codaio

Mikhail Beliansky

Sep 01, 2019

CONTENTS:

Python wrapper for coda.io API

Project home: https://github.com/blasterai/codaio

class codaio.**Coda**(*api_key: str*, *href: str = 'https://coda.io/apis/v1beta1'*)

Raw API client. It is used in *codaio* objects like Document to access the raw API endpoints. Can also be used by itself to access Raw API.

account() → Dict

At this time, the API exposes some limited information about your account. However, /whoami is a good endpoint to hit to verify that you're hitting the API correctly and that your token is working as expected.

Docs: https://coda.io/developers/apis/v1beta1#tag/Account

create_doc(*title: str*, *source_doc: Optional[str] = None*, *tz: Optional[str] = None*) → Dict

Creates a new Coda doc, optionally copying an existing doc.

Docs: https://coda.io/developers/apis/v1beta1#operation/createDoc

Parameters

- **title** – Title of the new doc.
- **source_doc** – An optional doc ID from which to create a copy.
- **tz** – The timezone to use for the newly created doc.

Returns

delete(*endpoint: str*) → Dict

Make a DELETE request to the API endpoint.

Parameters **endpoint** – API endpoint to request

Returns

delete_doc(*doc_id: str*) → Dict

Deletes a doc.

Docs: https://coda.io/developers/apis/v1beta1#operation/deleteDoc

Parameters **doc_id** – ID of the doc. Example: "AbCDeFGH"

Returns

delete_row(*doc_id*, *table_id_or_name: str*, *row_id_or_name: str*) → Dict

Deletes the specified row from the table. This endpoint will always return a 202, so long as the row exists and is accessible (and the update is structurally valid). Row deletions are generally processed within several seconds. When deleting using a name as opposed to an ID, an arbitrary row will be removed.

Docs: https://coda.io/developers/apis/v1beta1#operation/deleteRow

Parameters

- **doc_id** – ID of the doc. Example: "AbCDeFGH"
- **table_id_or_name** – ID or name of the table. Names are discouraged because they're easily prone to being changed by users.

If you're using a name, be sure to URI-encode it. Example: "grid-pqRst-U"

Parameters **row_id_or_name** – ID or name of the row. Names are discouraged because they're easily prone to being changed by users.

If you're using a name, be sure to URI-encode it. If there are multiple rows with the same value in the identifying column, an arbitrary one will be selected.

classmethod from_environment() → codaio.coda.Coda
Initializes a Coda instance using API key store in environment variables under *CODA_API_KEY*

Returns

get(*endpoint: str*, *data: Dict = None*, *limit=None*, *offset=None*) → Dict
Make a GET request to API endpoint.

Parameters

- **endpoint** – API endpoint to request
- **data** – dictionary of optional query params
- **limit** – Maximum number of results to return in this query.
- **offset** – An opaque token used to fetch the next page of results.

Returns

get_column(*doc_id: str*, *table_id_or_name: str*, *column_id_or_name: str*) → Dict
Returns details about a column in a table.

Docs: https://coda.io/developers/apis/v1beta1#operation/getColumn

Parameters

- **doc_id** – ID of the doc. Example: "AbCDeFGH"
- **table_id_or_name** – ID or name of the table. Names are discouraged because they're easily prone to being changed by users. If you're using a name, be sure to URI-encode it. Example: "grid-pqRst-U"
- **column_id_or_name** – ID or name of the column. Names are discouraged because they're easily prone to being changed by users. If you're using a name, be sure to URI-encode it. Example: "c-tuVwxYz"

Returns

get_control(*doc_id: str*, *control_id_or_name: str*) → Dict
Returns info on a control.

Docs: https://coda.io/developers/apis/v1beta1#operation/getControl

Parameters

- **doc_id** – ID of the doc. Example: "AbCDeFGH"
- **control_id_or_name** – ID or name of the control. Names are discouraged because they're easily prone to being changed by users.

If you're using a name, be sure to URI-encode it. Example: "ctrl-cDefGhij"

get_doc(*doc_id: str*) → Dict
Returns metadata for the specified doc.

Docs: https://coda.io/developers/apis/v1beta1#operation/getDoc

Parameters **doc_id** – ID of the doc. Example: "AbCDeFGH"

Returns

get_folder(*doc_id: str*, *folder_id_or_name: str*) → Dict
Returns details about a folder.

Docs: https://coda.io/developers/apis/v1beta1#operation/getFolder

Parameters

- **doc_id** – ID of the doc. Example: "AbCDeFGH"
- **folder_id_or_name** – ID or name of the folder. Names are discouraged because they're easily prone to being changed by users. If you're using a name, be sure to URI-encode it. Example: "section-IjkLmnO"

Returns

get_formula(*doc_id: str*, *formula_id_or_name: str*) → Dict
Returns info on a formula.

Docs: https://coda.io/developers/apis/v1beta1#operation/getFormula

Parameters

- **doc_id** – ID of the doc. Example: "AbCDeFGH"
- **formula_id_or_name** – ID or name of the formula. Names are discouraged because they're easily prone to being changed by users.

If you're using a name, be sure to URI-encode it. Example: "f-fgHijkLm"

get_row(*doc_id: str*, *table_id_or_name: str*, *row_id_or_name: str*) → Dict
Returns details about a row in a table.

Docs: https://coda.io/developers/apis/v1beta1#operation/getRow

Parameters

- **doc_id** – ID of the doc. Example: "AbCDeFGH"
- **table_id_or_name** – ID or name of the table. Names are discouraged because they're easily prone to being changed by users. If you're using a name, be sure to URI-encode it. Example: "grid-pqRst-U"
- **row_id_or_name** – ID or name of the row. Names are discouraged because they're easily prone to being changed by users. If you're using a name, be sure to URI-encode it. If there are multiple rows with the same value in the identifying column, an arbitrary one will be selected.

get_section(*doc_id: str*, *section_id_or_name: str*) → Dict
Returns details about a section.

Docs: https://coda.io/developers/apis/v1beta1#operation/getSection

Parameters

- **doc_id** – ID of the doc. Example: "AbCDeFGH"
- **section_id_or_name** – ID or name of the section. Names are discouraged because they're easily prone to being changed by users. If you're using a name, be sure to URI-encode it. Example: "canvas-IjkLmnO"

Returns

get_table(*doc_id: str*, *table_id_or_name: str*) → Dict
Returns details about a specific table.

Docs: https://coda.io/developers/apis/v1beta1#operation/getTable

Parameters

- **doc_id** – ID of the doc. Example: "AbCDeFGH"

- **table_id_or_name** – ID or name of the table. Names are discouraged because they're easily prone to being changed by users. If you're using a name, be sure to URI-encode it. Example: "grid-pqRst-U"

Returns

get_view(*doc_id: str*, *view_id_or_name: str*) → Dict

Returns details about a specific view.

Docs: https://coda.io/developers/apis/v1beta1#operation/getView

Parameters

- **doc_id** – ID of the doc. Example: "AbCDeFGH"
- **view_id_or_name** – ID or name of the view. Names are discouraged because they're easily prone to being changed by users. If you're using a name, be sure to URI-encode it. Example: "table-pqRst-U"

Returns

list_columns(*doc_id: str*, *table_id_or_name: str*, *offset: Optional[int] = None*, *limit: Optional[int] = None*) → Dict

Returns a list of columns in a table.

Parameters

- **doc_id** – ID of the doc. Example: "AbCDeFGH"
- **table_id_or_name** – ID or name of the table. Names are discouraged because they're easily prone to being changed by users. If you're using a name, be sure to URI-encode it. Example: "grid-pqRst-U"
- **limit** – Maximum number of results to return in this query.
- **offset** – An opaque token used to fetch the next page of results.

Returns

list_controls(*doc_id: str*, *offset: Optional[int] = None*, *limit: Optional[int] = None*) → Dict

Controls provide a user-friendly way to input a value that can affect other parts of the doc. This API lets you list controls and get their current values.

Docs: https://coda.io/developers/apis/v1beta1#tag/Controls

Parameters

- **doc_id** – ID of the doc. Example: "AbCDeFGH"
- **limit** – Maximum number of results to return in this query.
- **offset** – An opaque token used to fetch the next page of results.

Returns

list_docs(*is_owner: bool = False*, *query: Optional[str] = None*, *source_doc_id: Optional[str] = None*, *limit: Optional[int] = None*, *offset: Optional[int] = None*) → Dict

Returns a list of Coda docs accessible by the user. These are returned in the same order as on the docs page: reverse chronological by the latest event relevant to the user (last viewed, edited, or shared).

Docs: https://coda.io/developers/apis/v1beta1#operation/listDocs

Parameters

- **is_owner** – Show only docs owned by the user.
- **query** – Search term used to filter down results.

- **source_doc_id** – Show only docs copied from the specified doc ID.
- **limit** – Maximum number of results to return in this query.
- **offset** – An opaque token used to fetch the next page of results.

Returns

list_folders(*doc_id: str*, *offset: Optional[int] = None*, *limit: Optional[int] = None*) → Dict
Returns a list of folders in a Coda doc.

Docs: https://coda.io/developers/apis/v1beta1#operation/listFolders

Parameters

- **doc_id** – ID of the doc. Example: "AbCDeFGH"
- **limit** – Maximum number of results to return in this query.
- **offset** – An opaque token used to fetch the next page of results.

Returns

list_formulas(*doc_id: str*, *offset: Optional[int] = None*, *limit: Optional[int] = None*) → Dict
Returns a list of named formulas in a Coda doc.

Docs: https://coda.io/developers/apis/v1beta1#operation/listFormulas

Parameters

- **doc_id** – ID of the doc. Example: "AbCDeFGH"
- **limit** – Maximum number of results to return in this query.
- **offset** – An opaque token used to fetch the next page of results.

list_rows(*doc_id: str*, *table_id_or_name: str*, *query: Optional[str] = None*, *use_column_names: bool = False*, *limit: Optional[int] = None*, *offset: Optional[int] = None*) → Dict
Returns a list of rows in a table.

Docs: https://coda.io/developers/apis/v1beta1#tag/Rows

Parameters

- **doc_id** – ID of the doc. Example: "AbCDeFGH"
- **table_id_or_name** – ID or name of the table. Names are discouraged because they're easily prone to being changed by users. If you're using a name, be sure to URI-encode it. Example: "grid-pqRst-U"
- **query** – Query used to filter returned rows, specified as <*column_id_or_name*>:<*value*>. If you'd like to use a column name instead of an ID, you must quote it (e.g., *"My Column":123*). Also note that *value* is a JSON value; if you'd like to use a string, you must surround it in quotes (e.g., *"groceries"*).
- **use_column_names** – Use column names instead of column IDs in the returned output. This is generally discouraged as it is fragile. If columns are renamed, code using original names may throw errors.
- **limit** – Maximum number of results to return in this query.
- **offset** – An opaque token used to fetch the next page of results.

list_sections(*doc_id: str*, *offset: Optional[int] = None*, *limit: Optional[int] = None*) → Dict
Returns a list of sections in a Coda doc.

Docs: https://coda.io/developers/apis/v1beta1#operation/listSections

Parameters

- **doc_id** – ID of the doc. Example: "AbCDeFGH"
- **limit** – Maximum number of results to return in this query.
- **offset** – An opaque token used to fetch the next page of results.

Returns

list_tables(*doc_id: str*, *offset: Optional[int] = None*, *limit: Optional[int] = None*) → Dict
Returns a list of tables in a Coda doc.

Docs: https://coda.io/developers/apis/v1beta1#operation/listTables

Parameters

- **doc_id** – ID of the doc. Example: "AbCDeFGH"
- **limit** – Maximum number of results to return in this query.
- **offset** – An opaque token used to fetch the next page of results.

Returns

list_views(*doc_id: str*, *offset: Optional[int] = None*, *limit: Optional[int] = None*) → Dict
Returns a list of views in a Coda doc.

Docs: https://coda.io/developers/apis/v1beta1#operation/listViews

Parameters

- **doc_id** – ID of the doc. Example: "AbCDeFGH"
- **limit** – Maximum number of results to return in this query.
- **offset** – An opaque token used to fetch the next page of results.

Returns

post(*endpoint: str*, *data: Dict*) → Dict
Make a POST request to the API endpoint.

Parameters

- **endpoint** – API endpoint to request
- **data** – data dict to be sent as body json

Returns

put(*endpoint: str*, *data: Dict*) → Dict
Make a PUT request to the API endpoint.

Parameters

- **endpoint** – API endpoint to request
- **data** – data dict to be sent as body json

Returns

resolve_browser_link(*url: str*, *degrade_gracefully: bool = False*) → Dict
Given a browser link to a Coda object, attempts to find it and return metadata that can be used to get more info on it. Returns a 400 if the URL does not appear to be a Coda URL or a 404 if the resource cannot be located with the current credentials.

Docs: https://coda.io/developers/apis/v1beta1#operation/resolveBrowserLink

Parameters

- **url** – The browser link to try to resolve. Example: "https://coda.io/d/_dAbCDeFGH/Launch-Status_sumnO"
- **degrade_gracefully** – By default, attempting to resolve the Coda URL of a deleted object will result in an error. If this flag is set, the next-available object, all the way up to the doc itself, will be resolved.

update_row(*doc_id: str*, *table_id_or_name: str*, *row_id_or_name: str*, *data: Dict*) → Dict

Updates the specified row in the table. This endpoint will always return a 202, so long as the row exists and is accessible (and the update is structurally valid). Row updates are generally processed within several seconds. When updating using a name as opposed to an ID, an arbitrary row will be affected.

Docs: https://coda.io/developers/apis/v1beta1#operation/updateRow

Parameters

- **doc_id** – ID of the doc. Example: "AbCDeFGH"
- **table_id_or_name** – ID or name of the table. Names are discouraged because they're easily prone to being changed by users.

If you're using a name, be sure to URI-encode it. Example: "grid-pqRst-U"

Parameters **row_id_or_name** – ID or name of the row. Names are discouraged because they're easily prone to being changed by users.

If you're using a name, be sure to URI-encode it. If there are multiple rows with the same value in the identifying column, an arbitrary one will be selected.

Parameters **data** – Example: {"row": {"cells": [{"column": "c-tuVwxYz", "value": "$12.34"}]}}

upsert_row(*doc_id: str*, *table_id_or_name: str*, *data: Dict*) → Dict

Inserts rows into a table, optionally updating existing rows if any upsert key columns are provided. This endpoint will always return a 202, so long as the doc and table exist and are accessible (and the update is structurally valid). Row inserts/upserts are generally processed within several seconds. When upserting, if multiple rows match the specified key column(s), they will all be updated with the specified value.

Docs: https://coda.io/developers/apis/v1beta1#operation/upsertRows

Parameters

- **doc_id** – ID of the doc. Example: "AbCDeFGH"
- **table_id_or_name** – ID or name of the table. Names are discouraged because they're easily prone to being changed by users.

If you're using a name, be sure to URI-encode it. Example: "grid-pqRst-U"

Parameters **data** – {"rows": [{"cells": [{"column": "c-tuVwxYz", "value": "$12.34"}]}], "keyColumns": ["c-bCdeFgh"]}

class codaio.**Document**(*id: str*, *coda: Coda*)

Main class for interacting with coda.io API using *codaio* objects.

classmethod from_environment(*doc_id: str*)

Initializes a Document instance using API key stored in environment variables under *CODA_API_KEY*

Parameters **doc_id** – ID of the doc. Example: "AbCDeFGH"

Returns

get_table(*table_id_or_name: str*) → codaio.coda.Table
Gets a Table object from table name or ID.

Parameters **table_id_or_name** – ID or name of the table. Names are discouraged because they're easily prone to being changed by users. If you're using a name, be sure to URI-encode it. Example: "grid-pqRst-U"

Returns

list_sections(*offset: Optional[int] = None*, *limit: Optional[int] = None*) → List[codaio.coda.Section]
Returns a list of *Section* objects for each section in the document.

Parameters

- **limit** – Maximum number of results to return in this query.
- **offset** – An opaque token used to fetch the next page of results.

Returns

list_tables(*offset: Optional[int] = None*, *limit: Optional[int] = None*) → List[codaio.coda.Table]
Returns a list of *Table* objects for each table in the document.

Parameters

- **limit** – Maximum number of results to return in this query.
- **offset** – An opaque token used to fetch the next page of results.

Returns

class codaio.**Table**(*id: str*, *type: str*, *href: str*, *name: str*, *document: Document*, *display_column: Dict = None*, *browser_link: str = None*, *row_count: int = None*, *sorts: List = []*, *layout: str = None*, *created_at=None*, *updated_at=None*, *columns_storage: List[Column] = []*)

columns(*offset: Optional[int] = None*, *limit: Optional[int] = None*) → List[codaio.coda.Column]
Returns a list of Table columns. Columns are stored in self.columns_storage for faster access as they tend to change less frequently than rows.

Parameters

- **limit** – Maximum number of results to return in this query.
- **offset** – An opaque token used to fetch the next page of results.

Returns

delete_row(*row: codaio.coda.Row*) → Dict
Delete row.

Parameters **row** – a *Row* object to delete.

delete_row_by_id(*row_id: str*)
Deletes row by id.

Parameters **row_id** – ID of the row to delete.

find_row_by_column_id_and_value(*column_id*, *value*) → List[codaio.coda.Row]
Fins rows by a value in column specified by id.

Parameters

- **column_id** – ID of the column.
- **value** – Search value.

Returns

find_row_by_column_name_and_value(*column_name: str*, *value: Any*) → List[codaio.coda.Row]
Finds rows by a value in column specified by name (discouraged).

Parameters

- **column_name** – Name of the column.
- **value** – Search value.

Returns

get_column_by_id(*column_id*) → codaio.coda.Column
Gets a Column by id.

Parameters **column_id** – ID of the column. Example: "c-tuVwxYz"

Returns

get_column_by_name(*column_name*) → codaio.coda.Column
Gets a Column by id.

Parameters **column_name** – Name of the column. Discouraged in case using column_id is possible. Example: "Column 1"

Returns

rows(*offset: Optional[int] = None*, *limit: Optional[int] = None*) → List[codaio.coda.Row]
Returns list of Table rows.

Parameters

- **limit** – Maximum number of results to return in this query.
- **offset** – An opaque token used to fetch the next page of results.

Returns

update_row(*row: Union[str, codaio.coda.Row], cells: List[codaio.coda.Cell]*) → Dict
Updates row with values according to list in cells.

Parameters

- **row** – a str ROW_ID or an instance of class Row
- **cells** –

upsert_row(*cells: List[codaio.coda.Cell]*) → Dict
Upserts a row using *Cell* objects in list.

Parameters **cells** – list of *Cell* objects.

upsert_rows(*list_cells: List[List[codaio.coda.Cell]]*) → Dict
Works similar to Table.upsert_row() but uses 1 POST request for multiple rows. Input is a list of lists of Cells.

Parameters **list_cells** – list of lists of *Cell* objects, one list for each row.

class codaio.**Column**(*id: str*, *type: str*, *href: str*, *document: Document*, *name: str*, *table: Table*, *display: bool = None*, *calculated: bool = False*)

class codaio.**Row**(*id: str*, *type: str*, *href: str*, *document: Document*, *name: str*, *created_at*, *index: int*, *updated_at*, *values*, *table: Table*, *browser_link: str = None*)

delete()
Delete row.

Returns

class codaio.**Cell**(*column: Column*, *value_storage: Any*, *row: Row = None*)

CHAPTER

ONE

INDICES AND TABLES

- genindex
- search

INDEX