SuperAnnotate Release July 24, 2022

Python SDK version `4.4.1`

SuperAnnotate

Setup

In [1]:

```
# SETUP #
#########

# importing SuperAnnotate Python SDK and Pretty Print
import superannotate
import pprint as pretty

# instantiating a SuperAnnotate team-level client object
sa = superannotate.SAClient(config_path="~/.superannotate/dev_config.json")

# getting and printing the name of the team to make sure the connection is establish
# printing the SDK version
print(sa.get_team_metadata()["name"])
print(superannotate.__version__)

# Constants
PROJECT = "Custom Metadata"
```

```
SA-PYTHON-SDK - INFO - Development version 4.4.1dev10 of SuperAnnotate
SDK is being used.
SA-PYTHON-SDK - WARNING - There is a newer version of SuperAnnotate Py
thon SDK available on PyPI. Run 'pip install --upgrade superannotate'
to upgrade from your version 4.4.1.dev10 to 4.4.1b4
Dev Branch
4.4.1dev10
```

Custom Metadata. Part 1: Custom Fields (Scope: Project)

Creating/Adding Custom Fields

`SAClient.create_custom_fields(project,fields)`

Create custom fields for items in a project in addition to built-in metadata. Using this function again with a different schema won't override the existing fields, but will add new ones. It is possible to have up to 25 custom fields per project. Use the `upload_custom_values()` function to fill them with values for each item.

Parameters:

- **project** *(str)* – project name (e.g., “project1”)
- **fields** *(dict)* – dictionary describing the fields and their specifications added to the project.

Returns: custom fields actual schema of the project

Return type: dict

Supported Field Types

number

Number type fields may have the following specifications with their corresponding values:

field spec	spec value
minimum	any number (int or float)
maximum	any number (int or float)
enum	list of numbers (int or float)

string

String type fields may have the following specifications with their corresponding values:

field spec	spec value
format	“email” (user@example.com (mailto:user@example.com)) or “date” (YYYY-MM-DD)
enum	list of strings

In [7]:

```
# to create or add custom fields for a Project, first, define the schema-dict.
# each key of the dict represents the name of the field and must have a type, plus c

custom_fields = {
    "study_date": {
        "type": "string",
        "format": "date"
    },
    "patient_id": {
        "type": "string"
    },
    "patient_sex": {
        "type": "string",
        "enum": ["male", "female"]
    },
    "medical_specialist": {
        "type": "string",
        "format": "email"
    }
}

sa.create_custom_fields(
    project = PROJECT,
    fields = custom_fields
)
```

Out[7]:

```
{'study_date': {'type': 'string', 'format': 'date'},
 'patient_id': {'type': 'string'},
 'patient_sex': {'enum': ['male', 'female'], 'type': 'string'},
 'medical_specialist': {'type': 'string', 'format': 'email'}}
```

In [8]:

```
# to add new fields to the existing ones

new_custom_field = {
    "patient_age": {
        "type": "number",
        "minimum": 18
    }
}

sa.create_custom_fields(
    project=PROJECT,
    fields=new_custom_field
)
```

Out[8]:

```
{'study_date': {'type': 'string', 'format': 'date'},
 'patient_id': {'type': 'string'},
 'patient_sex': {'enum': ['male', 'female'], 'type': 'string'},
 'medical_specialist': {'type': 'string', 'format': 'email'},
 'patient_age': {'type': 'number', 'minimum': 18}}
```

Getting Existing Custom Fields

`SAClient.get_custom_fields(project,fields)`

Get the schema of the custom fields defined for the project.

Parameters:

- **project** *(str)* – project name (e.g., “project1”)

Returns: custom fields actual schema of the project
Return type: dict

In [9]:

```
sa.get_custom_fields(PROJECT)
```

Out[9]:

```
{'study_date': {'type': 'string', 'format': 'date'},
 'patient_id': {'type': 'string'},
 'patient_sex': {'enum': ['male', 'female'], 'type': 'string'},
 'medical_specialist': {'type': 'string', 'format': 'email'},
 'patient_age': {'type': 'number', 'minimum': 18}}
```

Deleting Custom Fields

`SAClient.delete_custom_fields(project,fields)`

Remove custom fields from a project’s custom metadata schema.

Parameters:

- **project** *(str)* – project name (e.g., “project1”)
- **fields** *(list of strs)* – list of field names to remove

Returns: custom fields actual schema of the project
Return type: dict

In [5]:

```
# to delete selected custom fields

sa.delete_custom_fields(
    project=PROJECT,
    fields=["patient_sex", "patient_age"]
)
```

```
SA-PYTHON-SDK - INFO - Matched fields deleted from schema.
```

Out[5]:

```
{'study_date': {'type': 'string', 'format': 'date'},
 'patient_id': {'type': 'string'},
 'medical_specialist': {'type': 'string', 'format': 'email'}}
```

In [6]:

```
# to delete all existing custom fields

existing_field_names = list(sa.get_custom_fields(PROJECT).keys())
sa.delete_custom_fields(
    project=PROJECT,
    fields=existing_field_names
)
```

```
SA-PYTHON-SDK - INFO - Matched fields deleted from schema.
```

Out[6]:

```
{}
```

Custom Metadata. Part 2: Custom Values (Scope: Item)

Uploading Custom Field-Custom Value Pairs to Items

`SAClient.upload_custom_values(project, items)`

Attach custom metadata to items.
`SAClient.get_item_metadata()` , `SAClient.search_items()` , `SAClient.query()` methods will return the item metadata and custom metadata.

Parameters:

- **project** *(str)* – project name or folder path (e.g., “project1/folder1”)
- **items** *(list of dicts)* – list of name-data pairs. The key of each dict indicates an existing item name and the value represents the custom metadata dict. The values for the corresponding keys will be added to an item or will be overridden.

Returns: dictionary with succeeded and failed item names.
Return type: dict

In [10]:

```
# to attach custom values to items, first map each name of an existing item to the c
# prerequisites:
#   - the item should exist in the platform
#   - a custom field should exist in the schema of custom fields, defined at the lev
#   - a custom value should be valid against the type and the specs defined for the

items_values = [
    {
        "brain_study_1.jpeg": {                                  #
            "study_date": "2021-12-31",                          #
            "patient_id": "62078f8a756ddb2ca9fc9660",            #  Valid
            "patient_sex": "female",                             #  Item
            "medical_specialist": "robertboxer@ms.com",          #
        }                                                        #
    },
    {
        "brain_study_2.jpeg": {                                  #
            "study_date": "2021-12-31",                          #
            "patient_id": "62078f8a756ddb2ca9fc9660",            # Valid
            "patient_sex": "female"                              # Item
                                                                 #
        }                                                        #
    },
    {
        "brain_study_3.jpeg": {                                  #
            "study_date": "2021-12-31",                          #
            "patient_": "62078f8a756ddb2ca9fc9662",              # Invalid
            "patient_sex": "female",                             # Item
            "medical_specialist": "robertboxer",                 #
        }                                                        #
    }
]

upload_results = sa.upload_custom_values(
    project=PROJECT,
    items=items_values
)

print(upload_results)
```

```
SA-PYTHON-SDK - INFO - Validating metadata against the schema of the c
ustom fields. Valid metadata will be attached to the specified item.
SA-PYTHON-SDK - ERROR - The metadata dicts of 1 items are invalid beca
use they don't match the schema of the custom fields defined for the
"Custom Metadata" project.
{'succeeded': ['brain_study_1.jpeg', 'brain_study_2.jpeg'], 'failed':
['brain_study_3.jpeg']}
```

In [11]:

```
for item in upload_results['succeeded']:
    item_meta = sa.get_item_metadata(
        project=PROJECT,
        item_name=item,
        include_custom_metadata=True
    )
    pretty.pprint(item_meta)
```

```
{'annotation_status': 'NotStarted',
 'annotator_email': None,
 'approval_status': None,
 'createdAt': '2022-07-21T13:00:15.000Z',
 'custom_metadata': {'medical_specialist': 'robertboxer@ms.com',
                     'patient_id': '62078f8a756ddb2ca9fc9660',
                     'patient_sex': 'female',
                     'study_date': '2021-12-31'},
 'entropy_value': None,
 'is_pinned': False,
 'name': 'brain_study_1.jpeg',
 'path': 'Custom Metadata',
 'prediction_status': 'NotStarted',
 'qa_email': None,
 'segmentation_status': None,
 'updatedAt': '2022-07-21T13:00:15.000Z',
 'url': None}
{'annotation_status': 'NotStarted',
 'annotator_email': None,
 'approval_status': None,
 'createdAt': '2022-07-21T13:00:14.000Z',
 'custom_metadata': {'patient_id': '62078f8a756ddb2ca9fc9660',
                     'patient_sex': 'female',
                     'study_date': '2021-12-31'},
 'entropy_value': None,
 'is_pinned': False,
 'name': 'brain_study_2.jpeg',
 'path': 'Custom Metadata',
 'prediction_status': 'NotStarted',
 'qa_email': None,
 'segmentation_status': None,
 'updatedAt': '2022-07-21T13:00:14.000Z',
 'url': None}
```

In [12]:

```python
# case: adding a missing field-value pair to an item

pretty.pprint(
    sa.upload_custom_values(
        project=PROJECT,
        items=[
            {
                "brain_study_2.jpeg": {
                    "medical_specialist": "robertboxer@ms.com"
                }
            }
        ]
    )
)

pretty.pprint(
    sa.get_item_metadata(
        project=PROJECT,
        item_name="brain_study_2.jpeg",
        include_custom_metadata=True
    )
)
```

```
SA-PYTHON-SDK - INFO - Validating metadata against the schema of the c
ustom fields. Valid metadata will be attached to the specified item.
{'failed': [], 'succeeded': ['brain_study_2.jpeg']}
{'annotation_status': 'NotStarted',
 'annotator_email': None,
 'approval_status': None,
 'createdAt': '2022-07-21T13:00:14.000Z',
 'custom_metadata': {'medical_specialist': 'robertboxer@ms.com',
                     'patient_id': '62078f8a756ddb2ca9fc9660',
                     'patient_sex': 'female',
                     'study_date': '2021-12-31'},
 'entropy_value': None,
 'is_pinned': False,
 'name': 'brain_study_2.jpeg',
 'path': 'Custom Metadata',
 'prediction_status': 'NotStarted',
 'qa_email': None,
 'segmentation_status': None,
 'updatedAt': '2022-07-21T13:00:14.000Z',
 'url': None}
```

In [13]:

```
# case: overriding existing values

pretty.pprint(
    sa.upload_custom_values(
        project=PROJECT,
        items=[
            {
                "brain_study_2.jpeg": {
                    "medical_specialist": "nina.tucker@ms.com",    # prev. "robert.b
                    "patient_id": "62078f8a756ddb2ca9fc9661"       # prev. "62078f8a
                }
            }
        ]
    )
)

pretty.pprint(
    sa.get_item_metadata(
        project=PROJECT,
        item_name="brain_study_2.jpeg",
        include_custom_metadata=True
    )
)
```

```
SA-PYTHON-SDK - INFO - Validating metadata against the schema of the c
ustom fields. Valid metadata will be attached to the specified item.
{'failed': [], 'succeeded': ['brain_study_2.jpeg']}
{'annotation_status': 'NotStarted',
 'annotator_email': None,
 'approval_status': None,
 'createdAt': '2022-07-21T13:00:14.000Z',
 'custom_metadata': {'medical_specialist': 'nina.tucker@ms.com',
                     'patient_id': '62078f8a756ddb2ca9fc9661',
                     'patient_sex': 'female',
                     'study_date': '2021-12-31'},
 'entropy_value': None,
 'is_pinned': False,
 'name': 'brain_study_2.jpeg',
 'path': 'Custom Metadata',
 'prediction_status': 'NotStarted',
 'qa_email': None,
 'segmentation_status': None,
 'updatedAt': '2022-07-21T13:00:14.000Z',
 'url': None}
```

Deleting Custom Field-Custom Value Pairs from Items

`SAClient.delete_custom_values(project, items)`

Remove custom data from items

Parameters:

- **project** *(str)* – project name or folder path (e.g., “project1/folder1”)

- **items** *(list of dicts)* – - list of name-fields pairs. The key of each dict element indicates an existing item in the project root or folder. The value should be the list of fields to be removed from the given item. Please note, that the function removes pointed metadata from a given item. To delete metadata for all items you should delete it from the custom metadata schema. To override values for existing fields, use `SAClient.upload_custom_values()`

Returns:
Return type:

In [14]:

```
# to remove selected custom field-value pairs from items

sa.delete_custom_values(
    project=PROJECT,
    items=[
        {"brain_study_1.jpeg": ["medical_specialist", "patient_sex"]},
        {"brain_study_2.jpeg": ["medical_specialist", "patient_sex"]}
    ]
)

for item in sa.search_items(PROJECT, name_contains="study"):
    item_meta = sa.get_item_metadata(
        project=PROJECT,
        item_name=item['name'],
        include_custom_metadata=True
    )
    pretty.pprint(item_meta)
```

```
SA-PYTHON-SDK - INFO - Corresponding fields and their values removed f
rom items.
{'annotation_status': 'NotStarted',
 'annotator_email': None,
 'approval_status': None,
 'createdAt': '2022-07-21T13:00:15.000Z',
 'custom_metadata': {'patient_id': '62078f8a756ddb2ca9fc9660',
                     'study_date': '2021-12-31'},
 'entropy_value': None,
 'is_pinned': False,
 'name': 'brain_study_1.jpeg',
 'path': 'Custom Metadata',
 'prediction_status': 'NotStarted',
 'qa_email': None,
 'segmentation_status': None,
 'updatedAt': '2022-07-21T13:00:15.000Z',
 'url': None}
{'annotation_status': 'NotStarted',
 'annotator_email': None,
 'approval_status': None,
 'createdAt': '2022-07-21T13:00:14.000Z',
 'custom_metadata': {'patient_id': '62078f8a756ddb2ca9fc9661',
                     'study_date': '2021-12-31'},
 'entropy_value': None,
 'is_pinned': False,
 'name': 'brain_study_2.jpeg',
 'path': 'Custom Metadata',
 'prediction_status': 'NotStarted',
 'qa_email': None,
 'segmentation_status': None,
 'updatedAt': '2022-07-21T13:00:14.000Z',
 'url': None}
{'annotation_status': 'NotStarted',
 'annotator_email': None,
 'approval_status': None,
 'createdAt': '2022-07-21T13:00:14.000Z',
 'custom_metadata': {},
 'entropy_value': None,
 'is_pinned': False,
 'name': 'brain_study_3.jpeg',
 'path': 'Custom Metadata',
 'prediction_status': 'NotStarted',
 'qa_email': None,
 'segmentation_status': None,
 'updatedAt': '2022-07-21T13:00:14.000Z',
 'url': None}
```