Documentazione API Dettagliata e Professionale

Questa documentazione fornisce una guida completa all'API sviluppata utilizzando **FastAPI**, progettata per la gestione di contesti, file e configurazioni di catene (chains) in un sistema di elaborazione dati. L'API consente di creare, gestire ed eliminare contesti, caricare file associandoli a uno o più contesti, gestire i file caricati e configurare e caricare catene basate su specifici contesti e modelli di linguaggio (LLM).

Indice

Introduzione

L'API descritta in questa documentazione è progettata per interagire con un sistema backend esistente, facilitando la gestione dinamica di dati e modelli in applicazioni che utilizzano modelli di linguaggio

avanzati. Offre endpoint per operazioni di creazione, lettura, aggiornamento ed eliminazione (CRUD) su contesti e file, oltre a funzionalità avanzate per la configurazione e il caricamento di catene di elaborazione basate su modelli di linguaggio e archivi vettoriali (vector stores).

Panoramica Generale

L'API fornisce le seguenti funzionalità principali:

- **Gestione dei Contesti**: Creare, elencare ed eliminare contesti (directory) nel sistema, permettendo di organizzare i dati e le risorse in modo strutturato.
- **Gestione dei File**: Caricare file associandoli a uno o più contesti, elencare i file presenti in specifici contesti ed eliminare file sia per identificativo unico (UUID) che per percorso.
- **Configurazione e Caricamento delle Catene**: Configurare e caricare in memoria catene di elaborazione (chains) che utilizzano modelli di linguaggio (LLM) e archivi vettoriali, basati su contesti specifici.

L'API comunica con un backend esistente tramite richieste HTTP asincrone, utilizzando la libreria `httpx`.

Autenticazione e Sicurezza

Nota: Attualmente, l'API non implementa meccanismi di autenticazione o autorizzazione. Per ambienti di produzione, è fortemente consigliato implementare misure di sicurezza appropriate, come l'utilizzo di token di accesso, API key o protocolli di autenticazione come OAuth 2.0, per proteggere gli endpoint dall'accesso non autorizzato.

Gestione degli Errori

L'API gestisce gli errori restituendo risposte HTTP con codici di stato appropriati e messaggi di dettaglio nel formato JSON. In caso di errore, la risposta conterrà una struttura simile alla seguente:

```
{
  "detail": "Messaggio di errore dettagliato."
}
```

I codici di stato comuni includono:

- **200 OK**: Richiesta elaborata con successo.
- **400 Bad Request**: La richiesta contiene parametri non validi o mancanti.
- **404 Not Found**: La risorsa richiesta non è stata trovata.
- **500 Internal Server Error**: Errore interno del server durante l'elaborazione della richiesta.

Endpoint dell'API

1. Gestione dei Contesti

1.1 Creazione di un Nuovo Contesto

Endpoint

```
POST /contexts
```

Descrizione

Crea un nuovo contesto (directory) nel sistema backend.

Parametri

- **Form Data**:
  - `context_name` (string, obbligatorio): Nome del contesto da creare.
  - `description` (string, opzionale): Descrizione del contesto.

Esempio di Richiesta

```
curl -X POST "http://your-api-url/contexts" \
     -F "context_name=my_new_context" \
     -F "description=Contesto per progetti speciali"
```

Esempio di Risposta

```
{
  "path": "my_new_context",
  "custom_metadata": {
    "description": "Contesto per progetti speciali"
  }
}
```

1.2 Eliminazione di un Contesto

Endpoint

```
DELETE /contexts/{context_name}
```

Descrizione

Elimina un contesto esistente dal sistema backend, inclusi tutti i file e le risorse associate.

Parametri

- **Path Parameter**:
  - `context_name` (string, obbligatorio): Nome del contesto da eliminare.

Esempio di Richiesta

```
curl -X DELETE "http://your-api-url/contexts/my_context"
```

Esempio di Risposta

```
{
  "detail": "Contesto 'my_context' eliminato con successo."
}
```

1.3 Elenco dei Contesti Disponibili

Endpoint

```
GET /contexts
```

Descrizione

Recupera un elenco di tutti i contesti disponibili nel sistema backend.

Parametri

Nessuno.

Esempio di Richiesta

```
curl -X GET "http://your-api-url/contexts"
```

Esempio di Risposta

```
[
  {
    "path": "context1",
    "custom_metadata": {
      "description": "Descrizione del contesto 1"
    }
  },
  {
    "path": "context2",
    "custom_metadata": null
  }
]
```

---

2. Gestione dei File

2.1 Caricamento di un File su Contesti Multipli

Endpoint

```
POST /upload
```

Descrizione

Carica un file e lo associa a uno o più contesti specificati. Il file verrà elaborato e indicizzato per essere utilizzato nelle catene di elaborazione.

Parametri

- **Form Data**:
  - `file` (UploadFile, obbligatorio): Il file da caricare (ad esempio, PDF, DOCX).
  - `contexts` (string[], obbligatorio): Lista di contesti a cui associare il file. Può essere una stringa separata da virgole o un array.
  - `description` (string, opzionale): Descrizione del file.

Esempio di Richiesta

```
curl -X POST "http://your-api-url/upload" \
     -F "file=@/path/to/your/file.pdf" \
     -F "contexts=context1,context2" \
     -F "description=Documento importante"
```

Esempio di Risposta

```
{
  "file_id": "123e4567-e89b-12d3-a456-426614174000",
  "contexts": ["context1", "context2"]
}
```

2.2 Elenco dei File nei Contesti

Endpoint

```
GET /files
```

Descrizione

Recupera un elenco di file presenti in specifici contesti. Se nessun contesto è specificato, restituisce tutti i file disponibili nel sistema.

Parametri

- **Query Parameters**:
  - `contexts` (string[], opzionale): Lista di contesti per filtrare i file.

Esempio di Richiesta

```
curl -X GET "http://your-api-url/files?contexts=context1&contexts=context2"
```

Esempio di Risposta

```
[
  {
    "path": "context1/file1.pdf",
    "custom_metadata": {
      "file_uuid": "123e4567-e89b-12d3-a456-426614174000",
      "description": "Documento importante"
    }
  },
  {
    "path": "context2/file2.docx",
    "custom_metadata": {
      "file_uuid": "223e4567-e89b-12d3-a456-426614174001",
      "description": "Relazione annuale"
    }
  }
]
```

2.3 Eliminazione di un File

Endpoint

```
DELETE /files
```

Descrizione

Elimina un file dal sistema, identificandolo tramite il suo UUID (eliminazione da tutti i contesti) o tramite il suo percorso (eliminazione da un contesto specifico).

Parametri

- **Query Parameters** (almeno uno è obbligatorio):
  - `file_id` (string, opzionale): UUID del file da eliminare.
  - `file_path` (string, opzionale): Percorso del file da eliminare, ad esempio “context1/file1.pdf”.

Esempio di Richiesta (per UUID)

```
curl -X DELETE "http://your-api-url/files?file_id=123e4567-e89b-12d3-a456-426614174000"
```

Esempio di Richiesta (per percorso)

```
curl -X DELETE "http://your-api-url/files?file_path=context1/file1.pdf"
```

Esempio di Risposta

```
{
  "detail": "File con ID 123e4567-e89b-12d3-a456-426614174000 eliminato da tutti i contesti
}
```


---

3. Configurazione e Caricamento delle Catene

3.1 Configurazione e Caricamento di una Catena

Endpoint

```
POST /configure_and_load_chain/
```

Descrizione

Configura e carica in memoria una catena (chain) di elaborazione basata su un contesto specifico e un modello di linguaggio (LLM). Questa catena può essere utilizzata per operazioni di domanda e risposta (Q&A), ricerca semantica e altre applicazioni che coinvolgono modelli di linguaggio avanzati.

Parametri

- **Query Parameters**:
  - `context` (string, opzionale, default: “default”): Il contesto su cui basare la configurazione della catena.
  - `model_name` (string, opzionale, default: “gpt-4o-mini”): Nome del modello LLM da utilizzare. Esempi: “gpt-4o”, “gpt-3.5-turbo”.

Esempio di Richiesta

```
curl -X POST "http://your-api-url/configure_and_load_chain/?context=my_context&model_name=g
```

Esempio di Risposta


```
{
  "message": "Chain configurata e caricata con successo.",
  "llm_load_result": {
    "status": "success",
    "model": "chat-openai_gpt-4o-mini"
  },
  "config_result": {
    "chain_id": "my_context_qa_chain",
    "status": "configured"
  },
  "load_result": {
    "chain_id": "my_context_qa_chain",
    "status": "loaded"
  }
}
```

Modelli e Strutture Dati

ContextMetadata

Descrizione

Rappresenta i metadati associati a un contesto.

Attributi

- `path` (string): Il percorso o nome del contesto.
- `custom_metadata` (dict, opzionale): Metadati personalizzati associati al contesto.

Esempio

```
{
  "path": "my_context",
  "custom_metadata": {
    "description": "Contesto per progetti speciali"
  }
}
```

FileUploadResponse

Descrizione

Rappresenta la risposta dopo il caricamento di un file.

Attributi

- `file_id` (string): UUID univoco generato per il file.
- `contexts` (list of string): Lista dei contesti ai quali il file è stato associato.

Esempio

```
{
  "file_id": "123e4567-e89b-12d3-a456-426614174000",
  "contexts": ["context1", "context2"]
}
```

Esempi di Utilizzo

Caricamento di un File e Configurazione di una Catena

Passo 1: Caricare un File

Carichiamo un file PDF associandolo al contesto “project_alpha”.

```
curl -X POST "http://your-api-url/upload" \
     -F "file=@/path/to/your/document.pdf" \
     -F "contexts=project_alpha" \
     -F "description=Documentazione del Progetto Alpha"
```

Risposta

```
{
  "file_id": "a1b2c3d4-e5f6-7890-abcd-1234567890ab",
  "contexts": ["project_alpha"]
}
```

Passo 2: Configurare e Caricare la Catena

Configuriamo e carichiamo una catena utilizzando il contesto “project_alpha” e il modello LLM “gpt-4o-mini”.

```
curl -X POST "http://your-api-url/configure_and_load_chain/?context=project_alpha&model_nam
```

Risposta

```
{
  "message": "Chain configurata e caricata con successo.",
  "llm_load_result": {
    "status": "success",
    "model": "chat-openai_gpt-4o-mini"
  },
  "config_result": {
    "chain_id": "project_alpha_qa_chain",
    "status": "configured"
  },
  "load_result": {
    "chain_id": "project_alpha_qa_chain",
    "status": "loaded"
  }
}
```

Appendice

Note Tecniche

- **Comunicazione Asincrona**: L’API utilizza `httpx.AsyncClient` per effettuare richieste asincrone al backend, migliorando l’efficienza e riducendo i tempi di attesa.
- **Middleware CORS**: È stato configurato il middleware `CORSMiddleware` per permettere richieste cross-origin. In ambienti di produzione, si consiglia di restringere le origini consentite per motivi di sicurezza.

- **Gestione delle Eccezioni**: L'API gestisce le eccezioni utilizzando `HTTPException` di FastAPI, restituendo messaggi di errore chiari e codici di stato appropriati.
- **Timeout Personalizzati**: Alcune operazioni utilizzano impostazioni di timeout estese (ad esempio, `httpx.Timeout(600.0)`) per gestire richieste che possono richiedere più tempo, come il caricamento di modelli di grandi dimensioni.