

Documento Integrazione API

[GEFOS] Web Service

API – Gestione Corsi

Versione del Documento: **v1.9**

Data revisione: **16/07/2026**

Cronologia delle Revisioni

Versione	Data	Sintesi delle Modifiche
1.8	16/07/2026	Rinominato Lezione in Calendario. Aggiornata sezione Richiesta Accesso API Manager. Introdotta conferma tardiva (confermaTardiva) e vincolo dei 30 giorni sulla conferma standard. Aggiunto metodo Mista e dizionario Metodo Svolgimento Giornata; ristrutturato Calendario (dataOraDa/dataOraA, link/descrizione, metodo giornata). Limite 30 partecipanti per Presenza/Mista. Calendari obbligatori in creazione edizione e aggiornamento massivo (PUT ../calendari); rimossi POST/DELETE slot singoli e DELETE edizione. Vincolo slot rispetto alla data odierna e controllo generico sul monte ore tipologico (Presenza/Video-conferenza). Edizione: dataOralInizio/dataOraFine al posto di dataInizio/dataFine.
1.7	22/06/2026	Introdotta modello Edizione: il Corso contiene la tipologia; metodo di svolgimento, partecipanti e calendario (con dataInizio/dataFine) sono associati all'Edizione. Introdotta API per Conferma conclusione corso. Adattamenti vari.
1.6.2	19/05/2026	Aggiunto campo provNascita (str) all'oggetto Partecipante
1.6.1	18/05/2026	Calendario obbligatorio in creazione corso; slotTemporalI opzionale in oggetto Calendario
1.6	07/05/2026	Le operazioni che prima erano consentite solo entro la conferma del corso, ora sono effettuabili fino alla data di fine corso
1.5	06/05/2026	Header obbligatorio X-Gefos-Ente-Erogatore per tutte le API Gestione Corsi
1.4	05/05/2026	Aggiunti WS per Attestato
1.3	28/04/2026	Aggiunti WS per Tipologia Corso
1.2	24/04/2026	Aggiornamento documento
1.1	23/04/2026	Aggiornamento documento
1.0	10/04/2026	Prima emissione

Limiti di utilizzo del documento

In base alla classificazione del documento.

Indice

1	Introduzione	6
1.1	Scopo	6
1.2	Contatti	6
1.3	Links	6
1.4	Acronimi e definizioni	6
2	Richiesta Accesso API Manager	7
2.1	Richiesta Dati (Azione richiesta)	7
2.2	Processo Operativo per l'Ente Formatore / Software	7
2.3	Accesso tramite Piattaforma Web	7
3	API Gestione Corsi	8
3.0.1	Modello Corso ed Edizione	8
3.0.2	Regole trasversali	8
3.1	Recupero elenco Tipologie Corso	9
3.1.1	Parametri	10
3.1.2	Risposta	10
3.1.3	Esempi	10
3.2	Recupero Corsi per Tipologia	10
3.2.1	Parametri	11
3.2.2	Risposta	11
3.2.3	Esempi	11
3.3	Recupero Corso	11
3.3.1	Parametri	12
3.3.2	Esempi	12
3.4	Creazione nuovo corso	13
3.4.1	Parametri	13
3.4.2	Esempi	14
3.5	Modifica Corso	14
3.5.1	Parametri	14
3.5.2	Esempi	15
3.6	Aggiunta Edizione	15
3.6.1	Parametri	16
3.6.2	Esempi	16
3.7	Modifica Edizione	18
3.7.1	Parametri	18
3.7.2	Esempi	18
3.8	Aggiunta Partecipante ad un'edizione	19
3.8.1	Parametri	19
3.8.2	Esempi	19
3.9	Rimozione Partecipante	20
3.9.1	Parametri	20
3.9.2	Esempi	20
3.10	Aggiornamento Calendari	21
3.10.1	Parametri	21

3.10.2 Esempi.....	21
3.11 Inserimento Esito.....	24
3.11.1 Parametri	24
3.11.2 Esempi.....	24
3.12 Aggiornamento Esito	25
3.12.1 Parametri.....	25
3.12.2 Esempi.....	25
3.13 Inserimento Esiti bulk	26
3.13.1 Parametri.....	26
3.13.2 Esempi.....	27
3.14 Conferma conclusione edizione.....	27
3.14.1 Parametri.....	27
3.14.2 Esempi.....	28
3.15 Conferma conclusione tardiva.....	29
3.15.1 Parametri.....	29
3.15.2 Esempi.....	29
3.16 Download Attestato Partecipante.....	31
3.16.1 Parametri.....	31
3.16.2 Esempi.....	31
3.17 Download Attestati Tutti i Partecipanti.....	32
3.17.1 Parametri.....	32
3.17.2 Esempi.....	32
4 Oggetti	34
4.1 Corso	34
4.2 Corso leggero.....	34
4.3 Edizione richiesta	34
4.4 Edizione	35
4.5 Tipologia Corso.....	36
4.6 Dettaglio Partecipante.....	37
4.7 Calendario	37
5 Autenticazione e Accesso ad API Manager.....	39
5.1 Overview	39
5.2 Richiesta dell'Access Token.....	39
5.2.1 Esempio Chiamata CURL.....	39
5.2.2 Esempio Risposta Successo.....	39
5.3 Utilizzo del Token	40
5.3.1 Esempio di chiamata autorizzata.....	40
6 Dizionario dei valori codificati.....	41
6.1 Metodo Svolgimento	41
6.2 Metodo Svolgimento Giornata.....	41
6.3 Esito.....	41

Indice delle Tabelle

Tabella 1 Contatti.....	6
Tabella 2 Link.....	6
Tabella 3 Acronimi e definizioni.....	6
Tabella 4 Descrizione campi per la creazione di un nuovo corso	13
Tabella 5 Descrizione campi per la modifica di un corso.....	15
Tabella 6 Descrizione campi per la creazione di una nuova edizione	16
Tabella 7 Descrizione campi per la modifica di un'edizione	18
Tabella 8 Campi per inserimento Esito	24
Tabella 9 Campi per aggiornamento Esito	25
Tabella 10 Campi per inserimento Esiti in blocco	26
Tabella 11 Body conferma conclusione tardiva	29
Tabella 12 Oggetto Corso	34
Tabella 13 Oggetto Corso leggero.....	34
Tabella 14 Oggetto Edizione richiesta	35
Tabella 15 Oggetto Edizione	36
Tabella 16 Oggetto Tipologia Corso.....	36
Tabella 17 Oggetto Dettaglio Partecipante.....	37
Tabella 18 Oggetto Calendario	38
Tabella 19 Specifiche chiamata per ottenimento Token	39
Tabella 20 Metodo Svolgimento.....	41
Tabella 21 Metodo Svolgimento Giornata	41
Tabella 22 Possibili esiti del corso.....	41

1 Introduzione

1.1 Scopo

Il presente documento si pone l'obiettivo di descrivere le specifiche di integrazione API previste per la gestione corsi di GEFOS.

1.2 Contatti

Tipo	Contatto
E-mail	supporto.gefos@ariaspa.it

Tabella 1: Contatti

1.3 Links

	Link
Base_URL	<URL_servizio>/gefes/extapi
Swagger	<URL_servizio>/gefes/extapi/swagger-ui.htm
NOTA	A breve seguiranno gli indirizzi completi degli ambienti di collaudo e produzione.

Tabella 2: Link

1.4 Acronimi e definizioni

Acronimo	Descrizione
GEFOS	Gestione Formazione Sicurezza

Tabella 3: Acronimi e definizioni

2 Richiesta Accesso API Manager

Al fine di consentire l'utilizzo dei servizi in Cooperazione Applicativa e l'evasione dei relativi processi ARIA (API Manager), è necessario procedere al censimento preliminare degli Enti Formatori. Questo passaggio è obbligatorio per permettere ai referenti degli enti di accedere al Portale Key Manager-Welfare, censire la propria applicazione, sottoscrivere il servizio e generare le credenziali API Manager.

2.1 Richiesta Dati (Azione richiesta)

Per avviare la procedura, è necessario trasmettere l'elenco degli Enti e dei relativi referenti tecnici, completi dei seguenti dati:

- Nome e Cognome
- Codice Fiscale
- Indirizzo Email
- Numero di Cellulare

Nota: Una volta completato il censimento nei nostri sistemi, il referente tecnico sarà abilitato a richiedere e gestire le credenziali di Cooperazione Applicativa in totale autonomia.

2.2 Processo Operativo per l'Ente Formatore / Software

Dopo la conferma di avvenuto censimento, il referente tecnico dell'Ente o della Software House dovrà svolgere le seguenti attività interne per gestire la sottoscrizione dei servizi:

1. **Accesso:** Autenticarsi tramite SPID o CIE sul Portale Key Manager-Welfare.
2. **Selezione Organizzazione:** Scegliere l'Organizzazione per cui si opera (selezionando il proprio Ente o la Software House).
3. **Censimento Applicazione:** Procedere al censimento della propria applicazione all'interno del portale.
4. **Sottoscrizione:** Sottoscrivere il servizio/applicazione «GEFOS».
5. **Generazione Credenziali:** Creare le credenziali API MANAGER, selezionando l'ambiente desiderato (PRODUZIONE o SANDBOX).

2.3 Accesso tramite Piattaforma Web

Per gli utenti che, invece, non utilizzeranno la cooperazione applicativa ma dovranno accedere al servizio direttamente tramite interfaccia web, non è necessaria la procedura sopra indicata. A questi utenti verrà inviato successivamente l'indirizzo URL diretto per accedere all'applicativo.

3 API Gestione Corsi

Questo capitolo presenta le API REST per la gestione dei corsi di formazione in GEFOS. Per ciascun servizio sono descritti endpoint, parametri, strutture dati e risultati attesi, con esempi di integrazione.

3.0.1 Modello Corso ed Edizione

Un **Corso** rappresenta le informazioni generiche del percorso formativo (tipologia). A ciascun corso possono essere associate una o più **Edizioni**, ciascuna con il proprio metodo di svolgimento, date (`dataOraInizio`, `dataOraFine`), elenco partecipanti e calendari.

3.0.2 Regole trasversali

Finestra di modifica edizione: le operazioni di modifica su un'edizione (dati edizione, partecipanti, calendari, esiti) sono consentite fino alla **conferma conclusione edizione** (vedi POST `.../conferma` e POST `.../confermaTardiva`). Dopo la conferma, l'operazione è irreversibile e le richieste di modifica su **quella** edizione restituiscono errore `EDIZIONE_NON_MODIFICABILE`; le altre edizioni dello stesso corso restano modificabili se non confermate.

Riferimento normativo: la normativa di settore prevede un termine di **30 giorni** dalla data di `dataOraFine` dell'edizione per la comunicazione dei dati formativi. Trascorso tale termine, la conferma conclusione non è più consentita tramite POST `.../conferma`: è necessario utilizzare l'API dedicata POST `.../confermaTardiva` (vedi sezioni relative).

Obbligatorietà campi: nella colonna Obbligatorio delle tabelle oggetto, il valore **si** si riferisce alle operazioni di **creazione** (POST); **cond.** indica obbligatorietà condizionata (vedi note del campo). Nelle operazioni di **modifica** (PUT) tutti i campi ammessi dal contratto dell'endpoint devono essere sempre presenti nel body: non è consentito l'aggiornamento parziale. Per modificare partecipanti o esiti utilizzare le API dedicate; per i calendari vedi sotto.

Gestione calendari: in **creazione edizione** (POST `.../edizione`) l'array `calendari` è obbligatorio (almeno un elemento). Le successive modifiche al calendario avvengono **solo in modo massivo** tramite PUT `.../calendari`, inviando l'intera lista di slot. Non sono disponibili operazioni di aggiunta o rimozione di singoli slot.

Vincolo data odierna (calendari): confrontando la parte data di `dataOraDa` con la data odierna (fuso applicativo del servizio):

- non è consentito **aggiungere** slot con data precedente a oggi;
- non è consentito **modificare** né **eliminare** slot con data precedente a oggi.

Nel PUT massivo, gli slot già esistenti con data precedente a oggi devono essere **reinviiati invariati**; non possono essere omessi né alterati. Solo gli slot con data maggiore o uguale a oggi possono essere aggiunti, modificati o rimossi rispetto alla lista precedente.

Controllo monte ore tipologico: per le edizioni con metodo **Presenza Fisica** o **Video-Conferenza**, e solo per le tipologie **soggette al controllo sul monte ore**, la durata complessiva del calendario dell'edizione deve essere **coerente** (maggiore o uguale) rispetto al monte ore previsto dalla tipologia del corso. Il controllo non si applica a E-Learning e Mista. (Vedi errori `CALENDARIO_MONTE_ORE_INSUFFICIENTE` nelle API di creazione edizione e aggiornamento calendari.)

Metodo effettivo del calendario: l'obbligatorietà dei campi condizionali di ogni slot dipende dal metodo di svolgimento **effettivo** della giornata (vedi Sezione 4.7, Sezione 6.1 e Sezione 6.2).

Risposte di successo senza body: le operazioni che completano con successo ma non restituiscono dati nel body rispondono con codice HTTP `200 OK` (aggiornamenti), `201 Created` (creazioni) o `204 No Content` (eliminazioni), senza payload. Il codice HTTP è sufficiente a attestare l'esito positivo; non viene restituito un body di riepilogo.

Tutte le API descritte in questo capitolo richiedono l'header HTTP **obbligatorio** `X-Gefos-Ente-Erogatore`. Il valore deve essere il **codice fiscale** o la **partita IVA** dell'ente erogatore del corso (l'organizzazione che invoca il servizio in tale qualifica).

Errore: ente non censito sull'applicativo (400 Bad Request):

```
{
  "title": "Bad Request",
  "status": 400,
  "code": "ENTE_NON_CENSITO",
  "detail": "Ente non censito sull'applicativo"
}
```

Errore: ente non autorizzato sulla risorsa richiesta (403 Forbidden):

```
{
  "title": "Forbidden",
  "status": 403,
  "code": "ENTE_NON_AUTORIZZATO",
  "detail": "Ente non autorizzato ad eseguire questa operazione"
}
```

Errore: edizione non modificabile (422 Unprocessable Entity):

```
{
  "title": "Unprocessable Entity",
  "status": 422,
  "code": "EDIZIONE_NON_MODIFICABILE",
  "detail": "L'edizione non è più modificabile"
}
```

3.1 Recupero elenco Tipologie Corso

Restituisce l'elenco delle tipologie corso disponibili, ciascuna con i dati della denominazione associata (ogni tipologia è associata al massimo a una denominazione).

3.1.1 Parametri

Metodo	ENDPOINT
GET	/tipologieCorso

Nessun parametro richiesto nel path né nella query string.

3.1.2 Risposta

In caso di successo (200 OK) il body è un array JSON di oggetti **Tipologia Corso** (struttura nella Sezione 4.5).

3.1.3 Esempi

Esempio di chiamata CURL:

```
curl -X GET "<Base_URL>/tipologieCorso" \  
-H "X-Gefos-Ente-Erogatore: 02438750586" \  
-H "accept: application/json"
```

Esempio risposta in caso di successo (200 OK):

```
[  
  {  
    "id": 1,  
    "descrizione": "CORSO DI FORMAZIONE GENERALE PER LAVORATORI EX ART. 37 COMMA 1 DEL D.LGS  
81/2008 - ASR 59/2025 PARTE II PUNTO 2.1",  
    "idDenominazione": 1,  
    "descrDenominazione": "CORSO DI FORMAZIONE PER LAVORATORI, DIRIGENTI O PREPOSTI EX ART.37  
DEL D.LGS. 81/2008"  
  },  
  {  
    "id": 2,  
    "descrizione": "...",  
    "idDenominazione": 1,  
    "descrDenominazione": "CORSO DI FORMAZIONE PER LAVORATORI, DIRIGENTI O PREPOSTI EX ART.37  
DEL D.LGS. 81/2008"  
  }  
]
```

3.2 Recupero Corsi per Tipologia

Restituisce l'elenco dei corsi dell'ente erogatore (identificato dall'header `X-Gefos-Ente-Erogatore`) associati alla tipologia corso indicata.

3.2.1 Parametri

Metodo	ENDPOINT
GET	/corsi/tipologiaCorsoId/{tipologiaCorsoId}

Path: `tipologiaCorsoId` — identificativo della tipologia corso (come restituito da GET / `tipologieCorso`).

3.2.2 Risposta

In caso di successo (200 OK) il body è un array JSON di oggetti **Corso leggero** (struttura nella Sezione 4.2).

3.2.3 Esempi

Esempio di chiamata CURL:

```
curl -X GET "<Base_URL>/corsi/tipologiaCorsoId/1" \  
-H "X-Gefos-Ente-Erogatore: 02438750586" \  
-H "accept: application/json"
```

Esempio risposta in caso di successo (200 OK):

```
[  
  {  
    "id": 12345,  
    "tipologiaCorsoId": 1  
  },  
  {  
    "id": 12346,  
    "tipologiaCorsoId": 1  
  }  
]
```

Esempio risposta con errori: Tipologia non valida

```
{  
  "title": "Unprocessable Entity",  
  "status": 422,  
  "code": "INVALID_TIPOLOGIA",  
  "detail": "tipologiaCorsoId non valido"  
}
```

3.3 Recupero Corso

Restituisce il dettaglio completo di un corso, incluse tutte le edizioni associate con i relativi dati (metodo di svolgimento, date, partecipanti, calendari ed esiti).

3.3.1 Parametri

Metodo	ENDPOINT
GET	/corso/{corsoid}

Nessun body. In caso di successo (200 OK) il body è un oggetto **Corso** (struttura nella Sezione 4.1) con l'elenco completo delle edizioni (oggetto **Edizione** nella Sezione 4.4, con `id` valorizzato).

3.3.2 Esempi

Esempio di chiamata CURL:

```
curl -X GET "<Base_URL>/corso/12345" \  
-H "X-Gefos-Ente-Erogatore: 02438750586" \  
-H "accept: application/json"
```

Esempio risposta in caso di successo (200 OK):

```
{  
  "id": 12345,  
  "tipologiaCorsoId": 1,  
  "edizioni": [  
    {  
      "id": 67890,  
      "metodoSvolgimentoId": 1,  
      "dataOraInizio": "2025-05-01T09:00",  
      "dataOraFine": "2025-05-01T13:00",  
      "partecipanti": [  
        {  
          "nome": "Mario",  
          "cognome": "Rossi",  
          "codiceFiscale": "RSSMRA80A01H501U",  
          "dataNascita": "1980-01-01",  
          "luogoNascita": "Roma",  
          "provNascita": "RM",  
          "email": "mario.rossi@example.com",  
          "esitoId": 1  
        }  
      ],  
      "calendari": [  
        {  
          "id": 10001,  
          "dataOraDa": "2025-05-01T09:00",  
          "dataOraA": "2025-05-01T13:00",  
          "indirizzo": "Via Roma",  
          "nCivico": "1",  
          "cap": "00100",  
          "citta": "Roma",  
          "provincia": "RM"  
        }  
      ]  
    }  
  ],  
}
```

```
"id": 67891,
"metodoSvolgimentoId": 2,
"dataOraInizio": "2025-05-02T00:00",
"dataOraFine": "2025-05-05T23:59",
"partecipanti": [
  {
    "nome": "Giulia",
    "cognome": "Verdi",
    "codiceFiscale": "VRDGPP85B02F205X",
    "dataNascita": "1985-02-02",
    "luogoNascita": "Milano",
    "provNascita": "MI",
    "email": "giulia.verdi@example.com"
  }
],
"calendari": [
  {
    "id": 10002,
    "dataOraDa": "2025-05-02T00:00",
    "dataOraA": "2025-05-05T23:59"
  }
]
}
```

3.4 Creazione nuovo corso

Questo servizio permette l'inserimento di un nuovo corso. Le edizioni vanno aggiunte successivamente tramite POST /corso/{corsold}/edizione (inclusi i calendari). I partecipanti si gestiscono tramite le API dedicate.

3.4.1 Parametri

Metodo	ENDPOINT
POST	/corso

I campi previsti nel file JSON per la creazione del corso sono i seguenti:

Campo	Tipo dato	Obbligatorio	Note
tipologiaCorsold	int	si	Id della tipologia del corso. Deve corrispondere a un elemento dell'elenco restituito da GET /tipologieCorso (vedi oggetto nella Sezione 4.5). Unico campo accettato.

Tabella 4: Descrizione campi per la creazione di un nuovo corso

3.4.2 Esempi

Esempio di chiamata CURL:

```
curl -X POST "<Base_URL>/corso" \  
-H "X-Gefos-Ente-Erogatore: 02438750586" \  
-H "accept: application/json" \  
-H "Content-Type: application/json" \  
-d '{  
  "tipologiaCorsoId": 1  
'
```

Esempio risposta in caso di successo (201 Created):

```
{  
  "id": 12345  
}
```

Esempio risposta con errori: Tipologia non valida

```
{  
  "title": "Unprocessable Entity",  
  "status": 422,  
  "code": "INVALID_TIPOLOGIA",  
  "detail": "tipologiaCorsoId non valido"  
}
```

3.5 Modifica Corso

Permette la modifica di un corso esistente tramite il suo identificativo univoco. È possibile aggiornare solo `tipologiaCorsoId` (modificabile solo se il corso non ha edizioni). Le edizioni si gestiscono tramite le API dedicate.

3.5.1 Parametri

Metodo	ENDPOINT
PUT	/corso/{corsoid}

I campi previsti nel file JSON per la modifica del corso sono i seguenti:

Campo	Tipo dato	Obbligatorio	Note
tipologiaCorsoId	int	si	Id della tipologia del corso. Modificabile solo se il corso non ha edizioni (vedi oggetto nella Sezione 4.5). Unico campo accettato.

Tabella 5: Descrizione campi per la modifica di un corso

3.5.2 Esempi

Esempio di chiamata CURL:

```
curl -X PUT "<Base_URL>/corso/12345" \  
-H "X-Gefos-Ente-Erogatore: 02438750586" \  
-H "accept: application/json" \  
-H "Content-Type: application/json" \  
-d '{  
  "tipologiaCorsoId": 2  
'
```

Risposta in caso di successo (204 No Content): Nessun body restituito.

Risposta con errore: Corso non trovato

```
{  
  "title": "Not Found",  
  "status": 404,  
  "code": "NOT_FOUND",  
  "detail": "Risorsa non trovata"  
}
```

Risposta con errore: Tipologia non modificabile

```
{  
  "title": "Unprocessable Entity",  
  "status": 422,  
  "code": "TIPOLOGIA_NON_MODIFICABILE",  
  "detail": "La tipologia del corso non può essere modificata in presenza di edizioni"  
}
```

3.6 Aggiunta Edizione

Permette l'aggiunta di una nuova edizione a un corso esistente. Il body deve includere l'intero calendario dell'edizione (array `calendari`, almeno un elemento). I partecipanti si gestiscono successivamente tramite le API dedicate. L'operazione è soggetta alle regole su data odierna, metodo effettivo e monte ore tipologico (vedi Sezione 3.0.2). In caso di successo viene restituito l'id della nuova edizione creata.

3.6.1 Parametri

Metodo	ENDPOINT
POST	/corso/{corsold}/edizione

I campi previsti nel file JSON sono i seguenti (vedi oggetto Edizione richiesta nella Sezione 4.3):

Campo	Tipo dato	Obbligatorio	Note
metodoSvolgimentold	int	si	Id del metodo di svolgimento (vedi Sezione 6.1).
dataOraInizio	str	si	Data e ora di inizio dell'edizione. Formato YYYY-MM-DDTHH:mm (es. 2025-06-01T09:00).
dataOraFine	str	si	Data e ora di fine dell'edizione. Formato YYYY-MM-DDTHH:mm (es. 2025-06-01T13:00). Non può essere precedente a <code>dataOraInizio</code> .
calendari	array	si	Elenco completo degli slot di calendario (vedi Sezione 4.7). Almeno un elemento. Non inviare il campo <code>id</code> degli slot.

Tabella 6: Descrizione campi per la creazione di una nuova edizione

3.6.2 Esempi

Esempio di chiamata CURL:

```
curl -X POST "<Base_URL>/corso/12345/edizione" \  
-H "X-Gefos-Ente-Erogatore: 02438750586" \  
-H "accept: application/json" \  
-H "Content-Type: application/json" \  
-d '{  
  "metodoSvolgimentoId": 1,  
  "dataOraInizio": "2025-06-01T09:00",  
  "dataOraFine": "2025-06-01T13:00",  
  "calendari": [  
    {  
      "dataOraDa": "2025-06-01T09:00",  
      "dataOraA": "2025-06-01T13:00",  
      "indirizzo": "Via Roma",  
      "nCivico": "1",  
      "cap": "00100",  
      "citta": "Roma",  
      "provincia": "RM"  
    }  
  ]  
'
```

Esempio risposta in caso di successo (201 Created): restituisce l'identificativo della nuova edizione creata.

```
{
  "id": 67892
}
```

Esempio risposta con errori: Metodo di svolgimento non valido

```
{
  "title": "Unprocessable Entity",
  "status": 422,
  "code": "INVALID_METODO_SVOLGIMENTO",
  "detail": "metodoSvolgimentoId non valido"
}
```

Esempio risposta errore: Calendario fuori dal range dell'edizione

```
{
  "title": "Unprocessable Entity",
  "status": 422,
  "code": "CALENDARIO_OUT_OF_RANGE",
  "detail": "dataOraDa e dataOraA devono essere comprese tra dataOraInizio e dataOraFine dell'edizione"
}
```

Esempio risposta errore: Slot con data precedente a oggi

```
{
  "title": "Unprocessable Entity",
  "status": 422,
  "code": "CALENDARIO_SLOT_PASSATO",
  "detail": "Non è consentito aggiungere slot di calendario con data precedente a oggi"
}
```

Esempio risposta errore: Monte ore tipologico insufficiente

```
{
  "title": "Unprocessable Entity",
  "status": 422,
  "code": "CALENDARIO_MONTE_ORE_INSUFFICIENTE",
  "detail": "La durata complessiva del calendario non è coerente con il monte ore previsto dalla tipologia del corso"
}
```

Esempio risposta errore: Indirizzo obbligatorio per metodo Presenza Fisica

```
{
  "title": "Unprocessable Entity",
  "status": 422,
  "code": "CALENDARIO_INDIRIZZO_OBBLIGATORIO",
  "detail": "Per il metodo effettivo Presenza Fisica i campi indirizzo del calendario sono obbligatori"
}
```

3.7 Modifica Edizione

Permette la modifica di un'edizione esistente. È possibile aggiornare solo `metodoSvolgimentoId`, `dataOraInizio` e `dataOraFine`; tutti e tre i campi devono essere sempre presenti nel body. I partecipanti si gestiscono tramite le API dedicate; i calendari solo tramite PUT `.../calendari`. L'operazione è soggetta alla regola sulla finestra di modifica edizione (vedi Sezione 3.0.2).

3.7.1 Parametri

Metodo	ENDPOINT
PUT	<code>/corso/{corsoid}/edizione/{edizioneid}</code>

I campi previsti nel file JSON sono i seguenti (vedi oggetto Edizione richiesta nella Sezione 4.3):

Campo	Tipo dato	Obbligatorio	Note
<code>metodoSvolgimentoId</code>	int	si	Id del metodo di svolgimento (vedi Sezione 6.1).
<code>dataOraInizio</code>	str	si	Data e ora di inizio dell'edizione. Formato YYYY-MM-DDTHH:mm (es. 2025-06-01T09:00).
<code>dataOraFine</code>	str	si	Data e ora di fine dell'edizione. Formato YYYY-MM-DDTHH:mm (es. 2025-06-01T13:00). Non può essere precedente a <code>dataOraInizio</code> .

Tabella 7: Descrizione campi per la modifica di un'edizione

3.7.2 Esempi

Esempio di chiamata CURL:

```
curl -X PUT "<Base_URL>/corso/12345/edizione/67890" \
-H "X-Gefos-Ente-Erogatore: 02438750586" \
-H "accept: application/json" \
-H "Content-Type: application/json" \
-d '{
```

```
"metodoSvolgimentoId": 3,
"dataOraInizio": "2025-05-01T09:00",
"dataOraFine": "2025-05-01T13:00"
}'
```

Risposta in caso di successo (204 No Content): Nessun body restituito.

Esempio risposta errore: Edizione non trovata

```
{
  "title": "Not Found",
  "status": 404,
  "code": "NOT_FOUND",
  "detail": "Risorsa non trovata"
}
```

3.8 Aggiunta Partecipante ad un'edizione

Permette l'inserimento di un nuovo partecipante in un'edizione già creata. **Limiti partecipanti per metodo di svolgimento:** in base al `metodoSvolgimentoId` dell'edizione (vedi Sezione 6.1):

- **Presenza Fisica (1) e Mista (4):** massimo **30** partecipanti; l'iscrizione del 31° viene rifiutata;
- **E-Learning (2):** massimo **1** partecipante per edizione.

3.8.1 Parametri

Metodo	ENDPOINT
POST	/corso/{corsold}/edizione/{edizioneld}/partecipante

PARAMETRI: Oggetto JSON Partecipante (vedi oggetto Partecipante nella Sezione 4.6).

3.8.2 Esempi

Esempio di chiamata CURL:

```
curl -X POST "<Base_URL>/corso/12345/edizione/67890/partecipante" \
-H "X-Gefos-Ente-Erogatore: 02438750586" \
-H "accept: application/json" \
-H "Content-Type: application/json" \
-d '{
  "nome": "Mario",
  "cognome": "Rossi",
  "codiceFiscale": "RSSMRA80A01H501U",
  "dataNascita": "1980-01-01",
  "luogoNascita": "Roma",
  "provNascita": "RM",
  "email": "mario.rossi@example.com"
}'
```

Risposta in caso di successo (201 Created): Nessun body restituito (vedi Sezione 3.0.2).

Esempio risposta errore: Partecipante già presente

```
{
  "title": "Conflict",
  "status": 409,
  "code": "PARTECIPANTE_DUPLICATO",
  "detail": "Il partecipante è già presente nell'edizione"
}
```

Esempio risposta errore: Limite partecipanti raggiunto (Presenza Fisica / Mista)

```
{
  "title": "Unprocessable Entity",
  "status": 422,
  "code": "MAX_PARTECIPANTI_SUPERATO",
  "detail": "Per i metodi Presenza Fisica e Mista è consentito un massimo di 30 partecipanti per edizione"
}
```

Esempio risposta errore: E-Learning con più partecipanti

```
{
  "title": "Unprocessable Entity",
  "status": 422,
  "code": "ELEARNING_MAX_PARTECIPANTI",
  "detail": "Per il metodo E-Learning è consentito al massimo un partecipante per edizione"
}
```

3.9 Rimozione Partecipante

Rimuove un partecipante dall'edizione specificata. L'operazione è soggetta alla regola sulla finestra di modifica edizione (vedi Sezione 3.0.2).

3.9.1 Parametri

Metodo	ENDPOINT
DELETE	/corso/{corsoId}/edizione/{edizioneId}/partecipante/{codiceFiscale}

3.9.2 Esempi

Esempio di chiamata CURL:

```
curl -X DELETE "<Base_URL>/corso/12345/edizione/67890/partecipante/RSSMRA80A01H501U" \
-H "X-Gefos-Ente-Erogatore: 02438750586" \
-H "accept: */*"
```

Risposta in caso di successo (204 No Content): Nessun body restituito (vedi Sezione 3.0.2).

Esempio risposta errore: Partecipante non presente

```
{
  "title": "Not Found",
  "status": 404,
  "code": "PARTECIPANTE_NON_PRESENTE",
  "detail": "Il partecipante non è presente nell'edizione"
}
```

3.10 Aggiornamento Calendari

Permette di aggiornare **in modo massivo** il calendario di un'edizione, inviando l'**intera** lista di slot. L'obbligatorietà dei campi condizionali dipende dal **metodo effettivo** della giornata (vedi Sezione 4.7). L'operazione è soggetta alle regole su finestra di modifica, data odierna e monte ore tipologico (vedi Sezione 3.0.2).

3.10.1 Parametri

Metodo	ENDPOINT
PUT	/corso/{corsold}/edizione/{edizioneld}/calendari

PARAMETRI: array JSON di oggetti Calendario (vedi Sezione 4.7). Deve contenere almeno un elemento. Per gli slot già esistenti con data precedente a oggi, reincludere il campo `id` e il contenuto invariato.

3.10.2 Esempi

Esempio di chiamata CURL:

```
curl -X PUT "<Base_URL>/corso/12345/edizione/67890/calendari" \
-H "X-Gefos-Ente-Erogatore: 02438750586" \
-H "accept: */*" \
-H "Content-Type: application/json" \
-d '[
  {
    "id": 10001,
    "dataOraDa": "2025-05-01T09:00",
    "dataOraA": "2025-05-01T13:00",
    "indirizzo": "Via Roma",
    "nCivico": "1",
```

```
    "cap": "00100",
    "citta": "Roma",
    "provincia": "RM"
  },
  {
    "dataOraDa": "2025-06-10T14:00",
    "dataOraA": "2025-06-10T18:00",
    "indirizzo": "Via Roma",
    "nCivico": "1",
    "cap": "00100",
    "citta": "Roma",
    "provincia": "RM"
  }
]
```

Risposta in caso di successo (200 OK): Nessun body restituito (vedi Sezione 3.0.2).

Esempio risposta errore: Edizione non trovata

```
{
  "title": "Not Found",
  "status": 404,
  "code": "NOT_FOUND",
  "detail": "Risorsa non trovata"
}
```

Esempio risposta errore: Calendario fuori dal range dell'edizione

```
{
  "title": "Unprocessable Entity",
  "status": 422,
  "code": "CALENDARIO_OUT_OF_RANGE",
  "detail": "dataOraDa e dataOraA devono essere comprese tra dataOraInizio e dataOraFine dell'edizione"
}
```

Esempio risposta errore: Slot passato non modificabile / non eliminabile

```
{
  "title": "Unprocessable Entity",
  "status": 422,
  "code": "CALENDARIO_SLOT_PASSATO",
  "detail": "Non è consentito aggiungere, modificare o eliminare slot di calendario con data precedente a oggi"
}
```

Esempio risposta errore: Monte ore tipologico insufficiente

```
{
  "title": "Unprocessable Entity",
  "status": 422,
  "code": "CALENDARIO_MONTE_ORE_INSUFFICIENTE",
  "detail": "La durata complessiva del calendario non è coerente con il monte ore previsto dalla tipologia del corso"
}
```

Esempio risposta errore: Indirizzo obbligatorio per metodo Presenza Fisica

```
{
  "title": "Unprocessable Entity",
  "status": 422,
  "code": "CALENDARIO_INDIRIZZO_OBBLIGATORIO",
  "detail": "Per il metodo effettivo Presenza Fisica i campi indirizzo del calendario sono obbligatori"
}
```

Esempio risposta errore: Link e descrizione obbligatori per Video-Conferenza

```
{
  "title": "Unprocessable Entity",
  "status": 422,
  "code": "CALENDARIO_LINK_OBBLIGATORIO",
  "detail": "Per il metodo effettivo Video-Conferenza i campi link e descrizione del calendario sono obbligatori"
}
```

Esempio risposta errore: Metodo giornata obbligatorio per edizione Mista

```
{
  "title": "Unprocessable Entity",
  "status": 422,
  "code": "CALENDARIO_METODO_GIORNATA_OBBLIGATORIO",
  "detail": "Per edizioni Mista il campo metodoSvolgimentoGiornataId è obbligatorio"
}
```

Esempio risposta errore: Metodo giornata non ammesso

```
{
  "title": "Unprocessable Entity",
  "status": 422,
  "code": "CALENDARIO_METODO_GIORNATA_NON_AMMESSO",
  "detail": "Il campo metodoSvolgimentoGiornataId non è ammesso per edizioni non Mista"
}
```

Esempio risposta errore: Metodo giornata non valido

```
{
  "title": "Unprocessable Entity",
  "status": 422,
  "code": "INVALID_METODO_SVOLGIMENTO_GIORNATA",
  "detail": "metodoSvolgimentoGiornataId non valido"
}
```

3.11 Inserimento Esito

Permette di registrare l'esito finale di un partecipante per una determinata edizione. L'operazione è soggetta alla regola sulla finestra di modifica edizione (vedi Sezione 3.0.2).

3.11.1 Parametri

Metodo	ENDPOINT
POST	/corso/{corsoid}/edizione/{edizioneld}/partecipante/esito

Campi del JSON:

Campo	Tipo dato	Obbligatorio	Note
codiceFiscale	str	si	Codice fiscale del partecipante.
esitoId	int	si	Id dell'esito (i possibili valori sono specificati nella Sezione 6.3).

Tabella 8: Campi per inserimento Esito

3.11.2 Esempi

Esempio di chiamata CURL:

```
curl -X POST "<Base_URL>/corso/12345/edizione/67890/partecipante/esito" \
-H "X-Gefos-Ente-Erogatore: 02438750586" \
-H "accept: application/json" \
-H "Content-Type: application/json" \
-d '{
  "codiceFiscale": "RSSMRA80A01H501U",
  "esitoId": 1
}'
```

Risposta in caso di successo (201 Created): Nessun body restituito (vedi Sezione 3.0.2).

Esempio risposta errore: Partecipante non presente

```
{
  "title": "Not Found",
  "status": 404,
  "code": "PARTECIPANTE_NON_PRESENTE",
  "detail": "Il partecipante non è presente nell'edizione"
}
```

Esempio risposta errore: Esito non valido

```
{
  "title": "Unprocessable Entity",
  "status": 422,
  "code": "ESITO_NON_VALIDO",
  "detail": "esitoId non previsto"
}
```

3.12 Aggiornamento Esito

Permette di modificare l'esito finale di un partecipante per una determinata edizione. L'operazione è soggetta alla regola sulla finestra di modifica edizione (vedi Sezione 3.0.2).

3.12.1 Parametri

Metodo	ENDPOINT
PUT	/corso/{corsold}/edizione/{edizioneld}/partecipante/esito

Campi del JSON:

Campo	Tipo dato	Obbligatorio	Note
codiceFiscale	str	si	Codice fiscale del partecipante.
esitoId	int	si	Id dell'esito (i possibili valori sono specificati nella Sezione 6.3).

Tabella 9: Campi per aggiornamento Esito

3.12.2 Esempi

Esempio di chiamata CURL:

```
curl -X PUT "<Base_URL>/corso/12345/edizione/67890/partecipante/esito" \
-H "X-Gefos-Ente-Erogatore: 02438750586" \
-H "accept: application/json" \
-H "Content-Type: application/json" \
-d '{
```

```
"codiceFiscale": "RSSMRA80A01H501U",  
"esitoId": 2  
}'
```

Risposta in caso di successo (200 OK): Nessun body restituito (vedi Sezione 3.0.2).

Esempio risposta errore: Partecipante non presente

```
{  
  "title": "Not Found",  
  "status": 404,  
  "code": "PARTECIPANTE_NON_PRESENTE",  
  "detail": "Il partecipante non è presente nell'edizione"  
}
```

Esempio risposta errore: Esito non valido

```
{  
  "title": "Unprocessable Entity",  
  "status": 422,  
  "code": "ESITO_NON_VALIDO",  
  "detail": "esitoId non previsto"  
}
```

3.13 Inserimento Esiti bulk

Permette di registrare gli esiti finali di più partecipanti per una determinata edizione in un'unica chiamata. L'operazione è soggetta alla regola sulla finestra di modifica edizione (vedi Sezione 3.0.2).

3.13.1 Parametri

Metodo	ENDPOINT
POST	/corso/{corsoid}/edizione/{edizioneId}/esiti

PARAMETRI: Array JSON di oggetti con i campi `codiceFiscale` e `esitoId` (vedi tabella sotto e Sezione 6.3).

Campo	Tipo dato	Obbligatorio	Note
codiceFiscale	str	si	Codice fiscale del partecipante.
esitoId	int	si	Id dell'esito (i possibili valori sono specificati nella Sezione 6.3).

Tabella 10: Campi per inserimento Esiti in blocco

3.13.2 Esempi

Esempio di chiamata CURL:

```
curl -X POST "<Base_URL>/corso/12345/edizione/67890/esiti" \
-H "X-Gefos-Ente-Erogatore: 02438750586" \
-H "accept: application/json" \
-H "Content-Type: application/json" \
-d '[
  {
    "codiceFiscale": "RSSMRA80A01H501U",
    "esitoId": 1
  },
  {
    "codiceFiscale": "VRDGPP85B02F205X",
    "esitoId": 2
  }
]'
```

Esempio risposta in caso di successo (201 Created): Nessun body restituito (vedi Sezione 3.0.2).

Esempio risposta errore: Partecipante non presente

```
{
  "title": "Not Found",
  "status": 404,
  "code": "PARTECIPANTE_NON_PRESENTE",
  "detail": "Uno o più partecipanti non sono presenti nell'edizione"
}
```

3.14 Conferma conclusione edizione

Consente di confermare la conclusione di un'edizione. L'operazione è **irreversibile**: dopo la conferma l'edizione non è più modificabile (vedi Sezione 3.0.2). La conferma è consentita anche prima della `dataOraFine` dell'edizione, ma **non oltre 30 giorni** dalla `dataOraFine`: trascorso tale termine è necessario utilizzare l'API di conferma tardiva (POST `.../confermaTardiva`).

Per poter confermare, l'edizione deve avere:

- almeno **un** partecipante iscritto;
- un **esito** valorizzato per **ogni** partecipante;
- almeno **un** calendario.

3.14.1 Parametri

Metodo	ENDPOINT
POST	/corso/{corsold}/edizione/{edizioneld}/conferma

Nessun body richiesto.

3.14.2 Esempi

Esempio di chiamata CURL:

```
curl -X POST "<Base_URL>/corso/12345/edizione/67890/conferma" \  
-H "X-Gefos-Ente-Erogatore: 02438750586" \  
-H "accept: */*"
```

Risposta in caso di successo (204 No Content): Nessun body restituito.

Esempio risposta errore: Edizione non trovata

```
{  
  "title": "Not Found",  
  "status": 404,  
  "code": "NOT_FOUND",  
  "detail": "Risorsa non trovata"  
}
```

Esempio risposta errore: Edizione già confermata

```
{  
  "title": "Unprocessable Entity",  
  "status": 422,  
  "code": "EDIZIONE_GIA_CONFERMATA",  
  "detail": "L'edizione è già stata confermata"  
}
```

Esempio risposta errore: Edizione senza partecipanti

```
{  
  "title": "Unprocessable Entity",  
  "status": 422,  
  "code": "EDIZIONE_SENZA PARTECIPANTI",  
  "detail": "L'edizione non può essere confermata: è necessario almeno un partecipante  
iscritto"  
}
```

Esempio risposta errore: Esiti incompleti

```
{  
  "title": "Unprocessable Entity",  
  "status": 422,  
  "code": "EDIZIONE_ESITI_INCOMPLETI",  
  "detail": "L'edizione non può essere confermata: ogni partecipante deve avere un esito  
valorizzato"  
}
```

Esempio risposta errore: Edizione senza calendari

```
{
  "title": "Unprocessable Entity",
  "status": 422,
  "code": "EDIZIONE_SENZA_CALENDARI",
  "detail": "L'edizione non può essere confermata: è necessario almeno un calendario"
}
```

Esempio risposta errore: Termine di 30 giorni scaduto

```
{
  "title": "Unprocessable Entity",
  "status": 422,
  "code": "CONFERMA_FUORI_TERMINE",
  "detail": "Sono trascorsi più di 30 giorni dalla dataOraFine dell'edizione: utilizzare l'API POST .../confermaTardiva"
}
```

3.15 Conferma conclusione tardiva

Consente di confermare la conclusione di un'edizione **oltre** il termine di **30 giorni** dalla `dataOraFine` (vedi Sezione 3.0.2). L'operazione è **irreversibile**: dopo la conferma l'edizione non è più modificabile. È obbligatorio indicare un giustificativo testuale.

Per poter confermare, l'edizione deve avere gli stessi requisiti previsti per la conferma standard:

- almeno **un** partecipante iscritto;
- un **esito** valorizzato per **ogni** partecipante;
- almeno **un** calendario.

3.15.1 Parametri

Metodo	ENDPOINT
POST	/corso/{corsold}/edizione/{edizioneId}/confermaTardiva

PARAMETRI: body JSON con i seguenti campi:

Campo	Tipo dato	Obbligatorio	Note
giustificativo	str	si	Motivazione testuale della conclusione tardiva.

Tabella 11: Body conferma conclusione tardiva

3.15.2 Esempi

Esempio di chiamata CURL:

```
curl -X POST "<Base_URL>/corso/12345/edizione/67890/confermaTardiva" \
-H "X-Gefos-Ente-Erogatore: 02438750586" \
-H "accept: */*" \
-H "Content-Type: application/json" \
-d '{
  "giustificativo": "Ritardo nella raccolta degli esiti dei partecipanti"
}'
```

Risposta in caso di successo (204 No Content): Nessun body restituito.

Esempio risposta errore: Edizione non trovata

```
{
  "title": "Not Found",
  "status": 404,
  "code": "NOT_FOUND",
  "detail": "Risorsa non trovata"
}
```

Esempio risposta errore: Edizione già confermata

```
{
  "title": "Unprocessable Entity",
  "status": 422,
  "code": "EDIZIONE_GIA_CONFERMATA",
  "detail": "L'edizione è già stata confermata"
}
```

Esempio risposta errore: Edizione senza partecipanti

```
{
  "title": "Unprocessable Entity",
  "status": 422,
  "code": "EDIZIONE_SENZA_PARTECIPANTI",
  "detail": "L'edizione non può essere confermata: è necessario almeno un partecipante iscritto"
}
```

Esempio risposta errore: Esiti incompleti

```
{
  "title": "Unprocessable Entity",
  "status": 422,
  "code": "EDIZIONE_ESITI_INCOMPLETI",
  "detail": "L'edizione non può essere confermata: ogni partecipante deve avere un esito valorizzato"
}
```

Esempio risposta errore: Edizione senza calendari

```
{
  "title": "Unprocessable Entity",
  "status": 422,
  "code": "EDIZIONE_SENZA_CALENDARI",
  "detail": "L'edizione non può essere confermata: è necessario almeno un calendario"
}
```

Esempio risposta errore: Giustificativo mancante

```
{
  "title": "Bad Request",
  "status": 400,
  "code": "GIUSTIFICATIVO_OBBLIGATORIO",
  "detail": "Il campo giustificativo è obbligatorio"
}
```

3.16 Download Attestato Partecipante

Consente di scaricare in formato PDF l'attestato di un partecipante iscritto all'edizione indicata. Il download via API è consentito **solo dopo la conferma conclusione edizione**.

3.16.1 Parametri

Metodo	ENDPOINT
GET	/corso/{corsoid}/edizione/{edizioneld}/partecipante/{codiceFiscale}/attestato

Nessun body. In caso di successo la risposta è il file binario PDF.

3.16.2 Esempi

Esempio di chiamata CURL:

```
curl -X GET "<Base_URL>/corso/12345/edizione/67890/partecipante/RSSMRA80A01H501U/attestato" \
-H "X-Gefos-Ente-Erogatore: 02438750586" \
-H "accept: application/pdf" \
-o "attestato_RSSMRA80A01H501U.pdf"
```

Esempio risposta in caso di successo (200 OK): file PDF (corpo della risposta).

Esempio risposta errore: Corso non trovato

```
{
  "title": "Not Found",
  "status": 404,
  "code": "NOT_FOUND",
}
```

```
"detail": "Risorsa non trovata"
}
```

Esempio risposta errore: Partecipante non presente

```
{
  "title": "Not Found",
  "status": 404,
  "code": "PARTECIPANTE_NON_PRESENTE",
  "detail": "Il partecipante non è presente nell'edizione"
}
```

Esempio risposta errore: Attestato non disponibile

```
{
  "title": "Unprocessable Entity",
  "status": 422,
  "code": "ATTESTATO_NON_DISPONIBILE",
  "detail": "L'attestato non è ancora disponibile: è necessaria la conferma conclusione edizione"
}
```

3.17 Download Attestati Tutti i Partecipanti

Consente di scaricare un archivio ZIP contenente un file PDF per ogni partecipante dell'edizione per cui l'attestato è disponibile. I partecipanti senza attestato disponibile non compaiono nell'archivio. Il download via API è consentito **solo dopo la conferma conclusione edizione**.

3.17.1 Parametri

Metodo	ENDPOINT
GET	/corso/{corsoid}/edizione/{edizioneId}/attestati

Nessun body. In caso di successo la risposta è il file binario ZIP.

3.17.2 Esempi

Esempio di chiamata CURL:

```
curl -X GET "<Base_URL>/corso/12345/edizione/67890/attestati" \
-H "X-Gefos-Ente-Erogatore: 02438750586" \
-H "accept: application/zip" \
-o "attestati_edizione_67890.zip"
```

Esempio risposta in caso di successo (200 OK): file ZIP (corpo della risposta).

Esempio risposta errore: Corso non trovato

```
{
  "title": "Not Found",
  "status": 404,
  "code": "NOT_FOUND",
  "detail": "Risorsa non trovata"
}
```

Esempio risposta errore: Nessun attestato disponibile

```
{
  "title": "Unprocessable Entity",
  "status": 422,
  "code": "NESSUN_ATTESTATO_DISPONIBILE",
  "detail": "Non è disponibile alcun attestato per i partecipanti dell'edizione"
}
```

4 Oggetti

4.1 Corso

Oggetto restituito da GET /corso/{coursold}. In creazione (POST /corso) viene restituito solo l'`id` del corso. POST /corso/{coursold}/edizione restituisce l'`id` della nuova edizione creata.

Campo	Tipo dato	Obbligatorio	Note
id	int	si	Identificativo del corso.
tipologiaCoursold	int	si	Id della tipologia del corso (vedi Sezione 4.5).
edizioni	array	si	Elenco completo delle edizioni associate (oggetto Edizione con tutti i campi valorizzati, incluso <code>id</code>).

Tabella 12: Oggetto Corso

4.2 Corso leggero

Oggetto restituito da GET /corsi/tipologiaCoursold/{tipologiaCoursold}.

Campo	Tipo dato	Obbligatorio	Note
id	int	si	Identificativo del corso.
tipologiaCoursold	int	si	Id della tipologia del corso (vedi Sezione 4.5).

Tabella 13: Oggetto Corso leggero

4.3 Edizione richiesta

Oggetto utilizzato nelle richieste POST e PUT per la gestione di un'edizione. In **creazione** (POST) include obbligatoriamente l'array `calendari`. In **modifica** (PUT .../edizione/{edizioneId}) sono ammessi solo `metodoSvolgimentoId`, `dataOraInizio` e `dataOraFine`; i calendari si aggiornano con PUT .../calendari. I partecipanti si gestiscono tramite le API dedicate.

Campo	Tipo dato	Obbligatorio	Note
metodoSvolgimentold	int	si	Id del metodo di svolgimento. Deve essere un id valido (i possibili valori sono specificati nella Sezione 6.1).
dataOrainizio	str	si	Data e ora di inizio dell'edizione. Formato YYYY-MM-DDTHH:mm (es. 2025-06-01T09:00).
dataOraFine	str	si	Data e ora di fine dell'edizione. Formato YYYY-MM-DDTHH:mm (es. 2025-06-01T13:00). Non può essere precedente a <code>dataOraInizio</code> .
calendari	array	cond.	Elenco completo degli slot (vedi Sezione 4.7). Obbligatorio in POST (almeno un elemento); non ammesso in PUT edizione.

Tabella 14: Oggetto Edizione richiesta

4.4 Edizione

Oggetto restituito da GET `/corso/{corsold}` e dalle risposte che includono il dettaglio completo dell'edizione. Ogni edizione include il proprio `id`. Il campo `partecipanti` non è accettato nelle richieste POST/PUT edizione. Il campo `calendari` è obbligatorio in creazione edizione; gli aggiornamenti successivi avvengono solo tramite PUT `.../calendari`.

Campo	Tipo dato	Obbligatorio	Note
id	int	si	Identificativo dell'edizione.
metodoSvolgimentold	int	si	Id del metodo di svolgimento. Deve essere un id valido (i possibili valori sono specificati nella Sezione 6.1).
dataOraInizio	str	si	Data e ora di inizio dell'edizione. Formato YYYY-MM-DDTHH:mm (es. 2025-06-01T09:00).
dataOraFine	str	si	Data e ora di fine dell'edizione. Formato YYYY-MM-DDTHH:mm (es. 2025-06-01T13:00). Non può essere precedente a <code>dataOraInizio</code> .
partecipanti	array	si	Elenco dei partecipanti iscritti (vedi oggetto Partecipante nella Sezione 4.6). Può essere vuoto. Può includere <code>esitoId</code> se valorizzato.
calendari	array	si	Elenco dei calendari (vedi oggetto Calendario nella Sezione 4.7). Sempre presente dopo la creazione dell'edizione.

Tabella 15: Oggetto Edizione

4.5 Tipologia Corso

Oggetto restituito da GET /tipologieCorso e riferito dal campo `tipologiaCorsoId` nelle API di creazione/modifica corso.

Campo	Tipo dato	Obbligatorio	Note
id	int	si	Identificativo della tipologia (valore da utilizzare come <code>tipologiaCorsoId</code>).
descrizione	str	si	Descrizione della tipologia corso.
idDenominazione	int	si	Identificativo della denominazione associata alla tipologia.
descrDenominazione	str	si	Descrizione testuale della denominazione.

Tabella 16: Oggetto Tipologia Corso

4.6 Dettaglio Partecipante

Campo	Tipo dato	Obbligatorio	Note
nome	str	si	Nome del partecipante.
cognome	str	si	Cognome del partecipante.
codiceFiscale	str	si	Formato standard 16 caratteri alfanumerici.
dataNascita	str	si	Formato YYYY-MM-DD.
luogoNascita	str	si	Comune o stato estero di nascita.
provNascita	str	si	Sigla della provincia di nascita (es. RM). Per nati all'estero, usare la codifica prevista (es. EE).
email	str	si	Indirizzo email valido.
esitoId	int	no	Id dell'esito del partecipante (vedi Sezione 6.3). Presente solo nelle risposte GET se valorizzato. Non inviare nelle richieste di inserimento partecipante.

Tabella 17: Oggetto Dettaglio Partecipante

4.7 Calendario

Oggetto utilizzato nella creazione edizione (POST .../edizione, array `calendari`), nell'aggiornamento massivo (PUT .../calendari) e nelle risposte che includono il dettaglio dei calendari (GET /corso).

Metodo effettivo: l'obbligatorietà dei campi condizionali dipende dal metodo di svolgimento **effettivo** della giornata:

- se l'edizione **non** è Mista (`metodoSvolgimentoId` $\neq$ 4) $\rightarrow$ il metodo effettivo è `metodoSvolgimentoId` dell'edizione;
- se l'edizione è Mista (`metodoSvolgimentoId` = 4) $\rightarrow$ il metodo effettivo è `metodoSvolgimentoGiornataId` della voce di calendario.

In base al metodo effettivo:

- **Presenza Fisica (1):** obbligatori i campi indirizzo;
- **E-Learning (2):** nessun campo aggiuntivo oltre a `dataOraDa` e `dataOraA`;
- **Video-Conferenza (3):** obbligatori `link` e `descrizione`.

Campo	Tipo dato	Obbligatorio	Note
id	int	no	Identificativo del calendario. Presente nelle risposte (GET /corso). In PUT .../calendari va reinviato per gli slot già esistenti (obbligatorio per gli slot con data precedente a oggi). Non inviare in creazione edizione.
dataOraDa	str	si	Data e ora di inizio. Formato YYYY-MM-DDTHH:mm (es. 2025-05-02T09:00). Deve essere compresa tra <code>dataOraInizio</code> e <code>dataOraFine</code> dell'edizione.
dataOraA	str	si	Data e ora di fine. Formato YYYY-MM-DDTHH:mm (es. 2025-05-02T13:00). Non può essere precedente a <code>dataOraDa</code> . Deve essere compresa tra <code>dataOraInizio</code> e <code>dataOraFine</code> dell'edizione.
metodoSvolgimento	ioinrtnatald	cond.	Id del metodo di svolgimento della giornata (vedi Sezione 6.2). Obbligatorio se l'edizione è Mista (id 4); non ammesso altrimenti.
link	str	cond.	URL della piattaforma di erogazione. Obbligatorio se il metodo effettivo è Video-Conferenza (id 3).
descrizione	str	cond.	Testo libero (es. istruzioni per l'account guest). Obbligatorio se il metodo effettivo è Video-Conferenza (id 3).
indirizzo	str	cond.	Indirizzo del luogo di svolgimento. Obbligatorio se il metodo effettivo è Presenza Fisica (id 1).
nCivico	str	cond.	Numero civico. Obbligatorio se il metodo effettivo è Presenza Fisica (id 1).
cap	str	cond.	Codice Avviamento Postale. Obbligatorio se il metodo effettivo è Presenza Fisica (id 1).
citta	str	cond.	Comune di svolgimento. Obbligatorio se il metodo effettivo è Presenza Fisica (id 1).
provincia	str	cond.	Sigla provincia (es. RM). Obbligatorio se il metodo effettivo è Presenza Fisica (id 1).

Tabella 18: Oggetto Calendario

5 Autenticazione e Accesso ad API Manager

5.1 Overview

Per poter invocare le API esposte sulla piattaforma API Manager, è necessario preventivamente autenticarsi presso API Manager per ottenere un token di accesso (Access Token) valido. Il meccanismo di autenticazione previsto è basato sul protocollo OAuth2 con flusso Client Credentials.

Il token ottenuto ha una validità temporale limitata (specificata nel campo `expires_in`) e deve essere utilizzato in tutte le chiamate successive verso gli endpoint applicativi.

5.2 Richiesta dell'Access Token

L'applicativo client deve effettuare una chiamata POST verso l'endpoint di autorizzazione fornendo le proprie credenziali tramite Basic Authentication.

Specifica	Valore
METODO	POST
ENDPOINT	<code>https://api.integrazione.lispa.it/oauth2/token</code>
HEADERS	Authorization: Basic <base64(client_id:client_secret)> Content-Type: application/x-www-form-urlencoded
BODY (Data)	<code>grant_type=client_credentials&scope=<codice_ente_formatore></code>

Tabella 19: Specifiche chiamata per ottenimento Token

5.2.1 Esempio Chiamata CURL

```
curl --location 'https://api.integrazione.lispa.it/oauth2/token' \  
--header 'Content-Type: application/x-www-form-urlencoded' \  
--header 'Authorization: Basic <auth_token>' \  
--data 'grant_type=client_credentials&scope=<codice_ente_formatore>'
```

Nota: Il valore `<auth_token>` nell'header Authorization è ottenuto codificando in Base64 la stringa composta da username e password del client (formato `username:password`). Il campo `<codice_ente_formatore>` contiene il codice dell'ente formatore di chi sta invocando la chiamata.

5.2.2 Esempio Risposta Successo

In caso di credenziali valide, il server restituisce un oggetto JSON contenente l'Access Token da utilizzare per le chiamate successive:

```
{
  "access_token": <bearer_token>,
  "scope": "",
  "token_type": "Bearer",
  "expires_in": 1800
}
```

5.3 Utilizzo del Token

Una volta ottenuto l'Access Token (valore del campo `access_token`), questo deve essere inserito nell'header di ogni chiamata verso le API di GEFOS come Bearer Token.

5.3.1 Esempio di chiamata autorizzata

Qualsiasi chiamata agli endpoint dell'applicativo dovrà includere l'header di autorizzazione come segue:

```
curl --location 'https://api.integrazione.lispa.it/app/...' \
--header 'Authorization: Bearer <bearer_token>'
```

Dove `<bearer_token>` è il valore dell'`access_token` ricevuto nella fase di autenticazione.

6 Dizionario dei valori codificati

6.1 Metodo Svolgimento

Valori ammessi per il campo `metodoSvolgimentoId` nell'oggetto Edizione.

Id	Descrizione
1	Presenza Fisica
2	E-Learning
3	Video-Conferenza
4	Mista

Tabella 20: Metodo Svolgimento

Se `metodoSvolgimentoId` è **Mista (4)**, ogni voce di calendario deve indicare un `metodoSvolgimentoGiornataId` (vedi Sezione 6.2 e Sezione 4.7).

6.2 Metodo Svolgimento Giornata

Valori ammessi per il campo `metodoSvolgimentoGiornataId` nell'oggetto Calendario.

Id	Descrizione
1	Presenza Fisica
2	E-Learning
3	Video-Conferenza

Tabella 21: Metodo Svolgimento Giornata

6.3 Esito

Id	Descrizione
1	Superato
2	Non Superato
3	Non Ammesso

Tabella 22: Possibili esiti del corso