

SpeechExec Enterprise App Interface service interface specification

Contents

Change history	5
1 Dictation API for mobile clients	6
1.1 Request authentication.....	6
1.1.1 POST /app/token: acquire an access token	6
1.1.1.1 Request specification	6
1.1.1.2 Parameters.....	6
1.1.1.3 Request body	6
1.1.1.4 Request body example.....	6
1.1.1.5 Responses	6
1.1.1.5.1 Token request is successful	6
1.1.1.5.2 Error responses.....	7
1.1.2 Specifying the authentication token in a custom request header (<code>x-sps-app-interface-authorization</code>).....	7
1.2 Generic authentication error responses.....	8
1.3 POST /app/dictations: upload new dictation.....	8
1.3.1 Request specification	8
1.3.2 Parameters.....	9
1.3.3 Request body	9
1.3.4 Request body validation	9
1.3.5 Request body example.....	10
1.3.6 Responses	11
1.3.6.1 Dictation upload is successful	11
1.3.6.2 Error responses	11
1.4 GET /app/dictations/{DictationID}: get information for one dictation.....	12
1.4.1 Request specification	12
1.4.2 Parameters.....	13
1.4.3 Request body	13
1.4.4 Request URL example	13
1.4.5 Responses	13
1.4.5.1 Dictation information is found.....	13
1.4.5.2 Error responses	13
1.5 GET /app/dictations: get information for all dictations in a folder	14
1.5.1 Request specification	14
1.5.2 Parameters.....	14

1.5.3	Request body	14
1.5.4	Request URL example	14
1.5.5	Responses	15
1.5.5.1	Querying dictations is successful	15
1.5.5.2	Error responses	15
1.6	POST /app/getdictationinfolist: get information for a list of dictations	15
1.6.1	Request specification	15
1.6.2	Parameters	16
1.6.3	Request body	16
1.6.4	Request body validation	16
1.6.5	Request body example.....	16
1.6.6	Responses	16
1.6.6.1	Querying dictations is successful	16
1.6.6.2	Error responses	17
1.7	GET /app/dictations/{DictationID}/attachment/download: download attachment of one dictation	17
1.7.1	Request specification	17
1.7.2	Parameters	18
1.7.3	Request body	18
1.7.4	Request URL example	18
1.7.5	Responses	18
1.7.5.1	Attachment download is successful.....	18
1.7.5.2	Error responses	18
1.8	GET /app/usersettings: Retrieve user settings	19
1.8.1	Request specification	19
1.8.2	Parameters	19
1.8.3	Request body	19
1.8.4	Request URL example	19
1.8.5	Responses	19
1.8.5.1	User settings are found.....	19
1.8.5.1.1	Returned speech recognition language codes.....	19
1.8.5.2	Error responses	20
2	MasterData API	21
2.1	POST /masterdata/dataitems: insert new master data into database.....	21
2.1.1	Request specification	21
2.1.2	Parameters.....	21

2.1.3	Request body	21
2.1.4	Request body example.....	21
2.1.5	Responses	22
2.1.5.1	Insert master data is successful	22
2.1.5.2	Error responses	22
3	Document Management System API	23
3.1	GET /dms/users: Retrieve all author users	23
3.1.1	Request specification	23
3.1.2	Parameters.....	23
3.1.3	Request body	23
3.1.4	Request URL example	23
3.1.5	Responses	23
3.1.5.1	Querying users is successful.....	23
3.1.5.2	Error responses	24
3.2	GET /dms/dictations/{AuthorName}: get information for all dictations belonging to a specific author.....	24
3.2.1	Request specification	24
3.2.2	Parameters.....	24
3.2.3	Request body	25
3.2.4	Request URL example	25
3.2.5	Responses	25
3.2.5.1	Querying dictations is successful	25
3.2.5.2	Error responses	25
3.3	GET /dms/dictations: get information for all dictations belonging to all authors.....	26
3.3.1	Request specification	26
3.3.2	Parameters.....	26
3.3.3	Request body	26
3.3.4	Request URL example	26
3.3.5	Responses	26
3.3.5.1	Querying dictations is successful	26
3.3.5.2	Error responses	27
3.4	GET /dms/dictations/{AuthorName}/{DictationID}/attachment: download attachment of one dictation.....	27
3.4.1	Request specification	27
3.4.2	Parameters.....	27
3.4.3	Request body	28

- 3.4.4 Request URL example 28
- 3.4.5 Responses 28
 - 3.4.5.1 Attachment download is successful..... 28
 - 3.4.5.2 Error responses 28
- 4 Transcription Service API 29
- 5 Diagnostics endpoints 29
 - 5.1 Request authentication..... 29
 - 5.2 GET /test/testconfig..... 29
 - 5.2.1 Request specification 29
 - 5.2.2 Parameters 29
 - 5.2.3 Request body 29
 - 5.2.4 Request URL example 29
 - 5.2.5 Responses 29
 - 5.2.5.1 All test steps are successful 29
 - 5.2.5.2 Error responses 30

Change history

Application version	Change date	Description
SEE 8.0	2022.01.24	- Added spec. for 'x-sps-app-interface-authorization' custom header
	2021.12.08	- Added spec. and possible values for SR language codes - Added spec for getting info for multiple dict. IDs
	2021.11.29	- Added description of generic authentication responses - For the uploaded file item, no longer require Content-Type: application/octet-stream
	2021.10.07	Initial version

1 Dictation API for mobile clients

1.1 Request authentication

1.1.1 POST /app/token: acquire an access token

1.1.1.1 Request specification

Purpose:

Authenticates the Enterprise Author user with the provided username and password and returns an OAuth access token.

HTTP method:

POST

URL template:

/app/token

Request headers:

No special headers.

1.1.1.2 Parameters

No special request parameters.

1.1.1.3 Request body

The following items must be defined as x-www-form-urlencoded. All of them are mandatory.

Item name	Value
grant_type	password
username	<username>
password	<password>

1.1.1.4 Request body example

The following example was created using Postman:

The screenshot shows a Postman interface for a POST request to `http://localhost/SEEIntegratorAPI/app/token`. The 'Body' tab is selected, and the encoding is set to 'x-www-form-urlencoded'. The request body is defined as follows:

KEY	VALUE
<input checked="" type="checkbox"/> grant_type	password
<input checked="" type="checkbox"/> username	seagile1
<input checked="" type="checkbox"/> password	*****

1.1.1.5 Responses

1.1.1.5.1 Token request is successful

Description:

User authentication/login was successful and an access token was generated.

Code:

200

Response body / value:

```
{
  "access_token": "<encoded_access_token_value>",
  "token_type": "bearer",
  "expires_in": <seconds_value>,
  "userName": "<user_name>",
  "roles": "<user_role_01>,<user_role_02>,...",
  "userAccountId": "<encoded_domain_and_username>",
  ".issued": "<UTC_issue_timestamp>",
  ".expires": "<UTC_expire_timestamp>"
}
```

1.1.1.5.2 Error responses

Code	Description	Response body / value
400	grant_type is not "password"	{ "error": "unsupported_grant_type" }
400	Username or password does not match	{ "error": "invalid_grant", "error_description": "Wrong_username_or_password" }
400	No such user found in SEERoot repository	{ "error": "invalid_grant", "error_description": "Enterprise_user_not_found" }
400	EMS license is not available for the user	{ "error": "invalid_grant", "error_description": "License_not_available" }

1.1.2 Specifying the authentication token in a custom request header (x-sps-app-interface-authorization)

Usually, the access token acquired via the POST /app/token (1.1.1) request is used as follows:

- Prepare a sub-sequent /app/* http request
- Use access token as value of the Authorization header
- Send the request

In certain special scenarios, however, the Authorization header must be used for a different purpose. For example, to allow communication go through an Azure App Proxy, the Authorization header of the request must contain the access token of the Azure App Proxy, effectively prohibiting successful authentication into Enterprise App Interface itself.

Since the Authorization header cannot contain multiple tokens, the access token of Enterprise App Interface must be stored in a special, custom request header as follows:

- Header name:
 x-sps-app-interface-authorization

- Header value:
Bearer <access_token_value>

Whenever the Enterprise App Interface web service receives a HTTP request where the x-sps-app-interface-authorization header is present, the service will:

- Read the entire value of the x-sps-app-interface-authorization header
- overwrite the Authorize header value
- continue request processing and eventually perform authentication checks

1.2 Generic authentication error responses

The following error responses may occur in all dictation API endpoints (using OAuth authentication):

Code	Description	Response body / value
401	Invalid header in the request	{ "CRI": "<GUID>", "FailureCode": "Invalid_action_context_request_or_request_header" }
401	No Authorization header found in the request	{ "CRI": "<GUID>", "FailureCode": "Authorization_header_not_found_in_the_request" }
401	Authorization header is not Bearer in the request	{ "CRI": "<GUID>", "FailureCode": "Invalid_Authorization_header_value_in_the_request" }
401	The required user role is not found in the access token	{ "CRI": "<GUID>", "FailureCode": "Invalid_user_role" }
401	Unknown error occurred during authentication	{ "CRI": "<GUID>", "FailureCode": "Unknown_error" }

1.3 POST /app/dictations: upload new dictation

1.3.1 Request specification

Purpose:

Upload a new dictation file and store it in the Finished folder of the authenticated Author user.

HTTP method:

POST

URL template:

/app/dictations

Request headers:

Header name	Value
-------------	-------

Authorization	Bearer <encoded_access_token_value>
Content-Type	multipart/form-data;

1.3.2 Parameters

No special request parameters.

1.3.3 Request body

The following items must be defined as form-data. All of them are mandatory.

Item name	Value
Content-Disposition	form-data; name="see_dictation_audio_file"; filename="<audiofile>.<ext>"
DictationId	string value
Status	int value
Priority	int value
Worktype	string value
CreationDateTime	datetime value
Title	string value
Category	string value
Comments	string value
Department	string value
Keyword	string value
DPMBarcode	string value
Custom1	string value
Custom2	string value
Custom3	string value
Custom4	string value
Custom5	string value

The following items are optional.

Item name	Value
NumberOfSpeakers	int value
SELiveSpeechRecLanguageCode6Char	string value

1.3.4 Request body validation

DictationId:

- must not be empty
- must have a valid Guid structure
- must not be empty Guid

Priority:

- must be an integer number

Status:

- must be an integer number between 0 and 14 (dictation state values)
- Note: more important dictation states:
 - o 2 (TRANSCRIPTION_PENDING)
 - o 11 (QUEUED_FOR_TRANSCRIPTION_SERVICE)
 - o 13 (QUEUED_FOR_SPEECHLIVE_SPEECH_RECOGNITION)

CreationDateTime:

- must be a valid DateTime with the following format:
yyyy-MM-dd'T'HH:mm:ss.FFF

The optional items are only validated when provided:

NumberOfSpeakers:

- 1 (One)
- 2 (Two)
- 3 (Three)
- 4 (FourToFive)
- 5 (SixAndMore)

SELiveSpeechRecLanguageCode6Char:

- must not be empty
- its length must exactly be 7 characters
- must have the following structure: <3_chars>-<3_chars>
e.g. eng-GBR

IMPORTANT:

When the provided value of Status is 11 (QUEUED_FOR_TRANSCRIPTION_SERVICE), the NumberOfSpeakers item must also be provided.

When the provided value of Status is 13

(QUEUED_FOR_SPEECHLIVE_SPEECH_RECOGNITION), the SELiveSpeechRecLanguageCode6Char item must also be provided.

1.3.5 Request body example

The following example was created using Postman:

POST ▼ http://localhost/SEEIntegratorAPI/app/dictations ...

Params Authorization ● Headers (9) **Body ●** Pre-request Script Tests

● none ● **form-data** ● x-www-form-urlencoded ● raw ● binary ● GraphQL

	KEY	VALUE
<input checked="" type="checkbox"/>	see_dictation_audio_file	Tim_0006.DS2 ×
<input checked="" type="checkbox"/>	DictationId	A1387477-1C7F-4E5D-90E9-7724E3C4A4E6
<input checked="" type="checkbox"/>	Status	2
<input checked="" type="checkbox"/>	Priority	0
<input checked="" type="checkbox"/>	Worktype	Memo
<input checked="" type="checkbox"/>	CreationDateTime	2021-08-03T13:55:15.666
<input checked="" type="checkbox"/>	Title	My first dictation
<input checked="" type="checkbox"/>	Category	
<input checked="" type="checkbox"/>	Comments	
<input checked="" type="checkbox"/>	Department	
<input checked="" type="checkbox"/>	Keyword	
<input checked="" type="checkbox"/>	DPMBarcode	
<input checked="" type="checkbox"/>	Custom1	
<input checked="" type="checkbox"/>	Custom2	
<input checked="" type="checkbox"/>	Custom3	
<input checked="" type="checkbox"/>	Custom4	
<input checked="" type="checkbox"/>	Custom5	

1.3.6 Responses

1.3.6.1 Dictation upload is successful

Description:

Dictation upload is completed, the dictation file is created.

Code:

201

Response body / value:

No response body.

1.3.6.2 Error responses

Code	Description	Response body / value
400	DictationId form-data is empty, missing or otherwise invalid	{ "CRI": "<GUID>", "FailureCode": "ID_not_valid" }

400	One of the required ContentDisposition items is invalid	{ "CRI": "<GUID>", "FailureCode": "ContentDisposition_not_valid" }
400	Audio file name or extension is invalid or not supported	{ "CRI": "<GUID>", "FailureCode": "Audio_filename_not_valid" }
400	One of the required form-data fields is invalid or missing	{ "CRI": "<GUID>", "FailureCode": "Metadata_not_valid_or_missing" }
400	Service couldn't construct a valid Enterprise dictation object from the received audio and XML file	{ "CRI": "<GUID>", "FailureCode": "Invalid_dictation_files" }
401	Unauthorized	{ "Message": "Authorization has been denied for this request." }
409	Target folder contains a dictation file with the same name	{ "CRI": "<GUID>", "FailureCode": "Dictation_file_already_exists" }
409	Service cannot create the target folder for dictations	{ "CRI": "<GUID>", "FailureCode": "Cannot_create_final_directory" }
415	Content-Type is not "multipart/form-data" in the request header	{ "CRI": "<GUID>", "FailureCode": "" }
500	Any other error encountered during request processing	{ "CRI": "<GUID>", "FailureCode": "Generic_error" }

1.4 GET /app/dictations/{DictationID}: get information for one dictation

1.4.1 Request specification

Purpose:

Retrieve metadata information of the dictation having {DictationID}.

HTTP method:

GET

URL template:

/app/dictations/{DictationID}

Request headers:

Header name	Value
Authorization	Bearer <encoded_access_token_value>

1.4.2 Parameters

Parameter name	Description
DictationID	URL path parameter. The dictation is searched by this ID. This parameter is mandatory.

1.4.3 Request body

No request body is needed.

1.4.4 Request URL example

/app/dictations/3D8A776F-86FF-4050-96EA-1C325DBF8834

1.4.5 Responses

1.4.5.1 *Dictation information is found*

Description:

Dictation information for the passed DictationID is found.

Code:

200

Response body / value:

```
{
  "CRI": "<GUID>",
  "data": [
    {
      "files": [ "only 1 <cache_item> entry" ]
    }
  ]
}
```

The cache item entry has the following JSON structure:

```
{
  "DictationID": "<GUID>",
  "AudioFileName": "DictFileName.ext",
  "DictationState": "<CurrentDictState>",
  "HasAttachment": true|false,
  "IsArchive": true|false,
  "SubfolderHierarchy":
  "\sub\folder\hierarchy\under\finished\or\archive",
  "LastChangedDateTimeUtc": "yyyy-MM-dd HH:mm:ss.fffff"
}
```

1.4.5.2 *Error responses*

Code	Description	Response body / value
400	DictationID is empty, missing or otherwise invalid in request URL	{ "CRI": "<GUID>",

		<pre>"FailureCode": "ID_not_valid" }</pre>
401	Unauthorized	<pre>{ "Message": "Authorization has been denied for this request." }</pre>
404	Dictation information cannot be found for the specified DictationID	<pre>{ "CRI": "<GUID>", "FailureCode": "Dictation_id_not_found" }</pre>
500	Any other error encountered during request processing	<pre>{ "CRI": "<GUID>", "FailureCode": "Generic_error" }</pre>

1.5 GET /app/dictations: get information for all dictations in a folder

1.5.1 Request specification

Purpose:

Retrieve metadata information of all dictations found in the folder specified by {isArchive}.

HTTP method:

GET

URL template:

/app/dictations?[isArchive=false|true]

Request headers:

Header name	Value
Authorization	Bearer <encoded_access_token_value>

1.5.2 Parameters

Parameter name	Description
isArchive	<p>Query parameter.</p> <p>Specify the folder to retrieve dictation metadata from.</p> <ul style="list-style-type: none"> - true: the authenticated user's Archive folder - false: the authenticated user's Finished dictations folder <p>This parameter is optional. Default if omitted: FALSE</p>

1.5.3 Request body

No request body is needed.

1.5.4 Request URL example

/app/dictations?isArchive=true

1.5.5 Responses

1.5.5.1 Querying dictations is successful

Description:

There were no errors when retrieving dictation metadata. Even if the required folder is empty and no dictation metadata was found, a success result is returned with an empty list.

Code:

200

Response body / value:

```
{
  "CRI": "GUID",
  "data": [
    {
      "files": [ "array of <cache_item> entries" ]
    }
  ]
}
```

The cache item entry has the following JSON structure:

```
{
  "DictationID": "<GUID>",
  "AudioFileName": "DictFileName.ext",
  "DictationState": "<CurrentDictState>",
  "HasAttachment": true|false,
  "IsArchive": true|false,
  "SubfolderHierarchy":
  "\sub\folder\hierarchy\under\finished\or\archive",
  "LastChangedDateTimeUtc": "yyyy-MM-dd HH:mm:ss.ffff"
}
```

1.5.5.2 Error responses

Code	Description	Response body / value
401	Unauthorized	{ "Message": "Authorization has been denied for this request." }
500	Any other error encountered during request processing	{ "CRI": "<GUID>", "FailureCode": "Generic_error" }

1.6 POST /app/getdictationinfolist: get information for a list of dictations

1.6.1 Request specification

Purpose:

Retrieve metadata information for all dictations specified in the request.

HTTP method:

POST

URL template:

/app/getdictationinfolist

Request headers:

Header name	Value
Authorization	Bearer <encoded_access_token_value>

1.6.2 Parameters

No special request parameters.

1.6.3 Request body

The following JSON string must be specified.

```
{
  "CRI": "GUID",
  "dictationIds": [
    "3fa85f64-5717-4562-b3fc-2c963f66afa6"
  ]
}
```

1.6.4 Request body validation

CRI:

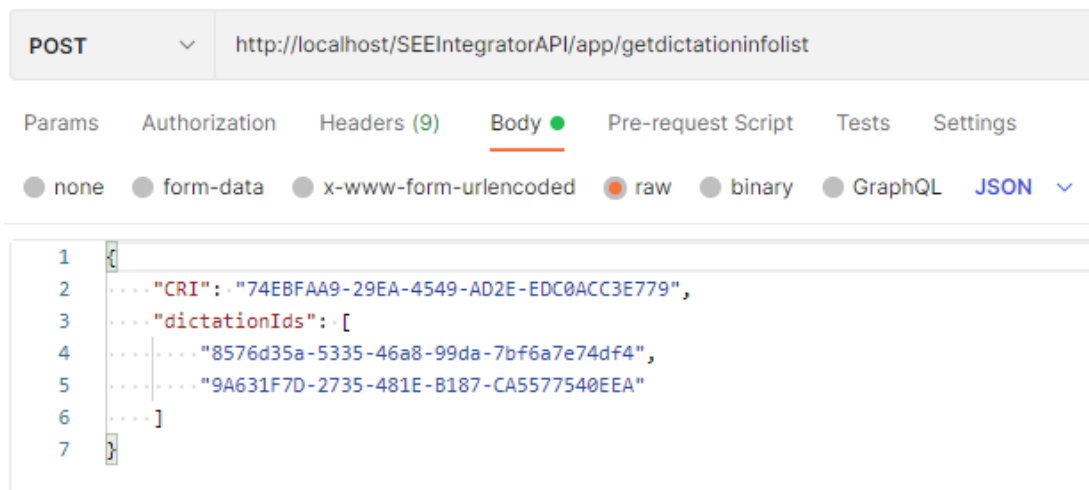
- optional, used for server logging
- recommended to be specified

dictationIds:

- must not be empty
- must contain at least 1 item

1.6.5 Request body example

The following example was created using Postman:



1.6.6 Responses

1.6.6.1 Querying dictations is successful

Description:

There were no errors when retrieving dictation metadata. Even if no dictation metadata was found, a success result is returned with an empty list.

Code:

200

Response body / value:

```
{
  "CRI": "GUID",
  "data": [
    {
      "files": [ "array of <cache_item> entries" ]
    }
  ]
}
```

The cache item entry has the following JSON structure:

```
{
  "DictationID": "<GUID>",
  "AudioFileName": "DictFileName.ext",
  "DictationState": "<CurrentDictState>",
  "HasAttachment": true|false,
  "IsArchive": true|false,
  "SubfolderHierarchy":
  "\sub\folder\hierarchy\under\finished\or\archive",
  "LastChangedDateTimeUtc": "yyyy-MM-dd HH:mm:ss.fffff"
}
```

1.6.6.2 Error responses

Code	Description	Response body / value
400	dictationIds list is empty, missing or otherwise invalid in the request	{ "CRI": "<GUID>", "FailureCode": "ID_not_valid" }
401	Unauthorized	{ "Message": "Authorization has been denied for this request." }
500	Any other error encountered during request processing	{ "CRI": "<GUID>", "FailureCode": "Generic_error" }

1.7 GET /app/dictations/{DictationID}/attachment/download: download attachment of one dictation

1.7.1 Request specification

Purpose:

Download attachment of the dictation having {DictationID}.

HTTP method:

GET

URL template:

/app/dictations/{DictationID}/attachment/download

Request headers:

Header name	Value
Authorization	Bearer <encoded_access_token_value>

1.7.2 Parameters

Parameter name	Description
DictationID	URL path parameter. The dictation is searched by this ID.

1.7.3 Request body

No request body is needed.

1.7.4 Request URL example

/app/dictations/3D8A776F-86FF-4050-96EA-1C325DBF8834/attachment/download

1.7.5 Responses

1.7.5.1 Attachment download is successful

Description:

Attachment file of the dictation having DictationID is found and downloaded.

Code:

200

Response headers:

Header name	Value
Content-Type	application/octet-stream
Content-Disposition	see_dictation_attachment; filename=<AudioFilename>.<ext1>.<ext2>
Content-Length	<Attachment_filesize_in_bytes>

Response body / value:

The response body contains the file content of the attachment in binary format.

1.7.5.2 Error responses

Code	Description	Response body / value
400	DictationID is empty, missing or otherwise invalid in request URL	{ "CRI": "<GUID>",&br/> "FailureCode": "ID_not_valid" }
401	Unauthorized	{ "Message": "Authorization has been denied for this request." }
404	Dictation information cannot be found for the specified DictationID	{ "CRI": "<GUID>",&br/> "FailureCode": "Dictation_id_not_found" }
404	Dictation information is available,	{

	but the dictation has no attachment	"CRI": "<GUID>", "FailureCode": "Attachment_not_found" }
500	Any other error encountered during request processing	{ "CRI": "<GUID>", "FailureCode": "Generic_error" }

1.8 GET /app/usersettings: Retrieve user settings

1.8.1 Request specification

Purpose:

Retrieve the SpeechService user settings of the authenticated Author user.

HTTP method:

GET

URL template:

/app/usersettings

Request headers:

Header name	Value
Authorization	Bearer <encoded_access_token_value>

1.8.2 Parameters

No special request parameters.

1.8.3 Request body

No request body is needed.

1.8.4 Request URL example

/app/usersettings

1.8.5 Responses

1.8.5.1 User settings are found

Description:

SpeechService user settings of the authenticated Author user have been found.

Code:

200

Response body / value:

```
{
  "IsNdevSrEnabled": true|false,
  "PreferredNdevLanguageCodes": " AllowedSRLanguageCodes_list",
  "SupportedNdevLanguageCodes": " AllowedSRLanguageCodes_list",
  "IsTranscriptionServiceEnabled": true|false,
  "IsMultiSpeakerTranscriptionEnabled": true|false
}
```

1.8.5.1.1 Returned speech recognition language codes

The "PreferredNdevLanguageCodes" and "SupportedNdevLanguageCodes" lists:

- Are semi-colon (“;”) separated
- Contain the same values

Possible values that can be returned are shown in the ‘6 char’ column of the following table:

Language	6 char *
Czech	ces-CZE
Danish	dan-DNK
Dutch	nld-NLD
English (Australia)*	eng-AUS
English (GB)*	eng-GBR
English (US)*	eng-USA
Finnish	fin-FIN
French (Canada)	fra-CAN
French (France)*	fra-FRA
German*	deu-DEU
Hungarian	hun-HUN
Italian	ita-ITA
Norwegian	nor-NOR
Polish	pol-POL
Portuguese (Portugal)	por-PRT
Russian	rus-RUS
Slovak	slk-SVK
Spanish (Spain)	spa-ESP
Spanish (LatAm)	spa-XLA
Swedish	swe-SWE
Turkish	tur-TUR

1.8.5.2 Error responses

Code	Description	Response body / value
401	Unauthorized	{ "Message": "Authorization has been denied for this request." }
500	Any other error encountered during request processing	{ "CRI": "<GUID>", "FailureCode": "Generic_error" }

2 MasterData API

2.1 POST /masterdata/dataitems: insert new master data into database

2.1.1 Request specification

Purpose:

Insert a new row into the master data SQL database.

HTTP method:

POST

URL template:

/masterdata/dataitems

Request headers:

Header name	Value
x-sps-api-key	<API_key>

2.1.2 Parameters

No special request parameters.

2.1.3 Request body

The request body must have the following JSON structure:

```
{
  "CRI": "GUID",
  "dataitem": {
    "ID": "New_ID_string",
    "Label01": "Label01Value",
    "Label02": "Label02Value",
    "Label03": "Label03Value",
    "Label04": "Label04Value",
    "Label05": "Label05Value",
    "Label06": "Label06Value",
    "Label07": "Label07Value",
    "Label08": "Label08Value",
    "Label09": "Label09Value",
    "Int01": Int01Value,
    "Int02": Int02Value,
    "Int03": Int03Value,
    "Int04": Int04Value,
    "Int05": Int05Value,
    "Datetime01": "Datetime01Value",
    "Datetime02": "Datetime02Value",
    "Datetime03": "Datetime03Value",
    "Datetime04": "Datetime04Value",
    "Datetime05": "Datetime05Value"
  }
}
```

2.1.4 Request body example

The following example was created using Postman:

```

POST http://localhost/SEEIntegratorAPI/masterdata/dataitems

Params Authorization Headers (9) Body Pre-request Script Tests Settings
● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL JSON

1 {
2   ... "CRI": "18E70035-1625-4750-B309-492320CFFF40",
3   ... "dataitem":
4     ... {
5       ... "ID": "5C10D427-E8E2-45FF-8AA8-584AC6DCC5BA",
6       ... "Label101": "data01",
7       ... "Label102": "data02",
8       ... "Label103": "data03",
9       ... "Label104": "data04",
10      ... "Label105": "data05",
11      ... "Label106": "data06",
12      ... "Label107": "data07",
13      ... "Label108": "data08",
14      ... "Label109": "data09",
15      ... "Int01": "1",
16      ... "Int02": "2",
17      ... "Int03": "3",
18      ... "Int04": "4",
19      ... "Int05": "5",
20      ... "Datetime01": "2021-09-08 16:20:56",
21      ... "Datetime02": "2017-01-25 23:56:05",
22      ... "Datetime03": "1985-05-13 01:19:11",
23      ... "Datetime04": "1999-08-01 08:38:59",
24      ... "Datetime05": "2005-12-31 13:37:41"
25    }
26  }

```

2.1.5 Responses

2.1.5.1 Insert master data is successful

Description:

Insert operation was successful, the new master data record is added to the database.

Code:

201

Response body / value:

No response body.

2.1.5.2 Error responses

Code	Description	Response body / value
400	ID is empty, missing or otherwise invalid in the request	{ "CRI": "<GUID>", "FailureCode": "ID_not_valid" }
400	Any DateTimeX value is non-empty and has wrong format	{ "CRI": "<GUID>", "FailureCode": "DateTime_value_not_valid" }

		}
401	Unauthorized	{ "CRI": "<GUID>", "FailureCode": "Invalid_api_key" }
409	ID already exists in database (primary key violation)	{ "CRI": "<GUID>", "FailureCode": "ID_already_exists" }
409	Any other SQL error not handled in a specific way	{ "CRI": "<GUID>", "FailureCode": "Generic_database_error" }
500	Any other error encountered during request processing	{ "CRI": "<GUID>", "FailureCode": "Generic_error" }

3 Document Management System API

3.1 GET /dms/users: Retrieve all author users

3.1.1 Request specification

Purpose:

Retrieve all existing author users including the ones not having an EMS CAL license.

HTTP method:

GET

URL template:

/dms/users

Request headers:

Header name	Value
x-sps-api-key	<API_key>

3.1.2 Parameters

No special request parameters.

3.1.3 Request body

No request body is needed.

3.1.4 Request URL example

/app/users

3.1.5 Responses

3.1.5.1 Querying users is successful

Description:

There were no errors when retrieving the list of users. Even if no users were found, a success result is returned with an empty list.

Code:

200

Response body / value:

```
{
  "CRI": "GUID",
  "users": [
    {
      "username": "UsernameX",
    }
  ]
}
```

3.1.5.2 Error responses

Code	Description	Response body / value
401	Unauthorized	{ "CRI": "<GUID>", "FailureCode": "Invalid_api_key" }
500	Any other error encountered during request processing	{ "CRI": "<GUID>", "FailureCode": "Generic_error" }

3.2 GET /dms/dictations/{AuthorName}: get information for all dictations belonging to a specific author**3.2.1 Request specification****Purpose:**

Retrieve metadata information of all dictations which

- belong to {AuthorName}
- have Transcription finished state
- have attached document

HTTP method:

GET

URL template:

/dms/dictations/{AuthorName}

Request headers:

Header name	Value
x-sps-api-key	<API_key>

3.2.2 Parameters

Parameter name	Description
AuthorName	URL path parameter. Those dictations are retrieved where the Author dictation property equals with this parameter.

3.2.3 Request body

No request body is needed.

3.2.4 Request URL example

/dms/dictations/user01

3.2.5 Responses

3.2.5.1 Querying dictations is successful

Description:

There were no errors when retrieving the list of dictations. Even if no dictations were found, a success result is returned with an empty list.

Code:

200

Response body / value:

```
{
  "CRI": "<GUID>",
  "data": [
    {
      "user": "author_name",
      "role": "0",
      "files": [ "array of <cache_item> entries" ]
    }
  ]
}
```

The cache item entry has the following JSON structure:

```
{
  "DictationID": "<GUID>",
  "AudioFileName": "DictFileName.ext",
  "DictationState": "<CurrentDictState>",
  "HasAttachment": true|false,
  "IsArchive": true|false,
  "SubfolderHierarchy":
  "\sub\folder\hierarchy\under\finished\or\archive",
  "LastChangedDateTimeUtc": "yyyy-MM-dd HH:mm:ss.fffff"
}
```

3.2.5.2 Error responses

Code	Description	Response body / value
400	AuthorName is empty, missing or otherwise invalid in request URL	{ "CRI": "<GUID>", "FailureCode": "Username_not_valid" }
400	User xml with author role is not available or not found	{ "CRI": "<GUID>", "FailureCode": "User_not_exists" }
401	Unauthorized	{ "CRI": "<GUID>", }

		"FailureCode": "Invalid_api_key" }
500	Any other error encountered during request processing	{ "CRI": "<GUID>", "FailureCode": "Generic_error" }

3.3 GET /dms/dictations: get information for all dictations belonging to all authors

3.3.1 Request specification

Purpose:

Retrieve metadata information of all dictations belonging to the all authors which

- have Transcription finished state
- have attached document

HTTP method:

GET

URL template:

/dms/dictations

Request headers:

Header name	Value
x-sps-api-key	<API_key>

3.3.2 Parameters

No special request parameters.

3.3.3 Request body

No request body is needed.

3.3.4 Request URL example

/dms/dictations

3.3.5 Responses

3.3.5.1 Querying dictations is successful

Description:

There were no errors when retrieving the list of authors or the list of dictations. Even if no author or no dictations were found, a success result is returned with empty lists.

Code:

200

Response body / value:

```
{
  "CRI": "<GUID>",
  "data": [
    {
      "user": "Authornamex",
      "role": "0",
      "files": [ "array of <cache_item> entries" ]
    }
  ]
}
```

```
]
}
```

The cache item entry has the following JSON structure:

```
{
  "DictationID": "<GUID>",
  "AudioFileName": "DictFileName.ext",
  "DictationState": "<CurrentDictState>",
  "HasAttachment": true|false,
  "IsArchive": true|false,
  "SubfolderHierarchy":
  "\sub\folder\hierarchy\under\finished\or\archive",
  "LastChangedDateTimeUtc": "yyyy-MM-dd HH:mm:ss.fffff"
}
```

3.3.5.2 Error responses

Code	Description	Response body / value
401	Unauthorized	{ "CRI": "<GUID>", "FailureCode": "Invalid_api_key" }
500	Any other error encountered during request processing	{ "CRI": "<GUID>", "FailureCode": "Generic_error" }

3.4 GET /dms/dictations/{AuthorName}/{DictationID}/attachment: download attachment of one dictation

3.4.1 Request specification

Purpose:

Download attachment of the dictation which

- has {DictationID}
- belongs to {AuthorName}
- has Transcription finished state
- has attached document

HTTP method:

GET

URL template:

/dms/dictations/{AuthorName}/{DictationID}/attachment

Request headers:

Header name	Value
x-sps-api-key	<API_key>

3.4.2 Parameters

Parameter name	Description
AuthorName	URL path parameter.

	The dictation's Author property must match with this name.
DictationID	URL path parameter. The dictation is searched by this ID.

3.4.3 Request body

No request body is needed.

3.4.4 Request URL example

/dms/dictations/user01/3D8A776F-86FF-4050-96EA-1C325DBF8834/attachment

3.4.5 Responses

3.4.5.1 Attachment download is successful

Description:

Attachment file of the dictation having {DictationID} and belonging to {AuthorName} is found and downloaded.

Code:

200

Response headers:

Header name	Value
Content-Type	application/octet-stream
Content-Disposition	see_dictation_attachment; filename=<AudioFilename>.<ext1>.<ext2>
Content-Length	<Attachment_filesize_in_bytes>

Response body / value:

The response body contains the file content of the attachment in binary format.

3.4.5.2 Error responses

Code	Description	Response body / value
400	AuthorName is empty, missing or otherwise invalid in request URL	{ "CRI": "<GUID>", "FailureCode": "Username_not_valid" }
400	DictationID is empty, missing or otherwise invalid in request URL	{ "CRI": "<GUID>", "FailureCode": "ID_not_valid" }
400	User xml with author role is not available or not found	{ "CRI": "<GUID>", "FailureCode": "User_not_exists" }
401	Unauthorized	{ "CRI": "<GUID>", "FailureCode": "Invalid_api_key" }
404	Dictation information cannot be found for the specified DictationID	{ "CRI": "<GUID>", "FailureCode": "Dictation_id_not_found" }

		}
404	Dictation information is available, but the dictation has no attachment	{ "CRI": "<GUID>", "FailureCode": "Attachment_not_found" }
409	Dictation with specified DictationID has wrong state	{ "CRI": "<GUID>", "FailureCode": "Dictation_state_not_valid_for_operation" }
500	Any other error encountered during request processing	{ "CRI": "<GUID>", "FailureCode": "Generic_error" }

4 Transcription Service API

Implementation planned in a future iteration.

5 Diagnostics endpoints

5.1 Request authentication

By default, incoming requests must be authenticated with Windows Authentication. For more information please refer to the Administrator's guide.

5.2 GET /test/testconfig

5.2.1 Request specification

Purpose:

Perform a validation test on the service configuration settings.

HTTP method:

GET

URL template:

/test/testconfig

Request headers:

No special headers.

5.2.2 Parameters

No special request parameters.

5.2.3 Request body

No request body is needed.

5.2.4 Request URL example

/test/testconfig

5.2.5 Responses

5.2.5.1 *All test steps are successful*

Description:

All validation tests passed.

Code:

200

Response body / value:

The response body is a list of items, each item has the following JSON structure:

```
{
  "StepID": "<step_id_name>",
  "StepResult": "SUCCESS|FAIL",
  "StepDescription": "<step_description_text>"
}
```

List of StepID items:

- 901_ServerLocalTime
- 001_SEERootFoundInConfig
- 005_SEERootExists
- 010_SEERootStructureCorrect
- 020_SEELicenseServerAddressFoundInConfig
- 025_SEELicenseServerPortFoundInConfig
- 030_SEELicenseServerCanConnect
- 035_TempRootFolderAccessible
- 101_AccessTokenLifetimeFoundInConfig

5.2.5.2 Error responses

Code	Description	Response body / value
566	The self-test of the service resulted in an error, the returned content must contain at least one "FAIL" item.	List of StepID items described above.

When a step fails, not all StepID items are returned, because the testing process is stopped at the failed step which will be the final one in the list.