

WP4 D3 Application Programming Interface for Scribe

Version 1

Date: 14 November 2025

Author: Andre Bester

Reviewed by: Jan-Willem van 't Klooster



Co-funded by
the European Union

Table of Contents

Overview of Scribe API	3
Sessions	3
Files	5
Processing	7
Analyses	10
Codebooks	13
Users	14
Index	<i>Error! Bookmark not defined.</i>

Overview of Scribe API

The Scribe API provides endpoints to the Scribe web application to do the following:

- Create, edit and delete sessions
- Retrieve user specific collections of sessions
- Upload and download files associated with a session
- Delete files associated with a session
- Retrieve a specific file
- Create, edit and delete analyses associated with a session
- Retrieve a specific analysis associated with a session
- Request transcription of an audio or video file into text
- Check the transcription status of a file
- Create, edit and delete codebooks
- Associate user data with sessions

Authentication is done on the endpoints by middleware, so that these endpoints are only accessible to authorized users subject to a timeout limit.

Sessions

The sessions endpoints are listed below with their associated HTTP methods.

/sessions	<ul style="list-style-type: none">• GET: retrieve a list of sessions RESPONSE: application/json<pre>{ "status": "string", "result": [{ "session": "string", "sessionFileAmount": 0, "sessionAnalysisAmount": 0 }] }</pre>
-----------	---

	<ul style="list-style-type: none"> • POST/PUT: create/update a session REQUEST/RESPONSE: application/json <pre> { "status": "string", "result": { "_id": "string", "name": "string", "description": "string", "researcher": { "_id": "string", "name": "string", "email": "string" }, "project_id": "string", "users": [{ "name": "string", "email": "string" }] } } </pre>
/sessions/{session_id}	<ul style="list-style-type: none"> • GET: retrieve the session with session id = id RESPONSE: application/json <pre> { "status": "string", "result": { "_id": "string", "name": "string", "description": "string", "researcher": { "_id": "string", "name": "string", "email": "string" }, "project_id": "string", "users": [{ "name": "string", "email": "string" }] } } </pre>

/sessions/user/{user_email}	<ul style="list-style-type: none"> GET: retrieve the sessions for a user with user_email = email RESPONSE: application/json
	<pre> { "status": "string", "result": [{ "session": "string", "sessionFileAmount": 0, "sessionAnalysisAmount": 0 }] } </pre>

Files

The files endpoints are listed below with their associated HTTP methods.

/files/session/{session_id}	<ul style="list-style-type: none"> GET: retrieve the files for the session with id = session_id RESPONSE: application/json <pre> { "status": "string", "result": [{ "_id": "string", "session_id": "string", "file_type": "image/png", "file_name": "string", "file_size": 0, "file_processing": [{ "type": "amberscript_transcription", "description": "string", "job_id": "string", "data": {} }] }] } </pre>
-----------------------------	--

/files/{file_id}	<ul style="list-style-type: none"> GET: retrieve the file with id = file_id RESPONSE: application/json
	<pre> { "status": "string", "result": [{ "_id": "string", "session_id": "string", "file_type": "image/png", "file_name": "string", "file_size": 0, "file_processing": [{ "type": "amberscript_transcription", "description": "string", "job_id": "string", "data": {} }] }] } </pre> <ul style="list-style-type: none"> DELETE: delete the file with file_id = id RESPONSE: application/json <pre> { "status": "string", "message": "string" } </pre>
/files/download/{file_id}	<ul style="list-style-type: none"> GET: download a file with file_id=id RESPONSE: filestream

/files/uploadfile	<ul style="list-style-type: none"> • POST: upload a file REQUEST: multipart file data with form value session_id RESPONSE: application/json <pre>{ "status": "string", "result": { "_id": "string", "session_id": "string", "file_type": "image/png", "file_name": "string", "file_size": 0, "file_processing": [{ "type": "amberscript_transcription", "description": "string", "job_id": "string", "data": {} }] } }</pre>
	<pre>] } }</pre>

Processing

The processing endpoints are listed below with their associated HTTP methods.

/processing/transcribe	<ul style="list-style-type: none"> GET: request the transcription of an audio file PARAMETERS: file_id, language, number of speakers RESPONSE: application/json <pre>{ "status": "string", "result": { "_id": "string", "session_id": "string", "file_type": "mp3", "file_name": "string", "file_size": 0, "file_processing": [{ "type": "transcription", "description": "string", "job_id": "string", "data": {} }] } }</pre>
/processing/transcribestatus/{id}	<ul style="list-style-type: none"> GET: retrieve the transcription status with job id = id RESPONSE: application/json <pre>{ Transcription service object containing status property }</pre>
/processing/transcription/{id}	<ul style="list-style-type: none"> GET: retrieve the transcription with job id = id RESPONSE: application/json <pre>{</pre>
	<pre> Transcription service object containing the transcription }</pre>

<p>/processing/filetimeseries/{file_id}</p>	<ul style="list-style-type: none"> GET: retrieve the timeseries of the file with id = file_id RESPONSE: application/json <pre>{ "status": "string", "result": { "starttime": 0, "numsamples": 0, "samplerate": 0, "datasamples": [] } }</pre>
<p>/processing/timeseries</p>	<ul style="list-style-type: none"> POST: create a new timeseries based on interpolation from the file identified by file_id REQUEST: application/json <pre>{ "file_id": "string", "session_id": "string", "starttime": 0, "numsamples": 0, "samplerate": 0 }</pre> RESPONSE: application/json <pre>{ "status": "string", "result": [] }</pre>
<p>/processing/baseline</p>	<ul style="list-style-type: none"> POST: create a new timeseries based on interpolation and mapped onto time baseline segments from the file identified by file_id REQUEST: application/json <pre>{ "file_id": "string", "session_id": "string", "baseline": [{ "starttime": 0, "endtime": 0, "value": 0 }] }</pre>

	<pre> RESPONSE: application/json { "status": "string", "result": [{ "starttime": 0, "endtime": 0, "value": 0 }] } </pre>
--	--

Analyses

The analyses endpoints are listed below with their associated HTTP methods.

/analyses	<ul style="list-style-type: none"> • GET: retrieve a list of analyses RESPONSE: application/json List of analyses • POST/PUT: create/update an analysis REQUEST: application/json <pre> { "_id": "string", "name": "string", "type": "none", "description": "none", "session_id": "string", "owner": { "name": "string", "email": "string" }, "coders": [{ "name": "string", "email": "string" }], "files": ["string"], "codebook": [], "table": { "locked": true, "columns": [{}], </pre>
-----------	--

```
"rows": [ {}
}
}
RESPONSE: application/json
{
  "status": "string",
  "result": {
    "_id": "string",
    "name": "string",
    "type": "none",
    "description": "none",
    "session_id": "string",
    "owner": {
      "name": "string",
      "email": "string"
    },
    "coders": [
      {
        "name": "string",
        "email": "string"
      }
    ],
    "files": [
      "string"
    ],
    "codebook": [ ],
    "table": {
      "locked": true,
      "columns": [ {} ],
      "rows": [ {} ],
    }
  }
}
```

<p>/analyses/{analysis_id}</p>	<ul style="list-style-type: none"> • GET: retrieve the analysis with analysis_id = id RESPONSE: application/json <pre>{ "status": "string", "result": { "_id": "string", "name": "string", "type": "none", "description": "none", date", "session_id": "string", "owner": { "name": "string", "email": "string" } }, </pre>
	<pre> "coders": [{ "name": "string", "email": "string" }], "files": ["string"], "codebook": [], "table": { "locked": true, "columns": [{}], "rows": [{}], } } } </pre> <ul style="list-style-type: none"> • DELETE: delete the analysis with analysis_id = id RESPONSE: application/json <pre>{ "status": "string", "message": "string" }</pre>

/analyses/session/{session_id}	<ul style="list-style-type: none"> • GET: retrieve the analysis list with session_id = id RESPONSE: application/json List of analysis from the session
--------------------------------	---

Codebooks

The files endpoints are listed below with their associated HTTP methods.

/codebooks	<ul style="list-style-type: none"> • GET: retrieve a list of codebooks RESPONSE: application/json <pre>{ "status": "string", "result": [</pre>
	<pre> { "_id": "string", "name": "string", "codes": [{ "code": "string", "definition": "string", "examples": [], "children": [] }], }]</pre>

	<ul style="list-style-type: none"> • POST/PUT: create/update a codebook REQUEST: application/json <pre> { "_id": "string", "name": "string", "codes": [{ "code": "string", "definition": "string", "examples": [], "children": [] }], } </pre> RESPONSE: application/json <pre> { "status": "string", "result": { "_id": "string", "name": "string", "codes": [{ "code": "string", "definition": "string", "examples": [], "children": [] }], } } </pre>
/codebooks/{codebook_id}	

Users

The users endpoints are listed below with their associated HTTP methods.

/users	<ul style="list-style-type: none"> • GET: retrieve a list of users RESPONSE: application/json <pre>{ "status": "string", "result": [{ "_id": "string", "firstname": "string", "surname": "string", "email": "string" }] }</pre> • POST/PUT: create/update a user REQUEST: application/json <pre>{ "_id": "string", "firstname": "string", "surname": "string", "email": "string" }</pre> RESPONSE: application/json <pre>{ "status": "string", "result": { "_id": "string", "firstname": "string", "surname": "string", "email": "string" } }</pre>
/users/{user_id}	<ul style="list-style-type: none"> • GET: retrieve the user with user_id = id RESPONSE: application/json <pre>{ "status": "string", "result": { "_id": "string", "firstname": "string", "surname": "string", "email": "string" } }</pre>

	<ul style="list-style-type: none"> DELETE: delete the user with user_id = id RESPONSE: application/json <pre>{ "status": "string", "message": "string" }</pre>
/users/email/{user_email}	<ul style="list-style-type: none"> GET: retrieve the user with user_email = email RESPONSE: application/json <pre>{ "status": "string", "result": { "_id": "string", "firstname": "string", "surname": "string", "email": "string" } }</pre>