metodo HTTP: PUT

Payload:

Parametri:

- tvSeriesId (int): l'ID della serie richiesta
- episodeId (int): l'ID dell'episodio richiesto

Query String:

Errori Possibili

- 403 Forbidden – L'utente non è autorizzato ad eseguire l'endpoint
- 409 Conflict – L'episodio non è eliminato
- 404 Not Found– La serie oppure l'episodio indicato non è stato trovato
- 500 Internal Server Error – Errore del Server

Stato della risposta:

- 200 OK– Risorsa Recuperata con successo

Risposta:

Admin:

```
{  
  "message": "Episode Successfully Restored"
```

```

"data": {
    "id": (int)
    "tv_series_id": (int)
    "name": (string),
    "description": (string|null),
    "season_number": (int),
    "episode_number": (int),
    "length": (time)
    "year": (int),
    "created_at": (timestamp|null),
    "updated_at": (timestamp|null),
    "deleted_at": (timestamp|null)
    "files_meta": [
        {
            "id": (int),
            "filename": (string),
            "label": (string),
            "visibility": (string),
            "role": (string),
            "size": (string),
            "mime_type": (string),
            "extension": (string),
            "hash": (string),
            "created_at": (timestamp|null),
            "updated_at": (timestamp|null),
            "deleted_at": (timestamp|null),
            "url": (string, presente se "visibility": "public")
        },
        ... per ogni file associato
    ]
}

```

Eliminazione Definitiva Serie

Utenti con Privilegio: Admin

Operazione: eliminazione risorsa definitiva

Formato Richiesta: nessun payload

Formato Risposta: nessuna risposta

Descrizione:

Cancella l'episodio indicato definitivamente, previa eliminazione "soft" (vedi Eliminazione Episodio). Tutte le informazioni associate sono cancellate e irrecuperabili

URL: <https://esempio.it/api/v1/tv-series/{tvSeriesId}/episodes/{episodeId}/force>

Metodo HTTP: DELETE

Payload:

Parametri:

tvSeriesId (int): l'ID della serie richiesta

episodeId (int): l'ID dell'episodio richiesto

Query String:**Errori Possibili**

403 Forbidden – L'utente non è autorizzato ad eseguire l'endpoint

409 Conflict – L'episodio indicato non è eliminato

404 Not Found – La serie o l'episodio indicato non è stato trovato

500 Internal Server Error – Errore del Server

Stato della risposta:

204 No Content – Risorsa Eliminata Definitivamente con successo

Risposta:**Admin:**

Utenti

Elenco Utenti

Utenti con Privilegio: Admin

Operazione: accesso di una risorsa

Formato Richiesta: nessun payload

Formato Risposta: JSON

Descrizione:

Richiede dal server l'elenco degli utenti registrati e le loro informazioni non sensibili

Stampa un messaggio informativo in caso non ci siano episodi salvati per la serie selezionata

URL: <https://esempio.it/api/v1/users>

Metodo HTTP: GET

Payload:

Parametri:

Query String:

Errori Possibili

403 Forbidden – L'utente non è autorizzato ad eseguire l'endpoint

500 Internal Server Error – Errore del Server

Stato della risposta:

200 Ok– Risorse Ricevute con successo

Risposta:

Admin:

```
{
  "message": "Users Successfully Retrieved"
  "data": [
    {
      "id": (int)
      "username": (string),
      "roles": [
        (string),
        (string),
        ... (per ogni ruolo)
      ]
      "state_id": (int),
      "state_until": (timestamp|null),
      "created_at": (timestamp|null),
      "updated_at": (timestamp|null),
    }
  ]
}
```

```

        "deleted_at": (timestamp|null)
      }
      ... (per ogni utente)
    ],
  }
}

```

Se non ci sono Utenti

```

{
  "message": "Users Successfully Retrieved"
  "data": "There are no Users Present at the moment"
}

```

Singolo Utente

Utenti con Privilegio: Admin

Operazione: accesso di una risorsa

Formato Richiesta: nessun payload

Formato Risposta: JSON

Descrizione:

Richiede al server di le informazioni non sensibili dell'utente selezionato

URL: <https://esempio.it/api/v1/users/{userId}>

Metodo HTTP: GET

Payload:

Parametri:

userId (int): l'ID dell'utente richiesto

Query String:

Errori Possibili

403 Forbidden – L'utente non è autorizzato ad eseguire l'endpoint

404 Not Found – L'utente non è stato trovato

500 Internal Server Error – Errore del Server

Stato della risposta:

200 Ok– Risorsa Ricevuta con successo

Risposta:

Admin:

```

{
  "message": "User Successfully Retrieved"
  "data": {
    "id": (int)
    "username": (string),
    "roles": [

```

```

        (string),
        (string),
        ... (per ogni ruolo)
    ]
    "state_id": (int),
    "state_until": (timestamp|null),
    "created_at": (timestamp|null),
    "updated_at": (timestamp|null),
    "deleted_at": (timestamp|null)
}

```

Aggiunta Utente

Utenti con Privilegio: Admin

Operazione: creazione di una nuova risorsa

Formato Richiesta: JSON

Formato Risposta: JSON

Descrizione:

Richiede al server di creare un nuovo utente, e ritorna le informazioni non sensibili della risorsa creata

URL: <https://esempio.it/api/v1/users>

Metodo HTTP: POST

Payload:

```

{
    "role": (string, ruolo più alto assegnato)
    "name": (string),
    "surname": (string),
    "email": (string),
    "birthdate": (date),
    "gender": ("male"|"female"|"other"),
    "username": (string),
    "password": (string),
    "country_id": (int),
    "italian_municipality_id": (int|null),
    "cap": (string),
    "street_address": (string),
    "house_number": (string),
    "locality": (string|null),
    "additional_info": (string|null)
    "state_id": (int),
    "state_until": (timestamp)
}

```

Parametri:

Query String:

Errori Possibili

403 Forbidden – L'utente non è autorizzato ad eseguire l'endpoint

500 Internal Server Error – Errore del Server

422 Unprocessable Entity – Errore nella validazione di uno dei campi

Stato della risposta:

201 Created – Risorsa Creata con successo

Risposta:

Admin:

```
{
  "message": "User Successfully Created"
  "data": {
    "id": (int)
    "username": (string),
    "roles": [
      (string),
      (string),
      ... (per ogni ruolo)
    ]
    "state_id": (int),
    "state_until": (timestamp|null),
    "created_at": (timestamp|null),
    "updated_at": (timestamp|null),
    "deleted_at": (timestamp|null)
  }
}
```

Modifica Utente

Utenti con Privilegio: Admin

Operazione: modifica parziale o totale di una risorsa

Formato Richiesta: JSON

Formato Risposta: JSON

Descrizione:

Richiede al server di modificare alcune (o tutte) le informazioni dell'utente non sensibili con quelle inviate nel Payload. Solo le informazioni inviate sono modificate

Le informazioni aggiornate non sensibili sono restituite

URL: <https://esempio.it/api/v1/users/{userId}>

Metodo HTTP: PUT

Payload:

```
{
  "username": (string),
  "role": (string, il ruolo più alto da assegnare)
```



```
}
```

Parametri

userId (int): l'ID dell'utente richiesto

Query String:

Errori Possibili

403 Forbidden – L'utente non è autorizzato ad eseguire l'endpoint

404 Not Found – L'utente indicato non è stato trovato

422 Unprocessable Entity – Errore nella validazione di uno dei campi

500 Internal Server Error – Errore del Server

Stato della risposta:

200 OK – Risorsa Modificata con successo

Risposta:

Admin:

```
{
  "message": "Film Successfully Created"
  "data": {
    "id": (int)
    "username": (string),
    "roles": [
      (string),
      (string),
      ... (per ogni ruolo)
    ]
    "state_id": (int),
    "state_until": (timestamp|null),
    "created_at": (timestamp|null),
    "updated_at": (timestamp|null),
    "deleted_at": (timestamp|null)
  }
}
```

Eliminazione Utente

Utenti con Privilegio: Admin

Operazione: eliminazione risorsa

Formato Richiesta: nessun payload

Formato Risposta: nessuna risposta

Descrizione:

Elimina l'utente indicato (in maniera reversibile).

URL: <https://esempio.it/api/v1/users/{userId}>

Metodo HTTP: DELETE

Payload:**Parametri:**

userId (int): l'ID dell'utente richiesto

Query String:**Errori Possibili**

403 Forbidden – L'utente non è autorizzato ad eseguire l'endpoint

409 Conflict – L'utente è già stato eliminato

404 Not Found– L'utente Indicato non è stato trovato

500 Internal Server Error – Errore del Server

Stato della risposta:

204 No Content – Risorsa Eliminata con successo

Risposta:**Admin:****Recupero Utente**

Utenti con Privilegio: Admin

Operazione: recupero risorsa

Formato Richiesta: nessun payload

Formato Risposta: JSON

Descrizione:

Recupera l'episodio indicato e ne ritorna le informazioni.

URL: <https://esempio.it/api/v1/users/{userId}/restore>

Metodo HTTP: PUT

Payload:**Parametri:**

userId (int): l'ID dell'utente richiesto

Query String:**Errori Possibili**

403 Forbidden – L'utente non è autorizzato ad eseguire l'endpoint

409 Conflict – L'utente non è eliminato

404 Not Found– L'utente indicato non è stato trovato

500 Internal Server Error – Errore del Server

Stato della risposta:

200 OK– Risorsa Recuperata con successo

Risposta:**Admin:**

```
{
  "message": "User Successfully Restored"
  "data": {
    "id": (int)
    "username": (string),
    "roles": [
      (string),
      (string),
      ... (per ogni ruolo)
    ]
    "state_id": (int),
    "state_until": (timestamp|null),
    "created_at": (timestamp|null),
    "updated_at": (timestamp|null),
    "deleted_at": (timestamp|null)
  }
}
```

Eliminazione Definitiva Utente

Utenti con Privilegio: Admin

Operazione: eliminazione risorsa definitiva

Formato Richiesta: nessun payload

Formato Risposta: nessuna risposta

Descrizione:

Cancella l'utente indicato definitivamente, previa eliminazione "soft" (vedi Eliminazione Utente). Tutte le informazioni associate sono cancellate e irrecuperabili

URL: <https://esempio.it/api/v1/users/{userId}/force>

Metodo HTTP: DELETE

Payload:

Parametri:

userId (int): l'ID dell'utente richiesto

Query String:

Errori Possibili

403 Forbidden – L'utente non è autorizzato ad eseguire l'endpoint

409 Conflict – L'utente indicato non è eliminato

404 Not Found – L'utente indicato non è stato trovato

500 Internal Server Error – Errore del Server

Stato della risposta:

204 No Content – Risorsa Eliminata Definitivamente con successo

Risposta:

Admin:

Ban Utente

Utenti con Privilegio: Admin

Operazione: modifica stato risorsa

Formato Richiesta: nessun payload

Formato Risposta: nessuna risposta

Descrizione:

Imposta lo stato dell'utente a "banned" (bannato)

URL: <https://esempio.it/api/v1/users/{userId}/ban>

Metodo HTTP: PUT

Payload:

Parametri:

userId (int): l'ID dell'utente richiesto

Query String:

Errori Possibili

403 Forbidden – L'utente non è autorizzato ad eseguire l'endpoint

409 Conflict – L'utente indicato è già bannato

404 Not Found – L'utente indicato non è stato trovato

500 Internal Server Error – Errore del Server

Stato della risposta:

200 OK – Utente Bannato con successo

Risposta:

Admin:

```
{
  "message": "User {username dell'utente} has been Banned!"
}
```

Sospensione Utente

Utenti con Privilegio: Admin

Operazione: modifica stato risorsa

Formato Richiesta: JSON

Formato Risposta: nessuna risposta

Descrizione:

Imposta lo stato dell'utente a "suspended" (sospeso) fino alla data passata nel payload

URL: <https://esempio.it/api/v1/users/{userId}/suspend>

Metodo HTTP: PUT

Payload:

```
{  
    "state_until"=> (datetime),  
}
```

Parametri:

userId (int): l'ID dell'utente richiesto

Query String:**Errori Possibili**

403 Forbidden – L'utente non è autorizzato ad eseguire l'endpoint

409 Conflict – L'utente indicato è già sospeso

404 Not Found– L'utente indicato non è stato trovato

500 Internal Server Error – Errore del Server

Stato della risposta:

200 OK– Utente Sospeso con successo

Risposta:**Admin:**

```
{  
    "message": "User {username dell'utente} has been Suspended until {data di scadenza della  
sospensione}!"  
}
```

Blocco Utente

Utenti con Privilegio: Admin

Operazione: modifica stato risorsa

Formato Richiesta: JSON (payload opzionale)

Formato Risposta: nessuna risposta

Descrizione:

Imposta lo stato dell'utente a "locked" (bloccato) in modo indefinito oppure fino alla data di scadenza passata nel payload (opzionale)

URL: <https://esempio.it/api/v1/users/{userId}/suspend>

Metodo HTTP: PUT

Payload:

```
{  
    "state_until"=> (datetime),  
}
```

Parametri:

userId (int): l'ID dell'utente richiesto

Query String:

Errori Possibili

403 Forbidden – L'utente non è autorizzato ad eseguire l'endpoint

409 Conflict – L'utente indicato è già bloccato

404 Not Found– L'utente indicato non è stato trovato

500 Internal Server Error – Errore del Server

Stato della risposta:

200 OK– Utente Bloccato con successo

Risposta:

Admin, blocco senza scadenza:

```
{
  "message": "User {username dell'utente} has been Locked!"
}
```

Admin, blocco con scadenza:

```
{
  "message": "User {username dell'utente} has been Locked until {data di scadenza del blocco}!"
}
```

Attivazione Utente

Utenti con Privilegio: Admin

Operazione: modifica stato risorsa

Formato Richiesta: nessun payload

Formato Risposta: nessuna risposta

Descrizione:

Imposta lo stato dell'utente a "active" (attivo), ovvero senza alcuna limitazione di stato

URL: <https://esempio.it/api/v1/users/{userId}/activate>

Metodo HTTP: PUT

Payload:

Parametri:

userId (int): l'ID dell'utente richiesto

Query String:

Errori Possibili

403 Forbidden – L'utente non è autorizzato ad eseguire l'endpoint

409 Conflict – L'utente indicato è già attivo

404 Not Found– L'utente indicato non è stato trovato

500 Internal Server Error – Errore del Server

Stato della risposta:

200 OK– Utente Attivato con successo

Risposta:

Admin:

```
{  
    "message": "User {username dell'utente} has been Activated!"  
}
```

Profilo

Accesso Profilo

Utenti con Privilegio: Utente/Cliente, Admin

Operazione: accesso di una risorsa

Formato Richiesta: nessun payload

Formato Risposta: JSON

Descrizione:

Richiede al server le informazioni del proprio profilo.

URL: <https://esempio.it/api/v1/me>

Metodo HTTP: GET

Payload:

Parametri:

Query String:

Errori Possibili

500 Internal Server Error – Errore del Server

Stato della risposta:

200 Ok– Risorsa Ricevuta con successo

Risposta:

Admin/Cliente:

```
{
  "message": "Profile Successfully Retrieved"
  "data": {
    "username": (string),
    "name": (string),
    "surname": (string),
    "email": (string),
    "birthdate": (date),
    "gender": ("male"|"female"|"other"),
    "state": (string),
    "state_until": (timestamp),
    "created_at": (timestamp),
    "last_updated": (timestamp),
    "address": [
      "country": (string),
      "municipality": (string|null),
      "cap": (string),
      "street_address": (string),
```



```
        "house_number": (string),
        "locality": (string|null),
        "additional_info": (string|null)
    ]
}
```

Modifica Profilo

Utenti con Privilegio: Utente/Cliente, Admin

Operazione: modifica totale o parziale di una risorsa

Formato Richiesta: JSON

Formato Risposta: JSON

Descrizione:

Richiede al server di modificare con i dati inviati nel Payload quelli del proprio profilo. Sono modificati solamente i campi inviati, mentre quelli non menzionati nella richiesta vengono mantenuti.

URL: <https://esempio.it/api/v1/me>

Metodo HTTP: PUT

Payload:

```
{
    "name": (string),
    "surname": (string),
    "email": (string),
    "birthdate": (date),
    "gender": ("male"|"female"|"other"),
    "username": (string),
    "password": (string),
    "country_id": (int),
    "italian_municipality_id": (int|null),
    "cap": (string),
    "street_address": (string),
    "house_number": (string),
    "locality": (string|null),
    "additional_info": (string|null)
}
```

Parametri:

Query String:

Errori Possibili

422 Unprocessable Entity – Errore nella validazione di uno dei campi

500 Internal Server Error – Errore del Server

Stato della risposta:

200 Ok– Risorsa Aggiornata con successo

Risposta:**Admin/Cliente:**

```
{
  "message": "Profile Successfully Updated"
  "data": {
    "username": (string),
    "name": (string),
    "surname": (string),
    "email": (string),
    "birthdate": (date),
    "gender": ("male"|"female"|"other"),
    "state": (string),
    "state_until": (timestamp),
    "created_at": (timestamp),
    "last_updated": (timestamp),
    "address": [
      "country": (string),
      "municipality": (string|null),
      "cap": (string),
      "street_address": (string),
      "house_number": (string),
      "locality": (string|null),
      "additional_info": (string|null)
    ]
  }
}
```

Eliminazione Profilo

Utenti con Privilegio: Utente/Cliente, Admin

Operazione: eliminazione risorsa

Formato Richiesta: nessun payload

Formato Risposta: nessuna risposta

Descrizione:

Richiede al server di eliminare il proprio account.

L'eliminazione non corrisponde a una rimozione immediata delle informazioni: l'account può quindi essere recuperato tramite apposita richiesta da parte di un utente admin (vedi Recupero Utente)

URL: <https://esempio.it/api/v1/me>

Metodo HTTP: DELETE

Payload:

Parametri:

Query String:

Errori Possibili

500 Internal Server Error – Errore del Server

Stato della risposta:

204 No Content – Risorsa Eliminata con successo

Risposta:

Admin/Cliente:

Pagamenti

Accesso Credito

Utenti con Privilegio: Utente/Cliente

Operazione: accesso di una risorsa

Formato Richiesta: nessun payload

Formato Risposta: JSON

Descrizione:

Richiede al server il credito associato al proprio profilo.

URL: <https://esempio.it/api/v1/me/credit>

Metodo HTTP: GET

Payload:

Parametri:

Query String:

Errori Possibili

500 Internal Server Error – Errore del Server

Stato della risposta:

200 Ok– Risorsa Ricevuta con successo

Risposta:

Cliente:

```
{
  "message": "Your Total Credit: {credito dell'utente}"
}
```

Aggiunta Credito

Utenti con Privilegio: Utente/Cliente

Operazione: accesso di una risorsa

Formato Richiesta: JSON

Formato Risposta: JSON

Descrizione:

Richiede al server di aggiungere il credito specificato a quello associato al proprio profilo.

URL: <https://esempio.it/api/v1/me/credit/add>

Metodo HTTP: PUT

Payload:

```
{  
    "credit": (numeric > 0)  
}
```

Parametri:

Query String:

Errori Possibili

422 Unprocessable Entity – Errore nella validazione di uno dei campi

500 Internal Server Error – Errore del Server

Stato della risposta:

200 Ok– Risorsa Modificata con successo

Risposta:

Cliente:

```
{  
    "message": "Credit Added! Your Credit is now {credito aggiornato dell'utente}"  
}
```

Rimozione Credito

Utenti con Privilegio: Utente/Cliente

Operazione: accesso di una risorsa

Formato Richiesta: JSON

Formato Risposta: JSON

Descrizione:

Richiede al server di rimuovere il credito specificato a quello associato al proprio profilo.

URL: <https://esempio.it/api/v1/me/credit/remove>

Metodo HTTP: PUT

Payload:

```
{  
    "credit": (numeric, > 0 e <= credito attuale)  
}
```

Parametri:

Query String:

Errori Possibili

422 Unprocessable Entity – Errore nella validazione di uno dei campi

500 Internal Server Error – Errore del Server

Stato della risposta:

200 Ok– Risorsa Modificata con successo

Risposta:

Cliente:

```
{  
  "message": "Credit Removed! Your Credit is now {credito aggiornato dell'utente}"  
}
```

Note Finali

La Documentazione precedente è stata redatta per l'Esame della Quarta Sessione del Digital Campus Code a cura di Cringy.

Tutti i diritti sono Riservati.

© 2025, Cringy