Sapling

Release 0.1

Sapling Intelligence

Jun 14, 2023

CONTENTS

1 Contents **3**
1.1 Usage 3
1.2 API Reference 4

Index **11**

Sapling is a Python client to interface with Sapling.ai HTTP APIs.

Refer to the the *Usage* section for further information, including *Installation* instructions.

Here's a grammar check demo.

Note: This project is under active development. Please email support@sapling.ai with any questions or feedback.

CHAPTER

ONE

CONTENTS

1.1 Usage

1.1.1 Installation

Install the `sapling-py` package with pip. Here's the package link.

```
python -m pip install sapling-py
```

1.1.2 Quickstart

- Register for an account at Sapling.ai.
- After registering and signing in, generate a development API key in your dashboard.
- Install the Python client by following the installation steps above.

```
from sapling import SaplingClient

API_KEY = '<YOUR_API_KEY>'
client = SaplingClient(api_key=API_KEY)
edits = client.edits('Lets get started!', session_id='test_session')
```

- The result should be an array of edits of this form:

```
[{
  "id": "aa5ee291-a073-5146-8ebc-c9c899d01278",
  "sentence": "Lets get started!",
  "sentence_start": 0,
  "start": 0
  "end": 4,
  "replacement": "Let's",
  "error_type": "R:OTHER",
  "general_error_type": "Other",
}]
```

- More information on request options and response structure.
- Get a production key by following this documentation.

1.2 API Reference

Sapling HTTP API documentation

class sapling.client.**SaplingClient**(*api_key*, *timeout=120*, *hostname=None*, *pathname=None*)

Sapling client class. Provides a mapping of Python functions to Sapling HTTP REST APIs.

Parameters

- **api_key** (*str*) – 32-character API key
- **timeout** (*int*) – Timeout for API call in seconds. Defaults to 120 seconds.
- **hostname** (*str*) – Hostname override for SDK and self-hosted deployments.
- **pathname** (*str*) – Pathname override for SDK and self-hosted deployments as well as version requirements.

accept_complete(*complete_uuid*, *query*, *completion*, *session_id=None*)

Use this API endpoint to have Sapling improve completions over time.

Each suggested autocomplete has a UUID. You can pass this information back to Sapling to indicate the suggestion was helpful.

Parameters

- **complete_uuid** (*str, uuid*) – Opaque UUID of the edit returned from the complete endpoint.
- **query** (*str*) – The query text passed to the complete endpoint.
- **completion** (*str*) – The suggested completion text returned from the complete endpoint.

accept_edit(*edit_uuid*, *session_id=None*)

Use this API endpoint to have Sapling adapt its system over time.

Each suggested edit has an edit UUID. You can pass this information back to Sapling to indicate the edit suggestion was helpful. For each unique edit in each document, use the accept or reject API endpoint only once in total.

Parameters

- **edit_uuid** (*str, uuid*) – Opaque UUID of the edit returned from the edits endpoint
- **session_id** (*str*) – Unique name or UUID of text that is being processed

aidetect(*text*, *sent_scores=None*)

Score a piece of text on how likely it was generated by AI.

Parameters

- **text** (*str*) – Text to
- **sent_scores** (*bool*) – If true, each sentence will also be scored individually.

Return type dict

Returns

- score: float between 0 and 1, probability that text is AI generated
- sentence_scores: If sent_scores is set, will return a list of scores per sentence.
- text: text that was processed

chunk_html(*html*, *max_length*, *step_size=None*)

Break an input text into blocks of length of most max_length. When splitting the text, the API follows the following preference stack:

page break > paragraph breaks > line breaks > tabs > punctuation > all other whitespace

Note: This endpoint not only breaks up the HTML but also discards all HTML tags, resulting in plain text.

Parameters

- **html** (*str*) – HTML to be chunked
- **max_length** (*integer*) – Maximum length of text segments.
- **step_size** (*integer*) – Size of window to look for split points.

Return type dict

Returns

- chunks: List of resulting chunks representing the segmented text contained within the HTML

chunk_text(*text*, *max_length*, *step_size=None*)

Break an input text into blocks of length of most max_length. When splitting the text, the API follows the following preference stack:

page break > paragraph breaks > line breaks > tabs > punctuation > all other whitespace

Parameters

- **text** (*str*) – Text to be chunked
- **max_length** (*integer*) – Maximum length of text segments.
- **step_size** (*integer*) – Size of window to look for split points.

Return type dict

Returns

- chunks: List of resulting chunks

complete(*query*, *session_id=None*)

Provides predictions of the next few characters or words

Parameters

- **query** (*str*) – Text to get completions against.
- **session_id** (*str*) – Unique name or UUID of document or portion of text that is being checked

edits(*text*, *session_id=None*, *lang=None*, *variety=None*, *medical=None*, *auto_apply=False*)

Fetches edits (including for grammar and spelling) for provided text.

Parameters

- **text** (*str*) – Text to process for edits.
- **session_id** (*str*) – Unique name or UUID of document or portion of text that is being checked
- **session_id** – 2 letter ISO 639-1 language code
- **variety** (*str*) – Specifies regional English variety preference. Defaults to the configuration in the user Sapling dashboard.

- **medical** (*bool*) – If true, the backend will apply Sapling's medical dictionary.
- **auto_apply** (*bool*) – Whether to return a field with edits applied to the text

Return type list[dict]

Returns

- sentence: Unedited sentence
- sentence_start: Offset of sentence from start of text
- start: Offset of edit start relative to sentence
- end: Offset of edit end relative to sentence
- replacement: Suggested replacement
- error_type: Error type
- general_error_type: General Error type

Supported languages:

- *de*: German (Deutsch)
- *el*: Greek ()
- *en*: English (US/UK/CA/AU)
- *es*: Spanish (Español)
- *fr*: French (Français) (*fr-fr* and *fr-ca* coming soon)
- *it*: Italian (Italiano)
- *jp*: Japanese ()
- *ko*: Korean ()
- *nl*: Dutch (Nederlands)
- *pl*: Polish (Polski)
- *pt*: Portuguese (Português) (*pt-pt* and *pt-br* coming soon)
- *sv*: Swedish (Svenska)
- *tl*: Tagalog
- *zh*: Chinese ()

Supported varieties:

- *us-variety*: American English
- *gb-variety*: British English
- *au-variety*: Australian English
- *ca-variety*: Canadian English
- *null-variety*: Don't suggest changes based on English variety

postprocess(*text*, *session_id*, *operations*)

Performs a variety of operations that are useful for working with the outputs of an NLP (whether human or AI) syste

- Fixing or restoring punctuation

- Fixing capitalization
- Fixing or restoring whitespace

Example use cases include repairing transcriptions or captions.

Parameters

- **text** (*str*) – Text to postprocess
- **session_id** – Unique name or UUID of document or portion of text that is being chunked
- **operations** (*list[str]*) – Operations to apply. The currently accepted operations are: - capitalize - punctuate - fixspaces

Return type list[dict]

Returns Same as the edits endpoint: - sentence: Unedited sentence - sentence_start: Offset of sentence from start of text - start: Offset of edit start relative to sentence - end: Offset of edit end relative to sentence - replacement: Suggested replacement - error_type: Error type - general_error_type: General Error type

reject_edit(*edit_uuid*, *session_id=None*)

Use this API endpoint to have Sapling not recommend the same edit anymore.

Each suggested edit has an edit UUID. You can pass this information back to Sapling to indicate the edit suggestion was not helpful. For each unique edit in each document, use the accept or reject API endpoint only once in total.

Parameters

- **edit_uuid** (*str, uuid*) – Opaque UUID of the edit returned from the edits endpoint
- **session_id** (*str*) – Unique name or UUID of text that is being processed

spellcheck(*text*, *session_id=None*, *min_length=None*, *multiple_edits=None*, *lang=None*, *auto_apply=False*, *variety=None*, *user_data=None*)

Fetches spelling (no grammar or phrase level) edits for provided text.

Parameters

- **text** (*str*) – Text to process for edits.
- **session_id** (*str*) – Unique name or UUID of document or portion of text that is being checked
- **min_length** (*int*) – Default is 3. Minimum character length of words to suggest corrections for. Setting this too low will result in much higher false positives.
- **multiple_edits** (*bool*) – Default is false. If true, will return *candidates* field containing list of other potential corrections for each error.
- **lang** (*str*) – Default is English. Specify a language to spellcheck the text against.
- **auto_apply** (*bool*) – Whether to return a field with edits applied to the text. Cannot be set with multiple_edits option.
- **variety** (*str*) – Specifies regional English variety preference. Defaults to the configuration in the user Sapling dashboard.

Return type list[dict]

Supported languages:

- *en*: English

- *ar*:
- *bg*:
- *ca*: català
- *cs*: čeština
- *da*: dansk
- *de*: Deutsch
- *el*:
- *es*: español
- *et*: eesti keel
- *fa*:
- *fi*: suomi
- *fr*: français (*fr-fr* and *fr-ca* coming soon)
- *he*:
- *hi*: ",
- *hr*: hrvatski,
- *hu*: magyar nyelv
- *id*: bahasa Indonesia
- *is*: íslenska
- *it*: italiano
- *jp/ja*:
- *ko*:
- *lt*: lietuvių kalba
- *lv*: latviešu valoda
- *nl*: Nederlands
- *no*: norsk
- *pl*: polski
- *pt*: português
- *ro*: limba română
- *ru*:
- *sk*: slovenčina
- *sq*: shqip
- *sr*: srpski
- *sv*: svenska
- *th*:
- *tl*: Tagalog /
- *tr*: Türkçe

- *uk*:
- *vi*: Ting Vit
- *zh*:

Supported varieties:

- *us-variety*: American English
- *gb-variety*: British English
- *au-variety*: Australian English
- *ca-variety*: Canadian English
- *null-variety*: Don't suggest changes based on English variety

INDEX

A

accept_complete() (*sapling.client.SaplingClient method*), 4
accept_edit() (*sapling.client.SaplingClient method*), 4
aidetect() (*sapling.client.SaplingClient method*), 4

C

chunk_html() (*sapling.client.SaplingClient method*), 4
chunk_text() (*sapling.client.SaplingClient method*), 5
complete() (*sapling.client.SaplingClient method*), 5

E

edits() (*sapling.client.SaplingClient method*), 5

P

postprocess() (*sapling.client.SaplingClient method*), 6

R

reject_edit() (*sapling.client.SaplingClient method*), 7

S

SaplingClient (*class in sapling.client*), 4
spellcheck() (*sapling.client.SaplingClient method*), 7