Fireflies home pagedark logo

Search or ask...
⌘K
Support
Dashboard

Getting Started
Introduction
Quickstart
Chat with Fireflies AI Assistant
Fundamentals
General concepts
Limits
Errors
Authorization
Introspection
Examples
Overview
Basic
Advanced
Graphql API
Query
User
Users
Transcript
Transcripts
Bite
Bites
Mutation
Set User Role
Delete Transcript
Upload Audio
Create bite
Add to Live
Webhooks
Schema
AIFilter
AudioUploadStatus
Bite
MeetingAttendee
MeetingInfo
Sentence
Transcript
Summary
Speaker
User
Input

Getting Started

Introduction

Welcome to Fireflies public API documentation.

We are actively working to expose more functionality via our API. If you have specific requests, please reach out to us.

Introduction

Overview of the API

The Fireflies API is built on top of GraphQL, a powerful interface designed to provide you with efficient and flexible access to your data. This API allows you to retrieve exactly the data you need in a structured format. Whether you are building a web application, a mobile app, or a complex software system, our API caters to a wide range of data requirements.

Our API covers various functionalities, including queries for fetching data as well as uploading meeting audio. It is designed to be intuitive and easy to use, ensuring that you can start fetching and manipulating data with minimal setup.

Advantages of Using GraphQL

1. Precise Data Fetching: One of the key strengths of GraphQL is its ability to return exactly what you request and nothing more. This precision eliminates the over-fetching of data, common in traditional REST APIs, leading to more efficient network utilization and faster response times.

2. Single Endpoint: Unlike REST APIs, which often require multiple endpoints for different data needs, GraphQL operates through a single endpoint. This simplification streamlines interactions with the API and makes maintaining and managing the API more straightforward.

3. Flexibility and Scalability: GraphQL APIs are incredibly flexible, allowing for queries that can evolve with your needs. This flexibility, combined with efficient data retrieval, makes GraphQL an ideal choice for both small projects and large-scale applications.

Fireflies home pagedark logo

Search or ask...
⌘K
Support
Dashboard

Input
AudioUploadInput
Attendee
Role
Misclellaneous
Language codes
Error codes
Additional Info
Deprecated
Changelog
Getting Started

Quickstart

Make your first request in under 5 minutes.

This guide provides step-by-step instructions to make your first query with our GraphQL API and demonstrates basic functionality.

Step 1: Setting Up

Before diving into querying the API, it’s essential to set up your environment correctly. This includes obtaining authentication credentials and configuring your development environment.

Obtaining Authentication Credentials

To access our API, you will need an API key. Follow these steps to obtain your key:

Log in to your account at app.fireflies.ai
Navigate to the Integrations section
Click on Fireflies API
Copy and store your API key securely

It’s crucial to handle your API key with the utmost care to ensure the security of your data. For more information on Authorization and best practices, visit Authorization

Step 2: Making Your First Request

Execute a simple query to retrieve a list of users.

Replace your_api_key with your API key in the following requests

curl

python

javascript

java

```
curl \
   -X POST \
   -H "Content-Type: application/json" \
   -H "Authorization: Bearer your_api_key" \
   --data '{ "query": "{ users { name user_id } }" }' \
   https://api.fireflies.ai/graphql
```

When building GraphQL queries for this API, focus on precision and efficiency. Start with simple queries and gradually increase complexity. Ensure you only request the data you need to avoid over-fetching.

Review the Schema Documentation: For guidance, refer to the Schema section and use tools like GraphQL Playground for testing. Efficient queries lead to better performance and a smoother API experience. More details on building your GraphQL query are available here

Step 3: Analyzing the Response

You will receive a JSON response with the requested data. Example response:

curl

```
{
        "data":
        {
                "users": [
                        {
                                "name":"Justin Fly",
                                "user_id":"example-id"
                        }
                ]
        }
}
```

Continue to the next sections for more detailed examples and advanced usage instructions.

Fireflies home pagedark logo

Search or ask...
⌘K
Support
Dashboard

Changelog
Getting Started
Chat with Fireflies AI Assistant
Navigate Fireflies.ai Documentation using the AI Assistant

Overview

Welcome to the Fireflies.ai Documentation, your essential guide to unlocking the full potential of our API. We understand that diving into a new API can be daunting, and finding the specific information you need shouldn’t add to that challenge. That’s why we’ve designed our documentation to be both comprehensive and user-friendly, ensuring you get the answers you need without the hassle.

Explore with Fireflies AI Assistant

With our AI assistant, navigating the documentation becomes as intuitive as having a conversation. Whether you’re looking for detailed API endpoints, integration guides, or troubleshooting tips, our integrated AI assistant is here to assist you every step of the way.

Fireflies home pagedark logo

Search or ask...
⌘K
Support
Dashboard

Getting Started
Introduction
Quickstart
Chat with Fireflies AI Assistant
Fundamentals
General concepts
Limits
Errors
Authorization
Introspection
Examples
Overview
Basic
Advanced
Graphql API
Query
User
Users
Transcript
Transcripts

Fundamentals

General concepts

Foundational guide to the core aspects of the Fireflies API

Overview

This section serves as a primer on the essential features of the Fireflies API. It covers the requirements to make authenticated requests to the Fireflies API, key GraphQL concepts and a breakdown of core components like queries, mutations, schema and data types.

Host

The Fireflies API is hosted at https://api.fireflies.ai

Authorization

The Fireflies API requires a valid API key for all requests. Please see Authorization for more information

GraphQL API Concepts

Welcome to our GraphQL API! GraphQL is a powerful query language designed for APIs, offering clients the ability to request exactly what they need and nothing more.

Queries

Queries are used to fetch data in GraphQL. They are akin to the GET request in REST. A basic query might look like this:

Queries are read-only and won’t modify your data.

graphql

```
query {
  user(id: "1") {
    name
    email
  }
}
```

Mutations

Mutations allow you to modify data – create, update, or delete.

Mutations should be used with caution as they change server-side data.

graphql

```
mutation {
  setUserRole(user_id: "1", role: "user") {
    id
    name
  }
}
```

Schema and Types

Schema Definition

The schema defines the API’s capabilities, including types, queries, mutations, and more.

Understanding GraphQL Schema

The schema is a contract between client and server, defining how clients can access data.

Data Types

Standard Types: Int, Float, String, Boolean, ID.

Custom Types: Defining complex data structures. Custom types are user-defined and can include combinations of standard types.

Best Practices

Optimize query performance by requesting for only necessary data.
Use variables in queries to enhance readability and flexibility

Additional Resources

For more in-depth information, visit GraphQL Documentation.

Quickstart
Foundational guide to the core aspects of the Fireflies API

Limits
File size and API rate limits for the Fireflies API

Fireflies home pagedark logo

Search or ask...
⌘K
Support
Dashboard

Getting Started
Introduction
Quickstart
Chat with Fireflies AI Assistant
Fundamentals
General concepts
Limits
Errors
Authorization
Introspection
Examples
Overview
Basic
Advanced
Graphql API
Query
User
Users
Transcript
Transcripts
Bite
Bites
Mutation
Set User Role
Delete Transcript

Upload Audio
Create bite
Add to Live
Webhooks
Schema
AIFilter
AudioUploadStatus
Bite
MeetingAttendee
MeetingInfo
Sentence
Transcript
Summary
Speaker
User
Input
AudioUploadInput
Attendee
Role
Misclellaneous
Language codes
Error codes
Additional Info
Deprecated
Changelog
Fundamentals

Limits

File size and API rate limits for the Fireflies API

Overview

Understanding the limitations of our API is crucial for efficient and uninterrupted usage. Below you’ll find detailed information about the upload limits for different file types and the rate limits applicable to various subscription plans.

Upload Limits

The Fireflies API accommodates a range of file sizes for different user types. Here’s a breakdown of the maximum file sizes for audio and video uploads:

Upload Type	Free User Limit	Pro / Business / Enterprise Limit
Audio Files	Up to 200MB	Up to 200MB
Video Files	Up to 100MB	Up to 1.5GB

Understanding Upload Limits

Audio Files: All users can upload an audio file no greater than 200MB, ensuring ample capacity for high-quality audio content.

Video Files for Free Users: Free users have a maximum upload limit of 100MB for video files, suitable for short clips and previews.
Video Files for Pro/Business/Enterprise Users: Higher-tier users can upload larger video files up to 1.5GB, accommodating longer and higher resolution content.

API Rate Limits

To maintain the quality of service and availability for all users, our API enforces rate limits based on the type of subscription plan.

Plan	API Rate Limit
Free / Pro	50 requests per day
Business / Enterprise	60 requests per min

Navigating API Rate Limits

Free and Pro Plans: These plans are ideal for light to moderate usage, with a cap of 50 API requests per day. This rate is suitable for testing and small-scale applications.
Business and Enterprise Plans: Designed for more demanding use cases, these plans allow up to 60 requests per minute, providing ample capacity for larger applications and higher volume demands.

Additional Resources

General Concepts
Foundational guide to the core aspects of the Fireflies API

Fireflies home pagedark logo

Search or ask...
⌘K
Support
Dashboard

Getting Started
Introduction
Quickstart
Chat with Fireflies AI Assistant
Fundamentals
General concepts
Limits
Errors
Authorization
Introspection
Examples
Overview
Basic
Advanced
Graphql API
Query

Errors

Error standards for the Fireflies API

Overview

Understanding how errors are structured and returned by the Fireflies API is key to effectively handling and troubleshooting issues. This page outlines the common error format and details specific error types.

Error schema

Our GraphQL API follows a standard format for returning errors. Errors are encapsulated within an errors array in the response body.

Please visit Error codes to view explanations for error code types

message
String
required
Description of the error

code
String
required
Error code.

friendly
Boolean
required
friendly === true are safe to show to the frontend client. Unfriendly errors may have technical details that may not be useful to the UI layer.

extensions
Object
Contains useful metadata related to the error

Example

```
{
  "data": {},
  "errors": [
    {
      "message": "Error description",
          "friendly": true,
      "code": "error_code",
      "extensions": {
        "code": "error_code",
        "status": http_status_code,
                ... otherFields
      }
    }
  ]
}
```

Additional Resources
Limits
File size and API rate limits for the Fireflies API

Authorization
Authenticating your requests with the Fireflies API

Fireflies home pagedark logo

Search or ask...
⌘K
Support
Dashboard

Getting Started
Introduction
Quickstart
Chat with Fireflies AI Assistant
Fundamentals
General concepts
Limits
Errors
Authorization
Introspection
Examples
Overview
Basic
Advanced
Graphql API
Query
User
Users
Transcript
Transcripts
Bite
Bites
Mutation
Set User Role
Delete Transcript
Upload Audio
Create bite
Add to Live
Webhooks
Schema
AIFilter
AudioUploadStatus
Bite
MeetingAttendee
MeetingInfo
Sentence
Transcript
Summary
Speaker
User
Input

AudioUploadInput
Attendee
Role
Misclellaneous
Language codes
Error codes
Additional Info
Deprecated
Changelog
Fundamentals

Authorization

Authenticating your requests with the Fireflies API

Overview

The Fireflies API implements token-based authentication, which ensures that only authorized users can access certain data and functionalities.

Token-Based Authentication

We use a standard bearer token authentication mechanism. This means that to make authorized requests to the API, you must include an Authorization header with a valid token.

Acquiring a Token

Follow these steps to obtain your API key for the Fireflies API:

Log in to your account at fireflies.ai
Navigate to the Integrations section
Click on Fireflies API
Copy and store your API key securely

Making an Authenticated Request

To make an authenticated request, add the Authorization header followed by the word Bearer and your API key.

Example of an Authenticated Request Header

```
Authorization: Bearer your_api_key
```

Example request with Authorization header

curl

javascript

python

java

```
 curl \
   -X POST \
   -H "Content-Type: application/json" \
   -H "Authorization: Bearer your_api_key" \
   --data '{ "query": "{ user { name integrations } }" }' \
   https://api.fireflies.ai/graphql
```

Ensure to replace your_api_key with your actual API key.

Best Practices for Token Security

Keep it Secret: Treat your API key like a password. Never expose it in client-side code or share it publicly.

Store Securely: Store the API key securely in your application, ideally in environment variables or secure storage solutions.

Improper handling of API keys can lead to security vulnerabilities. Always ensure API keys are used and stored securely.

Troubleshooting

Invalid key: If you receive an error regarding an invalid API key, verify that the API key hasn’t expired and that it’s correctly included in the request header.

Missing key: Ensure that the Authorization header is present in your requests requiring authentication.

If you encounter issues with authentication or have questions about API key management, please contact our support team.

Fireflies home pagedark logo

Search or ask...
⌘K
Support
Dashboard

Getting Started
Introduction
Quickstart
Chat with Fireflies AI Assistant
Fundamentals
General concepts
Limits
Errors
Authorization
Introspection

Query generation and API exploration.

Overview

Introspection is a feature that allows querying a GraphQL server to discover its schema. This capability is crucial for developers to understand available queries, mutations, and the structure of the data they can work with, facilitating seamless API interaction and exploration.

Requirements

You will need a Fireflies.ai API key to use introspection. For more details, please visit Authorization

Introspection using Apollo Sandbox

For introspection using our builtin Apollo Sandbox, visit api.fireflies.ai/graphql and enter your api_key in the Headers section as a Bearer token

Introspection using Postman

Create a new Graphql Request in Postman with the url api.fireflies.ai/graphql and enter your api_key in the Headers section as a Bearer token

Fireflies home pagedark logo

Search or ask...
⌘K
Support
Dashboard

Getting Started
Introduction
Quickstart
Chat with Fireflies AI Assistant
Fundamentals
General concepts
Limits
Errors
Authorization
Introspection
Examples
Overview
Basic
Advanced
Graphql API
Query
User
Users
Transcript
Transcripts
Bite

Overview

This part of the documentation is designed to provide developers with practical insights into integrating and maximizing the Fireflies.ai API in various applications.

By visiting our Replit page at Fireflies.ai on Replit, you can directly access and interact with these examples, making your learning process interactive and engaging.

Fireflies.ai Replit

Explore a wide array of examples where we have created code snippets that allows you to interact with the Fireflies API.

To view, please visit replit.com/@firefliesai

You can also view the list of examples at the links Basic examples and Advanced examples

Additional Resources
Introspection
Query generation and API exploration.

Basic examples
Basic usage examples

Fireflies home pagedark logo

Search or ask...
⌘K
Support
Dashboard

Getting Started
Introduction
Quickstart
Chat with Fireflies AI Assistant
Fundamentals
General concepts
Limits
Errors
Authorization
Introspection
Examples
Overview
Basic
Advanced
Graphql API
Query
User
Users
Transcript
Transcripts
Bite
Bites
Mutation
Set User Role
Delete Transcript
Upload Audio
Create bite
Add to Live
Webhooks

Schema

AIFilter

AudioUploadStatus

Bite

MeetingAttendee

MeetingInfo

Sentence

Transcript

Summary

Speaker

User

Input

AudioUploadInput

Attendee

Role

Misclellaneous

Language codes

Error codes

Additional Info

Deprecated

Changelog

Examples

Basic

Library of basic usage examples

Overview

Explore basic usage examples for the Fireflies.ai API to get you started quickly. These basic usage examples simplify interact with the API through Javascript code.

Basic examples

Basic examples that allow you to perform basic operations with the Fireflies.ai API, like fetching transcripts or setting user role.

Create soundbite from meeting

Set user role

Upload audio

Download meeting video

Get audio/video url

Get transcript summary

Fireflies.ai Replit

Explore the complete list of usage examples at replit.com/@firefliesai

Additional Resources

Overview

Overview of Fireflies.ai API Examples

Advanced examples
Advanced usage examples

Fireflies home pagedark logo

Search or ask...
⌘K
Support
Dashboard

Getting Started
Introduction
Quickstart
Chat with Fireflies AI Assistant
Fundamentals
General concepts
Limits
Errors
Authorization
Introspection
Examples
Overview
Basic
Advanced
Graphql API
Query
User
Users
Transcript
Transcripts
Bite
Bites
Mutation
Set User Role
Delete Transcript
Upload Audio
Create bite
Add to Live
Webhooks
Schema
AIFilter
AudioUploadStatus
Bite
MeetingAttendee
MeetingInfo
Sentence
Transcript

Summary
Speaker
User
Input
AudioUploadInput
Attendee
Role
Misclellaneous
Language codes
Error codes
Additional Info
Deprecated
Changelog
Examples
Advanced
Library of advanced usage examples

Overview

Explore advanced usage examples that build on top of the Fireflies.ai API. This page is regularly updated

Find questions from external participants

This replit allows you to query questions asked by participants who do not belong to your organization. It can be useful in deriving insights and analysis for client calls, interviews, etc.

Fetch meetings from multiple users

This replit allows you to fetch meetings from multiple users by providing it a list of API keys. Please read our privacy policy and terms of usage for more details

Fireflies.ai Replit

Explore the complete list of usage examples at replit.com/@firefliesai

Additional Resources

Basic examples
Basic usage examples

User
Querying user details

Was this page helpful?

Yes

Fireflies home pagedark logo

Search or ask...
⌘K
Support
Dashboard

Getting Started
Introduction
Quickstart
Chat with Fireflies AI Assistant
Fundamentals
General concepts
Limits
Errors
Authorization
Introspection
Examples
Overview
Basic
Advanced
Graphql API
Query
User
Users
Transcript
Transcripts
Bite
Bites
Mutation
Set User Role
Delete Transcript
Upload Audio
Create bite
Add to Live
Webhooks
Schema
AIFilter
AudioUploadStatus
Bite
MeetingAttendee
MeetingInfo
Sentence
Transcript
Summary
Speaker
User

Input
AudioUploadInput
Attendee
Role
Misclellaneous
Language codes
Error codes
Additional Info
Deprecated
Changelog
Query
User
Querying user details

curl

javascript

python

java

```
curl -X POST \
        -H "Content-Type: application/json" \
        -H "Authorization: Bearer your_api_key" \
        --data '{ "query": "query User($userId: String!) { user(id: 
$userId) { name integrations } }", "variables": { "userId": 
"your_user_id" } }' \
        https://api.fireflies.ai/graphql
```

Response

```
{
  "data": {
    "user": {
      "name": "Justin Fly",
      "integrations": ["string"],
    }
  }
}
```

Overview
The user query is designed to fetch details associated with a specific user id.

Arguments
id

String
id is an optional argument. Not passing an ID to this query will return user details for the owner of the API key

Schema
Fields available to the User query

Usage Example

```
query User($userId: String!) {
  user(id: $userId) {
    user_id
    recent_transcript
    recent_meeting
    num_transcripts
    name
    minutes_consumed
    is_admin
    integrations
    email
  }
}
```

Additional Resources
Advanced examples
Advanced usage examples

Users
Querying list of users

Was this page helpful?

Fireflies home pagedark logo

Search or ask...
⌘K
Support
Dashboard

Getting Started
Introduction
Quickstart
Chat with Fireflies AI Assistant
Fundamentals
General concepts
Limits

Query

Users

Querying list of users

curl

javascript

python

java

```
curl -X POST \
        -H "Content-Type: application/json" \
        -H "Authorization: Bearer your_api_key" \
        --data '{ "query": "{ users { name integrations } }" }' \
        https://api.fireflies.ai/graphql
```

Response

```
{
  "data": {
    "users": [
      {
        "name": "Justin Fly",
        "integrations": []
      },
      {
        "name": "Peter Fire",
        "integrations": []
      }
    ]
  }
}
```

Overview

The users query is designed to fetch a list of all users within the team. You can also view this list on your dashboard at app.fireflies.ai/team

Schema

Fields available to the User query

Usage Example

```
query Users {
  users {
    user_id
    email
    name
    num_transcripts
    recent_meeting
    minutes_consumed
    is_admin
    integrations
  }
}
```

Additional Resources
User
Querying user details

Transcript
Querying transcript details

Was this page helpful?

Fireflies home pagedark logo

Search or ask...
⌘K
Support
Dashboard

Getting Started
Introduction
Quickstart
Chat with Fireflies AI Assistant
Fundamentals
General concepts
Limits
Errors
Authorization
Introspection
Examples
Overview
Basic
Advanced
Graphql API
Query
User
Users
Transcript
Transcripts
Bite
Bites
Mutation
Set User Role
Delete Transcript
Upload Audio

Create bite
Add to Live
Webhooks
Schema
AIFilter
AudioUploadStatus
Bite
MeetingAttendee
MeetingInfo
Sentence
Transcript
Summary
Speaker
User
Input
AudioUploadInput
Attendee
Role
Misclellaneous
Language codes
Error codes
Additional Info
Deprecated
Changelog

Query

Transcript

Querying transcript details

curl

javascript

python

java

```
curl -X POST \
        -H "Content-Type: application/json" \
        -H "Authorization: Bearer your_api_key" \
        --data '{ "query": "query Transcript($transcriptId: String!)
{ transcript(id: $transcriptId) { title id } }", "variables":
{ "transcriptId": "your_transcript_id" } }' \
        https://api.fireflies.ai/graphql
```

Response

```
{
  "data": {
    "transcript": {
      "title": "Weekly sync",
      "id": "transcript-id",
    }
  }
}
```

Overview

The transcript query is designed to fetch details associated with a specific transcript ID.

Arguments

id
String
required

Schema

Fields available to the Transcript query

Usage Example

```
query Transcript($transcriptId: String!) {
  transcript(id: $transcriptId) {
    id
    dateString
    privacy
        speakers {
                id
                name
        }
    sentences {
      index
      speaker_name
      speaker_id
      meeting_info {
        fred_joined
        silent_meeting
        summary_status
      }
      text
      raw_text
      start_time
      end_time
      ai_filters {
        task
        pricing
        metric
        question
        date_and_time
        text_cleanup
        sentiment
      }
    }
    title
    host_email
    organizer_email
    calendar_id
    user {
      user_id
      email
      name
      num_transcripts
      recent_meeting
      minutes_consumed
      is_admin
      integrations
    }
    fireflies_users
    participants
    date
    transcript_url
    audio_url
    video_url
    duration
    meeting_attendees {
      displayName
      email
      phoneNumber
      name
      location
    }
    summary {
      gist
      bullet_gist
      short_overview
      action_items
      keywords
      outline
      overview
      shorthand_bullet
    }
  }
}
```

Additional Resources

Users

Querying list of users

Transcripts
Querying list of transcripts

Was this page helpful?

Fireflies home pagedark logo

Search or ask...
⌘K
Support
Dashboard

Getting Started
Introduction
Quickstart
Chat with Fireflies AI Assistant
Fundamentals
General concepts
Limits
Errors
Authorization
Introspection
Examples
Overview
Basic
Advanced
Graphql API
Query
User
Users
Transcript
Transcripts
Bite
Bites
Mutation
Set User Role
Delete Transcript
Upload Audio
Create bite
Add to Live
Webhooks
Schema
AIFilter
AudioUploadStatus
Bite
MeetingAttendee

MeetingInfo
Sentence
Transcript
Summary
Speaker
User
Input
AudioUploadInput
Attendee
Role
Misclellaneous
Language codes
Error codes
Additional Info
Deprecated
Changelog
Query
Transcripts
Querying list of transcripts

curl

javascript

python

java

```
curl -X POST \
    -H "Content-Type: application/json" \
    -H "Authorization: Bearer your_api_key" \
    --data '{ "query": "query Transcripts($userId: String)
{ transcripts(user_id: $userId) { title id } }" }' \
    https://api.fireflies.ai/graphql
```

Response

```
{
  "data": {
    "transcripts": [
                {
                        "title": "Weekly sync",
                        "id": "transcript-id",
                },
                {
                        "title": "ClientMeeting.mp3",
                        "id": "transcript-id-2",
                }
```

```
        ]
  }
}
```

Overview

The transcripts query is designed to fetch a list of transcripts against input arguments.

Arguments

title
String
Title of the transcript

fromDate
DateTime
Return all transcripts created after fromDate. The fromDate parameter accepts a date-time string in the ISO 8601 format, specifically in the form YYYY-MM-DDTHH:mm.sssZ. For example, a valid timestamp would be 2024-07-08T22:13:46.660Z.

toDate
DateTime
Return all transcripts created before toDate. The toDate parameter accepts a date-time string in the ISO 8601 format, specifically in the form YYYY-MM-DDTHH:mm.sssZ. For example, a valid timestamp would be 2024-07-08T22:13:46.660Z.

date
Float
This field is deprecated. Please use fromDate and toDate instead. Return all transcripts created within the date specified. Query input value must be in milliseconds. For example, you can use the JavaScript new Date().getTime() to get the datetime in milliseconds which should look like this 1621292557453. The timezone for this field is UTC +00:00

For more details regarding time since EPOCH

limit
Int
Number of transcripts to return. Maxiumum 50 in one query

skip
Int
Number of transcripts to skip.

host_email
String

Filter all meetings accordingly to meetings that have this email as the hosting.

organizer_email
String
Filter all meetings accordingly to meetings that have this email as the organizer.

participant_email
String
Filter all meetings accordingly to all meetings that contains this email as an attendee.

user_id
String
User id. Filter all meetings accordingly to meetings that have this user ID as the hosting or participant.

Schema

Fields available to the Transcript query

Usage Example

```
query Transcripts(
  $title: String
  $date: DateTime
  $limit: Int
  $skip: Int
  $hostEmail: String
  $participantEmail: String
  $userId: String
) {
  transcripts(
    title: $title
    date: $date
    limit: $limit
    skip: $skip
    host_email: $hostEmail
    participant_email: $participantEmail
    user_id: $userId
  ) {
    id
    sentences {
      index
      speaker_name
      speaker_id
      text
      raw_text
      start_time
      end_time
      ai_filters {
        task
        pricing
        metric
        question
        date_and_time
        text_cleanup
        sentiment
      }
    }
    title
        speakers {
                id
                name
        }
    host_email
    organizer_email
    meeting_info {
      fred_joined
      silent_meeting
      summary_status
    }
    calendar_id
    user {
      user_id
      email
      name
      num_transcripts
      recent_meeting
      minutes_consumed
      is_admin
      integrations
    }
    fireflies_users
    participants
    date
    transcript_url
    audio_url
    video_url
    duration
    meeting_attendees {
      displayName
      email
      phoneNumber
      name
      location
```

```
    }
    summary {
      gist
      bullet_gist
      short_overview
      action_items
      keywords
      outline
      overview
      shorthand_bullet
    }
  }
}
```

Additional Resources
Transcript
Querying transcript details

Bite
Querying bite details

Was this page helpful?

Fireflies home pagedark logo

Search or ask...
⌘K
Support
Dashboard

Getting Started
Introduction
Quickstart
Chat with Fireflies AI Assistant
Fundamentals
General concepts
Limits
Errors
Authorization
Introspection
Examples
Overview
Basic
Advanced
Graphql API
Query
User
Users
Transcript

Query
Bite

Querying bite details

curl

javascript

python

java

```
curl -X POST \
        -H "Content-Type: application/json" \
        -H "Authorization: Bearer your_api_key" \
        --data '{ "query": "query Bite($biteId: ID!) { bite(id:
$biteId) { user_id name status summary } }", "variables": { "biteId":
```

```
"your_bite_id" } }' \
        https://api.fireflies.ai/graphql
```

Response

```
{
  "data": {
    "bite": {
      "user_id": "user-id",
      "id": "bite-id",
    }
  }
}
```

Overview

The bite query is designed to fetch details associated with a specific bite ID.

Arguments

id

ID

required

Unique identifier of the bite

Schema

Fields available to the Bite query

Usage Example

```
query Bite($biteId: ID!) {
  bite(id: $biteId) {
    transcript_id
    name
    id
    thumbnail
    preview
    status
    summary
    user_id
    start_time
    end_time
    summary_status
    media_type
    created_at
    created_from {
      description
      duration
      id
      name
      type
    }
    captions {
      end_time
      index
      speaker_id
      speaker_name
      start_time
      text
    }
    sources {
      src
      type
    }
    privacies
    user {
      first_name
      last_name
      picture
      name
      id
    }
  }
}
```

Additional Resources
Transcripts
Querying list of transcripts

Bites
Querying list of bites

Was this page helpful?

Fireflies home pagedark logo

Search or ask...
⌘K
Support
Dashboard

Getting Started
Introduction
Quickstart
Chat with Fireflies AI Assistant
Fundamentals

curl

javascript

python

java

```
curl -X POST \
        -H "Content-Type: application/json" \
        -H "Authorization: Bearer your_api_key" \
        --data '{ "query": "query Bites($mine: Boolean) { bites(mine:
$mine) { user_id name end_time } }", "variables": { "mine": true } }'
\
        https://api.fireflies.ai/graphql
```

Response

```
{
  "data": {
    "bites": [
                {
                        "user_id": "user-id",
                        "id": "bite-id",
        },
                {
                        "user_id": "user-id",
                        "id": "bite-id-2",
        }
        ]
  }
}
```

Overview

The bites query is designed to fetch a list of bites against input arguments.

Arguments

mine

Boolean

required

The mine parameter, when set to true, fetches results specific to the owner of the API key

transcript_id

ID

You can use transcript_id to query all bites against a specific transcript.

my_team
Boolean
The my_team parameter, when set to true, fetches results for the owner of the API key

limit
Int
Maximum number of bites to fetch in a single query. Maximum of 50

skip
Int
Number of records to skip over. Helps paginate results when used in combination with the limit param.

Schema
Fields available to the Bites query

Usage Example

```
query Bites($mine: Boolean) {
  bites(mine: $mine) {
    transcript_id
    name
    id
    thumbnail
    preview
    status
    summary
    user_id
    start_time
    end_time
    summary_status
    media_type
    created_at
    created_from {
      description
      duration
      id
      name
      type
    }
    captions {
      end_time
      index
      speaker_id
      speaker_name
      start_time
      text
    }
    sources {
      src
      type
    }
    privacies
    user {
      first_name
      last_name
      picture
      name
      id
    }
  }
}
```

Additional Resources
Bite
Querying bite details

Set User Role
Use the API to set user roles

Fireflies home pagedark logo

Search or ask...
⌘K
Support
Dashboard

Getting Started
Introduction
Quickstart
Chat with Fireflies AI Assistant
Fundamentals
General concepts
Limits
Errors
Authorization
Introspection
Examples
Overview
Basic
Advanced
Graphql API
Query

User
Users
Transcript
Transcripts
Bite
Bites
Mutation
Set User Role
Delete Transcript
Upload Audio
Create bite
Add to Live
Webhooks
Schema
AIFilter
AudioUploadStatus
Bite
MeetingAttendee
MeetingInfo
Sentence
Transcript
Summary
Speaker
User
Input
AudioUploadInput
Attendee
Role
Misclellaneous
Language codes
Error codes
Additional Info
Deprecated
Changelog

Mutation

Set User Role

Use the API to set user roles

curl

javascript

python

java

```
curl -X POST https://api.fireflies.ai/graphql \
        -H "Content-Type: application/json" \
        -H "Authorization: Bearer your_api_key" \
        -d '{
                "query": "mutation($user_id: String!, $role: Role!)
{ setUserRole(user_id: $user_id, role:$role) { name is_admin } }",
                "variables": {
                        "user_id": "your_user_id",
                        "role": "admin"
                }
        }'
```

Response

```
{
  "data": {
    "setUserRole": {
      "name": "Justin Fly",
      "is_admin": "true",
    }
  }
}
```

Overview

The setUserRole mutation allows for the updating of a user’s role within a team.

Arguments

user_id
String
required
The unique identifier of the user.

role
Role
required
The Role to be assigned to the user. Valid types for user are admin and user

Usage Example

To set a user’s role, provide the user’s ID and the desired role as arguments to the mutation. Here’s an example of how this mutation could be used:

```
mutation setUserRole($userId: String!, $role: Role!) {
  setUserRole(user_id: $userId, role: $role) {
    id
    name
    email
    role
  }
}
```

Additional Resources

Bites
Querying list of bites

Delete Transcript
Use the API to manage transcript deletion

Was this page helpful?

Fireflies home pagedark logo

Search or ask...
⌘K
Support
Dashboard

Getting Started
Introduction
Quickstart
Chat with Fireflies AI Assistant
Fundamentals
General concepts
Limits
Errors
Authorization
Introspection
Examples
Overview
Basic
Advanced
Graphql API
Query
User
Users
Transcript
Transcripts
Bite
Bites
Mutation
Set User Role
Delete Transcript
Upload Audio
Create bite

Add to Live
Webhooks
Schema
AIFilter
AudioUploadStatus
Bite
MeetingAttendee
MeetingInfo
Sentence
Transcript
Summary
Speaker
User
Input
AudioUploadInput
Attendee
Role
Misclellaneous
Language codes
Error codes
Additional Info
Deprecated
Changelog
Mutation
Delete Transcript
Use the API to manage transcript deletion

curl

javascript

python

java

```
curl -X POST \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer your_api_key" \
  -d '{"query": "mutation($transcriptId: String!)
{ deleteTranscript(id: $transcriptId) { title date duration
organizer_email } }", "variables": {"transcriptId":
"your_transcript_id"}}' \
  https://api.fireflies.ai/graphql
```

Response

```
{
  "data": {
    "deleteTranscript": {
      "duration": "1",
      "date": 1699570138000,
      "organizer_email": "justin@fly.ai",
      "title": "Video title"
    }
  }
}
```

Overview

The deleteTranscript mutation is designed to delete a specific transcript by its ID.

Arguments

id
String
required
Transcript ID

Usage Example

To delete a transcript, provide the unique id of the transcript as an argument to the mutation. The returned subfields will be from the deleted transcript. Here’s an example of how this mutation could be used:

```
mutation deleteTranscript($id: String!) {
  deleteTranscript(id: $id) {
    id
    title
    host_email
    organizer_email
    fireflies_users
    participants
    date
    transcript_url
    audio_url
    duration
  }
}
```

Additional Resources

Set User Role
Use the API to set user roles

Upload Audio
Use the API to upload audio to Fireflies.ai

Was this page helpful?

Fireflies home pagedark logo

Search or ask...
⌘K
Support
Dashboard

User
Input
AudioUploadInput
Attendee
Role
Misclellaneous
Language codes
Error codes
Additional Info
Deprecated
Changelog
Mutation
Upload Audio
Use the API to upload audio to Fireflies.ai

curl

javascript

python

java

```
curl -X POST \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer your_api_key" \
  -d '{
    "query": "mutation($input: AudioUploadInput) { uploadAudio(input: 
$input) { success title message } }",
    "variables": {
      "input": {
        "url": "https://url-to-the-audio-file",
        "title": "title of the file",
        "attendees": [
          {
            "displayName": "Fireflies Notetaker",
            "email": "notetaker@fireflies.ai",
            "phoneNumber": "xxxxxxxxxxxxxxxx"
          },
          {
            "displayName": "Fireflies Notetaker 2",
            "email": "notetaker2@fireflies.ai",
            "phoneNumber": "xxxxxxxxxxxxxxxx"
          }
        ]
      }
    }
  }' \
  https://api.fireflies.ai/graphql
```

Response

```
{
  "data": {
    "uploadAudio": {
      "success": true,
      "title": "title of the file",
      "message": "Uploaded audio has been queued for processing."
    }
  }
}
```

Overview
The uploadAudio mutation allows you to upload audio files to Fireflies.ai for transcription.

Arguments
input
AudioUploadInput

Show child attributes

Usage Example
To upload a file, provide the necessary input parameters to the mutation. Here’s an example of how this mutation could be used:

```
mutation uploadAudio($input: AudioUploadInput) {
  uploadAudio(input: $input) {
    success
    title
    message
  }
}
```

Additional Resources
Delete Transcript
Use the API to manage transcript deletion

Create Bite
Use the API to create a bite

Was this page helpful?

Yes

Input
AudioUploadInput
Attendee
Role
Misclellaneous
Language codes
Error codes
Additional Info
Deprecated
Changelog
Mutation

Create bite

Use the API to create a bite

curl

javascript

python

java

```
curl -X POST https://api.fireflies.ai/graphql \
        -H "Content-Type: application/json" \
        -H "Authorization: Bearer your_api_key" \
        -d '{
                "query": "mutation CreateBite($transcriptId: ID!,
$startTime: Float!, $endTime: Float!) { createBite(transcript_id:
$transcriptId, start_time: $startTime, end_time: $endTime) { summary
status id } }",
                "variables": {
                        "transcriptId": "your_transcript_id",
                        "startTime": 0,
                        "endTime": 5
                }
        }'
```

Response

```
{
  "data": {
    "createBite": {
      "name": "bite-name",
      "status": "pending",
    }
  }
}
```

Overview

The createBite mutation allows you to create a bite through the API.

Arguments

transcript_id
ID
required
ID of the transcript

name
String
Name of the bite

start_time
Float
required
Start time of the bite in seconds

end_time
Float
required
End time of the bite in seconds

media_type
String
Type of the bite, either ‘video’ or ‘audio’

privacies
[String]
Array specifying the visibility of the Soundbite. Possible values are ‘public’, ‘team’, and ‘participants’. For example, [“team”, “participants”] indicates visibility to both team members and participants, while [“public”] denotes full public access.

summary
String
Summary for the bite

Usage Example

To create a bite, provide the necessary input parameters to the mutation. Here’s an example of how this mutation could be used:

```
mutation Mutation($transcriptId: ID!, $startTime: Float!, $endTime:
Float!) {
  createBite(transcript_id: $transcriptId, start_time: $startTime,
end_time: $endTime) {
    status
    name
    id
  }
}
```

Additional Resources
Upload Audio
Use the API to upload audio to Fireflies.ai

Add to Live
Use the API to add the Fireflies.ai bot to an ongoing meeting

Fireflies home pagedark logo

Search or ask...
⌘K
Support
Dashboard

Getting Started
Introduction
Quickstart
Chat with Fireflies AI Assistant
Fundamentals
General concepts
Limits
Errors
Authorization
Introspection
Examples
Overview
Basic
Advanced
Graphql API
Query
User
Users
Transcript
Transcripts
Bite
Bites
Mutation
Set User Role
Delete Transcript
Upload Audio
Create bite
Add to Live
Webhooks

Schema
AIFilter
AudioUploadStatus
Bite
MeetingAttendee
MeetingInfo
Sentence
Transcript
Summary
Speaker
User
Input
AudioUploadInput
Attendee
Role
Misclellaneous
Language codes
Error codes
Additional Info
Deprecated
Changelog
Mutation

Add to Live

Use the API to add the Fireflies.ai bot to an ongoing meeting

curl

javascript

python

java

```
curl -X POST \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer your_api_key" \
  -d '{
    "query": "mutation AddToLiveMeeting($meetingLink: String!)
{ addToLiveMeeting(meeting_link: $meetingLink) { success } }",
    "variables": {
      "meetingLink": "https://meet.google.com/code-here"
    }
  }' \
  https://api.fireflies.ai/graphql
```

Response

```
{
  "data": {
    "addToLiveMeeting": {
      "success": true,
    }
  }
}
```

Overview

The addToLiveMeeting mutation allows you to add the Fireflies.ai bot to an ongoing meeting.

Arguments

meeting_link
String!
required
A valid http URL for the meeting link, i.e. gooogle meet, zoom, etc

title
String
Title or name of the meeting, this will be used to identify the transcribed file. If title is not provided, a default title will be set automatically

meeting_password
String
Password for the meeting, if applicable.

duration
Int
Meeting duration in minutes. Minimum of 15 and maximum of 120 minutes. Defaults to 60 minutes if param is not provided

language
String
Language of the meeting. Defaults to English if not provided. For a complete list of language codes, please view Language Codes

attendees
[Attendee]
Array of Attendees for expected meeting participants.

Usage Example

To upload an audio file, provide the necessary input parameters to the mutation. Here’s an example of how this mutation could be used:

```
mutation AddToLiveMeeting($meetingLink: String!) {
  addToLiveMeeting(meeting_link: $meetingLink) {
    success
  }
}
```

Additional Resources
Create Bite
Use the API to create a bite

Webhooks
Create notifications using webhooks

Was this page helpful?

Fireflies home pagedark logo

Search or ask...
⌘K
Support
Dashboard

Getting Started
Introduction
Quickstart
Chat with Fireflies AI Assistant
Fundamentals
General concepts
Limits
Errors
Authorization
Introspection
Examples
Overview
Basic
Advanced
Graphql API
Query
User
Users
Transcript
Transcripts
Bite
Bites
Mutation
Set User Role
Delete Transcript
Upload Audio
Create bite
Add to Live

Graphql API

Webhooks

Webhook events for the Fireflies.ai API

Overview

Webhooks enable your application to set up event based notifications. In this section, you’ll learn how to configure webhooks to receive updates from Fireflies.

Events supported

The webhooks support the following events:

Transcription complete: Triggers when a meeting has been processed and the transcript is ready for viewing

Saving a webhook

Follow the instructions below to save a webhook URL that sends notifications for all subscribed events. This webhook will only be fired for meetings that you own.

1
Visit the Fireflies.ai dashboard settings
2
Navigate to the Developer settings tab
3

Enter a valid https URL in the webhooks field and save
You may test your webhook using the upload audio API or by uploading through the dashboard at app.fireflies.ai/upload

Upload audio webhook
You can also include a webhook URL as part of an upload audio request. This is different from the saved webhook as it will only send notifications for that singular audio upload request.

curl

javascript

python

java

```
curl -X POST \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer your_api_key" \
  -d '{
    "query": "mutation($input: AudioUploadInput) { uploadAudio(input:
$input) { success title message } }",
    "variables": {
      "input": {
        "url": "https://url_to_the_audio_file",
        "title": "title of the file",
        "webhook": "https://url_for_the_webhook"
      }
    }
  }' \
  https://api.fireflies.ai/graphql
```

Webhook Schema
meetingId
String
required
Identifier for the meeting / transcript that the webhook has triggered for. MeetingId and TranscriptId are used interchangeably for the Fireflies.ai Platform.

eventType
String
Name of the event type that has been fired against the webhook

clientReferenceId
ID

Custom identifier set by the user during upload. You may use this to identify your uploads in your events.

Example Payload

```
{
    "meetingId": "ASxwZxCstx",
    "eventType": "Transcription completed",
    "clientReferenceId": "be582c46-4ac9-4565-9ba6-6ab4264496a8"
}
```

Additional Resources
Add to Live
Use the API to add the Fireflies.ai bot to an ongoing meeting

AIFilter
Schema for AIFilter

Fireflies home pagedark logo

Search or ask...
⌘K
Support
Dashboard

Getting Started
Introduction
Quickstart
Chat with Fireflies AI Assistant
Fundamentals
General concepts
Limits
Errors
Authorization
Introspection
Examples
Overview
Basic
Advanced
Graphql API
Query
User
Users
Transcript
Transcripts
Bite
Bites
Mutation

Set User Role
Delete Transcript
Upload Audio
Create bite
Add to Live
Webhooks
Schema
AIFilter
AudioUploadStatus
Bite
MeetingAttendee
MeetingInfo
Sentence
Transcript
Summary
Speaker
User
Input
AudioUploadInput
Attendee
Role
Misclellaneous
Language codes
Error codes
Additional Info
Deprecated
Changelog
Schema
AIFilter
Schema for AIFilter

task
String
Description

pricing
String
Description

metric
String
Description

question
String
Description

date_and_time
String

Description

text_cleanup
String
Description

sentiment
String
Description

Additional Resources
Webhooks
Create notifications using webhooks

AudioUploadStatus
Schema for AudioUploadStatus

Fireflies home pagedark logo

Search or ask...
⌘K
Support
Dashboard

Getting Started
Introduction
Quickstart
Chat with Fireflies AI Assistant
Fundamentals
General concepts
Limits
Errors
Authorization
Introspection
Examples
Overview
Basic
Advanced
Graphql API
Query
User
Users
Transcript
Transcripts
Bite
Bites
Mutation
Set User Role

Delete Transcript
Upload Audio
Create bite
Add to Live
Webhooks
Schema
AIFilter
AudioUploadStatus
Bite
MeetingAttendee
MeetingInfo
Sentence
Transcript
Summary
Speaker
User
Input
AudioUploadInput
Attendee
Role
Misclellaneous
Language codes
Error codes
Additional Info
Deprecated
Changelog

Schema

AudioUploadStatus

Schema for AudioUploadStatus

success
Boolean
Indicates whether the AudioUpload request was a success or not.

title
String
Title of the uploaded file.

message
String
Message from AudioUpload request.

Additional Resources

AIFilter
Schema for AIFilter

Bite
Schema for Bite

Was this page helpful?

Fireflies home pagedark logo

Search or ask...
⌘K
Support
Dashboard

Getting Started
Introduction
Quickstart
Chat with Fireflies AI Assistant
Fundamentals
General concepts
Limits
Errors
Authorization
Introspection
Examples
Overview
Basic
Advanced
Graphql API
Query
User
Users
Transcript
Transcripts
Bite
Bites
Mutation
Set User Role
Delete Transcript
Upload Audio
Create bite
Add to Live
Webhooks
Schema
AIFilter
AudioUploadStatus
Bite
MeetingAttendee
MeetingInfo
Sentence
Transcript
Summary

Speaker
User
Input
AudioUploadInput
Attendee
Role
Misclellaneous
Language codes
Error codes
Additional Info
Deprecated
Changelog
Schema
Bite
Schema for Bite

id
String
A unique identifier for the Bite

transcript_id
String
A unique identifier for the transcript the Bite is associated to

start_time
String
Start time for the Bite

end_time
String
End time for the Bite

name
String
A string representing the title of the Bite

thumbnail
String
URL of the Bite’s thumbnail image

preview
String
URL to a short preview video of the Bite

status
String
Current processing status of the Bite. Acceptable values include ‘pending’, ‘processing’, ‘ready’, and ‘error’

summary
String
An AI-generated summary describing the content of the Bite

userId
String
Identifier of the user who created the Bite

summary_status
String
Status of the AI summary generation process

media_type
String
Type of the Bite, either ‘video’ or ‘audio’

privacies
[BitePrivacy]
Array specifying the visibility of the Bite. Possible values are public, team, and participants. For example, ["team", "participants"] indicates visibility to both team members and participants, while ["public"] allows anyone to access the bite through its link

created_at
String
The date when this Bite was created

user
BiteUser
Object representing the user who created the Bite, including relevant user details

Show properties

sources
[MediaSource]
Array of MediaSource objects for the Bite

Show properties

captions
[BiteCaption]
Array of Object describing text captions associated with the Bite

Show properties

created_from
BiteOrigin
Object describing the origin of the Bite with the following properties

Show properties

Additional Resources
AudioUploadStatus
Schema for AudioUploadStatus

MeetingAttendee
Schema for MeetingAttendee

Was this page helpful?

Fireflies home pagedark logo

Search or ask...
⌘K
Support
Dashboard

Getting Started
Introduction
Quickstart
Chat with Fireflies AI Assistant
Fundamentals
General concepts
Limits
Errors
Authorization
Introspection
Examples
Overview
Basic
Advanced
Graphql API
Query
User
Users
Transcript
Transcripts
Bite
Bites
Mutation
Set User Role
Delete Transcript

Schema

MeetingAttendee

Schema for MeetingAttendee

displayName
String
Display name of the meeting attendee.

email
String
Email address of the meeting attendee.

phoneNumber
String
Phone number of the meeting attendee.

name
String
Full name of the meeting attendee.

location
String
Deprecated field

Additional Resources
Upload Audio
Use the API to upload audio to Fireflies.ai

MeetingInfo
Schema for MeetingInfo

Was this page helpful?

Fireflies home pagedark logo

Search or ask...
⌘K
Support
Dashboard

Getting Started
Introduction
Quickstart
Chat with Fireflies AI Assistant
Fundamentals
General concepts
Limits
Errors
Authorization
Introspection
Examples
Overview
Basic
Advanced
Graphql API
Query
User
Users
Transcript
Transcripts
Bite
Bites
Mutation
Set User Role
Delete Transcript
Upload Audio
Create bite
Add to Live
Webhooks
Schema
AIFilter
AudioUploadStatus

Bite
MeetingAttendee
MeetingInfo
Sentence
Transcript
Summary
Speaker
User
Input
AudioUploadInput
Attendee
Role
Misclellaneous
Language codes
Error codes
Additional Info
Deprecated
Changelog
Schema

MeetingInfo

Schema for MeetingInfo

fred_joined
Boolean
Boolean value that returns true if the bot joined the call, false otherwise.

silent_meeting
Boolean
Boolean value that returns true if the meeting does not contain any spoken words. Otherwise false.

summary_status
SummaryStatus
String value representing the summary status. Possible values are processing, processed, failed, skipped.

Additional Resources
MeetingAttendee
Schema for MeetingAttendee

Sentence
Schema for Sentence

Was this page helpful?

Fireflies home pagedark logo

Search or ask...
⌘K
Support
Dashboard

Getting Started
Introduction
Quickstart
Chat with Fireflies AI Assistant
Fundamentals
General concepts
Limits
Errors
Authorization
Introspection
Examples
Overview
Basic
Advanced
Graphql API
Query
User
Users
Transcript
Transcripts
Bite
Bites
Mutation
Set User Role
Delete Transcript
Upload Audio
Create bite
Add to Live
Webhooks
Schema
AIFilter
AudioUploadStatus
Bite
MeetingAttendee
MeetingInfo
Sentence
Transcript
Summary
Speaker
User
Input
AudioUploadInput
Attendee

Role
Misclellaneous
Language codes
Error codes
Additional Info
Deprecated
Changelog
Schema
Sentence
Schema for Sentence

index
Int
Index

text
String
Default transcription sentence or user edited transcription sentence.

raw_text
String
Transcribed sentence from meeting audio

start_time
String
Start time of Sentence

end_time
String
End time of Sentence

speaker_id
ID
Unique identifier for Speaker

speaker_name
String
Name of the speaker.

ai_filters
AIFilter
Sentiment analysis from meeting audio. Type of AIFilter

Fireflies home pagedark logo

Search or ask...
⌘K
Support

Additional Info
Deprecated
Changelog
Schema
Transcript
Schema for Transcript

id
ID
Unique identifier of the Transcript.

title
String
Title of the Transcript.

host_email
String
DEPRECATED
Email address of the meeting host.

organizer_email
String
Email address of the meeting organizer.

user
User
The User who Fred recorded the meeting on behalf of

speakers
[Speaker]
The speakers array contains the id and name of the speaker as it appears within the transcript

transcript_url
String
The url to view the transcript in the dashboard

participants
[String]
An array of email addresses of meeting participants guests, including participants that do not have Fireflies account.

meeting_attendees
[MeetingAttendee]
List of MeetingAttendee

fireflies_users
[String]
An array of email addresses of only Fireflies users participants that

have fireflies account that participated in the meeting

duration
Number
Duration of the audio

dateString
DateTime
String representation of DateTime. Example: 2024-04-22T20:14:04.454Z

date
Float
Date the transcript was created represented in milliseconds from EPOCH.

The timezone for this field is UTC +00:00

audio_url
String
Secure, newly generated hashed url that allows you download meeting audio. This url expires after every 24 hours. You'd have to make another request to generate a new audio_url.

You need to be subscribed to subscribed to a pro or higher plan to query audio_url. View plans here

video_url
String
Secure, newly generated hashed url that allows you download meeting video. This url expires after every 24 hours. You'd have to make another request to generate a new video_url. You will need to enable RECORD MEETING VIDEO setting on your Fireflies dashboard for this to work.

You need to be subscribed to a business or higher plan to query video_url. View plans here

sentence
[Sentence]
An array of Sentence(s), containing transcript details like raw_text, speaker_name, etc.

calendar_id
String
Calendar provider event ID. This field represents calId for google calendar and iCalUID for outlook calendar.

summary
Summary

AI generated Summary of the meeting.

meeting_info
MeetingInfo
MeetingInfo metadata fields.

Fireflies home pagedark logo

Search or ask...
⌘K
Support
Dashboard

Getting Started
Introduction
Quickstart
Chat with Fireflies AI Assistant
Fundamentals
General concepts
Limits
Errors
Authorization
Introspection
Examples
Overview
Basic
Advanced
Graphql API
Query
User
Users
Transcript
Transcripts
Bite
Bites
Mutation
Set User Role
Delete Transcript
Upload Audio
Create bite
Add to Live
Webhooks
Schema
AIFilter
AudioUploadStatus
Bite
MeetingAttendee
MeetingInfo
Sentence

Transcript
Summary
Speaker
User
Input
AudioUploadInput
Attendee
Role
Misclellaneous
Language codes
Error codes
Additional Info
Deprecated
Changelog
Schema

Summary

AI generated summary of the meeting.

Meeting AI summary prompts are defined by AI apps and can be customized to your needs.

action_items
String
A list of action items generated by the AI based on the meeting transcript.

keywords
String
A list of keywords generated by the AI based on the meeting transcript.

outline
String
An outline of the meeting with timestamps generated by the AI based on the meeting transcript.

overview
String
A summary of the meeting generated by the AI.

shorthand_bullet
String
A list of shorthand bullets generated by the AI.

gist
String
A summary of the meeting in 1 sentence

bullet_gist

String
Summary of the meeting in a few bullet points with descriptive emojis

short_summary
String
Summary of the meeting in a single paragraph

Fireflies home pagedark logo

Search or ask...
⌘K
Support
Dashboard

Getting Started
Introduction
Quickstart
Chat with Fireflies AI Assistant
Fundamentals
General concepts
Limits
Errors
Authorization
Introspection
Examples
Overview
Basic
Advanced
Graphql API
Query
User
Users
Transcript
Transcripts
Bite
Bites
Mutation
Set User Role
Delete Transcript
Upload Audio
Create bite
Add to Live
Webhooks
Schema
AIFilter
AudioUploadStatus
Bite
MeetingAttendee

Schema

User

Schema for User

user_id
String
Unique identifier for the User.

email
String
Email address for the User.

name
String
Full name of the User.

num_transcripts
Int
Total number of transcripts associated with the user.

recent_transcript
String
Most recent transcript generated by the user.

recent_meeting
String
Details about the user’s most recent meeting.

minutes_consumed
Int
Total number of minutes consumed by the user in meetings.

is_admin

Boolean
Indicates whether the user has administrative privileges. True if the user is an admin, false otherwise.

integrations
[String]
A list of integrations enabled for the user’s account.

Was this page helpful?

Fireflies home pagedark logo

Search or ask...
⌘K
Support
Dashboard

Getting Started

Fundamentals

Examples

Graphql API
Query

Mutation

Schema

Input

AudioUploadInput

Schema for AudioUploadInput

url
String
required
URL from which the audio file will be fetched. This should be a direct link to the audio resource.

title
String
Title assigned to the uploaded file. If not provided, the file’s original name will be used as its title.

attendees
[Attendee]
Array of Attendee objects, as defined in the Attendee schema. Each element in this array represents an attendee.

custom_language
String
Custom language code for the meeting

client_reference_id
String
Custom identifier set by the user during upload. You may use this to identify your uploads in your events.

save_video

Boolean
Boolean value that specifies whether the content video needs to be saved.

Was this page helpful?

Fireflies home pagedark logo

Search or ask...
⌘K
Support
Dashboard

Getting Started
Introduction
Quickstart
Chat with Fireflies AI Assistant
Fundamentals
General concepts
Limits
Errors
Authorization
Introspection
Examples
Overview
Basic
Advanced
Graphql API
Query
User
Users
Transcript
Transcripts
Bite
Bites
Mutation
Set User Role
Delete Transcript
Upload Audio
Create bite
Add to Live
Webhooks
Schema
AIFilter
AudioUploadStatus
Bite
MeetingAttendee
MeetingInfo

Sentence
Transcript
Summary
Speaker
User
Input
AudioUploadInput
Attendee
Role
Misclellaneous
Language codes
Error codes
Additional Info
Deprecated
Changelog

Input

Attendee

Schema for Attendee

displayName
String
Display name of the attendee as it appears in meeting platforms like Google Chat, Zoom, etc.

email
String
Email address of the attendee.

phoneNumber
String
Phone number of the attendee.

Was this page helpful?

Fireflies home pagedark logo

Search or ask...
⌘K
Support
Dashboard

Getting Started
Introduction
Quickstart
Chat with Fireflies AI Assistant
Fundamentals
General concepts
Limits

Errors
Authorization
Introspection
Examples
Overview
Basic
Advanced
Graphql API
Query
User
Users
Transcript
Transcripts
Bite
Bites
Mutation
Set User Role
Delete Transcript
Upload Audio
Create bite
Add to Live
Webhooks
Schema
AIFilter
AudioUploadStatus
Bite
MeetingAttendee
MeetingInfo
Sentence
Transcript
Summary
Speaker
User
Input
AudioUploadInput
Attendee
Role
Misclellaneous
Language codes
Error codes
Additional Info
Deprecated
Changelog

Input
Role
Schema for Role

role
enum

Valid types for role are admin and user

Was this page helpful?

Fireflies home pagedark logo

Search or ask...
⌘K
Support
Dashboard

Getting Started
Introduction
Quickstart
Chat with Fireflies AI Assistant
Fundamentals
General concepts
Limits
Errors
Authorization
Introspection
Examples
Overview
Basic
Advanced
Graphql API
Query
User
Users
Transcript
Transcripts
Bite
Bites
Mutation
Set User Role
Delete Transcript
Upload Audio
Create bite
Add to Live
Webhooks
Schema
AIFilter
AudioUploadStatus
Bite
MeetingAttendee
MeetingInfo
Sentence
Transcript

Summary
Speaker
User
Input
AudioUploadInput
Attendee
Role
Misclellaneous
Language codes
Error codes
Additional Info
Deprecated
Changelog
Misclellaneous

Language codes

Language code abbreviations for the Fireflies.ai API

Overview

This page lists the language codes supported by the Fireflies.ai API. You may use these codes with the uploadAudio or addToLive functionality to specify custom languages for your meetings. Each language entry includes the language name and its corresponding code, providing a quick and easy reference.

References

Add to Live
Upload Audio

Codes

```
[
  {
    "languageName": "Arabic",
    "languageCode": "ar"
  },
  {
    "languageName": "Bulgarian",
    "languageCode": "bg"
  },
  {
    "languageName": "Chinese",
    "languageCode": "zh"
  },
  {
    "languageName": "Croatian",
    "languageCode": "hr"
  },
  {
    "languageName": "Czech",
    "languageCode": "cs"
  },
  {
    "languageName": "Danish",
    "languageCode": "da"
  },
  {
    "languageName": "Dutch",
    "languageCode": "nl"
  },
  {
    "languageName": "English",
    "languageCode": "en"
  },
  {
    "languageName": "US English",
    "languageCode": "en-US"
  },
  {
    "languageName": "Australia English",
    "languageCode": "en-AU"
  },
  {
    "languageName": "UK English",
    "languageCode": "en-GB"
  },
  {
    "languageName": "Finnish",
    "languageCode": "fi"
  },
  {
    "languageName": "French",
    "languageCode": "fr"
  },
  {
    "languageName": "German",
    "languageCode": "de"
  },
  {
    "languageName": "Hebrew",
    "languageCode": "he"
  },
  {
    "languageName": "Hindi",
    "languageCode": "hi"
  },
  {
    "languageName": "Hungarian",
    "languageCode": "hu"
  },
  {
    "languageName": "Indonesian",
    "languageCode": "id"
  },
  {
    "languageName": "Italian",
    "languageCode": "it"
  },
  {
    "languageName": "Japanese",
    "languageCode": "ja"
  },
  {
    "languageName": "Korean",
    "languageCode": "ko"
  },
  {
    "languageName": "Malay",
    "languageCode": "ms"
  },
  {
    "languageName": "Norwegian",
    "languageCode": "no"
  },
  {
    "languageName": "Polish",
    "languageCode": "pl"
  },
  {
    "languageName": "Portuguese",
    "languageCode": "pt"
  },
  {
    "languageName": "Romanian",
    "languageCode": "ro"
  },
  {
    "languageName": "Russian",
    "languageCode": "ru"
  },
  {
    "languageName": "Slovak",
    "languageCode": "sk"
  },
  {
    "languageName": "Spanish",
    "languageCode": "es"
  },
  {
    "languageName": "Latin American Spanish",
    "languageCode": "es-419"
  },
  {
    "languageName": "Swedish",
    "languageCode": "sv"
  },
  {
    "languageName": "Tamil",
    "languageCode": "ta"
  },
  {
    "languageName": "Thai",
    "languageCode": "th"
  },
  {
    "languageName": "Filipino",
    "languageCode": "tl"
  },
  {
    "languageName": "Turkish",
    "languageCode": "tr"
  },
  {
    "languageName": "Ukrainian",
    "languageCode": "uk"
  },
  {
    "languageName": "Vietnamese",
    "languageCode": "vi"
  }
]
```

Was this page helpful?

Fireflies home pagedark logo

Search or ask...
⌘K
Support
Dashboard

Getting Started
Introduction
Quickstart
Chat with Fireflies AI Assistant

Overview

This page lists the error codes and their corresponding reasons for the Fireflies.ai API. You can refer to this page to understand the meaning and possible causes of different error codes that you may encounter while using the API. It provides a comprehensive reference for troubleshooting and resolving issues. Please visit Errors page for more details

API Errors

invalid_arguments

Returned when invalid arguments are passed to a query or mutation

```
{
  "errors": [
    {
      ... other fields for error
      "message": "Invalid argument(s) were provided",
      "code": "invalid_arguments",
      "extensions": {
        "code": "invalid_arguments",
        "status": 400,
        "metadata": {
          "fields": [
            {
              "name": "fromDate",
              "message": "fromDate must be a Date instance",
              "constraints": [
                {
                  "type": "isDate",
                  "message": "fromDate must be a Date instance"
                }
              ]
            }
          ]
        }
      }
    }
  ]
}
```

object_not_found

Returned when the subject of your query or mutation is not found. For example, querying a non-existent userId would throw an object_not_found error of the type User

```
{
  "errors": [
    {
          ... other fields for error
      "message": "User not found",
      "code": "object_not_found",
      "extensions": {
        "code": "object_not_found",
        "status": 404,
        "metadata": {
          "objectType": "User"
        }
      }
    }
  ],
}
```

forbidden
Returned when you are not allowed to perform an action

```
{
  "errors": [
    {
          ... other fields for error
      "message": "You are not authorized to perform this action",
      "code": "forbidden",
      "extensions": {
        "code": "forbidden",
        "status": 403,
      }
    }
  ]
}
```

paid_required
Returned when you are required to be subscribed to a paid plan for the Fireflies.ai platform. The error will also mentioned the required tier for such actions. For example, making a request to uploadAudio as a free user will throw a paid_required error with tier pro_or_higher, which means that you need to be subscribed to a Pro or Higher plan to perform this action

```
{
  "errors": [
    {
          ... other fields for error
      "message": "You need to be subscribed to a paid plan to perform
this action",
      "code": "paid_required",
      "extensions": {
        "code": "paid_required",
        "status": 403,
        "metadata": {
          "tier": "pro_or_higher"
        },
      }
    }
  ]
}
```

not_in_team
Returned when you are attempting to query against a userId that is not a part of your team

```
{
  "errors": [
    {
          ... other fields for error
      "message": "You do not have permissions for this team",
      "code": "not_in_team",
      "extensions": {
        "code": "not_in_team",
        "status": 403,
      }
    }
  ]
}
```

require_elevated_privilege
Returned when you are attempting to perform admin actions as a non-admin user

```
{
  "errors": [
    {
          .. other fields for error
      "message": "You do not have permission to perform this action",
      "code": "require_elevated_privilege",
      "extensions": {
        "code": "require_elevated_privilege",
        "status": 403,
      }
    }
  ]
}
```

account_cancelled
Returned when your account has been cancelled due to non-payment or some other reason. Please contact support if you think this is a mistake

```
{
  "errors": [
    {
          ... other fields for error
      "message": "Your account is inactive. If this is not expected,
please contact support",
      "code": "account_cancelled",
      "extensions": {
        "code": "account_cancelled",
        "status": 403,
      }
    }
  ]
}
```

args_required
Returned when your query or mutation is missing one or more required arguments. The property extesions.metadata.fields will provide the list of fields that have this constraints

```
{
  "errors": [
    {
          ... other fields for error
      "message": "You must provide one of the following: mine,
transcript_id, my_team",
      "code": "args_required",
      "extensions": {
        "code": "args_required",
        "status": 400,
        "metadata": {
          "fields": [
            "mine",
            "transcript_id",
            "my_team"
          ]
        },
      }
  }
 ]
}
```

too_many_requests
Returned when you have been rate-limited due to making too many requests. The field extensions.metadata.retryAfter mentions the retryAfter time

```
{
 "errors": [
  {
     ... other fields for error
   "message": "Too many requests. Please retry after 2:45:45 AM
(UTC)",
   "code": "too_many_requests",
   "extensions": {
    "code": "too_many_requests",
    "status": 429,
    "metadata": {
     "retryAfter": 1720651545066
    }
   }
  }
 ]
}
```

payload_too_small
Returned when the content size for uploadAudio mutation is too small. Upload files larger than 500kb to avoid this error

```
{
 "errors": [
  {
     ... other fields for error
   "message": "Content size is too small. Please upload files
larger than 500kb",
   "code": "payload_too_small",
   "extensions": {
    "code": "payload_too_small",
    "status": 400,
   }
  }
 ]
}
```

request_timeout

Returned when your request has taken too long to respond.

```
{
  "errors": [
    {
          ... other fields for error
      "message": "Request timed out. Please try again or contact
support",
      "code": "request_timeout",
      "extensions": {
        "code": "request_timeout",
        "status": 408,
      }
    }
  ]
}
```

invalid_language_code

Returned when an invalid language code has been passed to a query or mutation

```
{
  "errors": [
    {
          ... other fields for error
      "message": "Language code is invalid or not supported. Please
refer to API docs for supported languages",
      "code": "invalid_language_code",
      "extensions": {
        "code": "invalid_language_code",
        "status": 400,
      }
    }
  ]
}
```

admin_must_exist

Returned when you are attempting to call setUserRole for a single member team

```
{
  "errors": [
    {
          ... other fields for error
      "message": "You must have at least one admin your team",
      "code": "admin_must_exist",
      "extensions": {
        "code": "admin_must_exist",
        "status": 400,
      }
    }
  ]
}
```

Was this page helpful?

Fireflies home pagedark logo

Search or ask...
⌘K
Support
Dashboard

Additional Info

Changelog

Recent updates to the Fireflies API

Overview

This document maintains a chronologically ordered list of notable changes for each version of the Fireflies API. It’s designed to make it easier for you to keep track of new features, improvements, and bug fixes.

2.3.11

Fixed bug in transcript.sentences query where it was incorrectly returning null

Made improvements to the performance and stability of the Fireflies API.

2.3.10

Added field attendees to addToLive mutation

2.3.9

Added field meeting_info for meeting metadata.

Updated uploadAudio mutation to allow saving video

2.3.8

Added field speakers in transcript.

Updated docs to include error codes and their explanations. View details here

2.3.7

Added new fields for transcript.summary. View details on the Summary schema page

2.3.6

Updated docs for webhooks to include webhook schema

Added client_reference_id field for audio upload

2.3.5

Added argument custom_language for uploadAudio.

2.3.4

Added query argument fromDate and toDate for transcripts. View more details here. Field date has been deprecated in favor of these arguments

2.3.3

Added mutation for addToLiveMeeting. View more details here

2.3.2

Made improvements to the performance and stability of the Fireflies API.

2.3.1

Added field calendar_id. This field represents calId for google calendar and iCalUID for outlook calendar. For more details, view Transcript

2.3.0

Added queries for Transcript Summary. For more details, view Summary schema and Transcript schema

2.2.0

Added queries for bites and bite. For more details, view Bites schema

Added mutation for Create Bite that allows you to progmatically create bites. For more details, view CreateBite mutation

2.1.1

Fixed bugs in Transcripts query

2.1.0

Added video_url field for Transcript schema

2.0.7

Made improvements to the performance and stability of the Fireflies API.

2.0.6

Made improvements to the performance and stability of the Fireflies API.

2.0.5

Fixed bug in in Transcripts query for fireflies_users field

2.0.4

Fixed bug in in Transcripts query for audio_url field

2.0.3

Made improvements to the performance and stability of the Fireflies API.

2.0.2

Fixed bug in Transcripts query arguments

2.0.1

Fixed schema inconsistency in AudioUpload mutation.

Fixed schema inconsistency in SetUserRole mutation

2.0.0

Field transcript/host_email has been deprecated. More details in Transcript