
TROMPA CE client

MTG-UPF

Jun 15, 2021

CONTENTS:

1	Installation	3
1.1	Usage	3
1.2	Demos	4
1.3	Mutations	7
1.4	Queries	38
1.5	Algorithms	40
2	Indices and tables	41
	Python Module Index	43
	Index	45

The TROMPA CE client is a Python library that can be used to read data from and write data to the TROMPA Contributor Environment using its GraphQL interface.

INSTALLATION

To install the package, use pip:

```
pip install trompace-client
```

1.1 Usage

1.1.1 Configuration

In order to connect to a CE server you require a configuration file. See the `trompace.ini` example file in the code repository

```
[server]
host = http://localhost:4000

[auth]
id = local
key = PZsG+oEW3K3QOoB5z0f30InzjXdBqM9LMtJa7BTglxo=
scopes = *
# If the server doesn't require auth, set required to no, otherwise it should be yes
required = yes

[logging]
# A python logging level (debug, info, warning, error)
level = debug
```

Set `server.host` to the URL that your CE is running from. For information about the `auth.id`, `auth.key`, and `auth.scopes` values, see the [CE documentation on authentication](#)

In order to perform queries you need to load the config file. To do so, use `trompace.config.config.load()` with the name of the config file:

```
from trompace.config import config
config.load('trompace.ini')
```

Alternatively, you can set the environment variable `TROMPACE_CLIENT_CONFIG` to the path of the config file and call `trompace.config.config.load()` with no arguments:

```
from trompace.config import config
config.load()
```

1.1.2 Making requests

`trompace-client` is a simple wrapper to send HTTP requests to a graphql endpoint. The request methods can be used with a manually constructed query, or by using the methods available in the `'Mutations'` section.

Use `trompace.connection.submit_query()` to send a graphql command to the CE.

`trompace.connection.submit_query(querystr: str, auth_required=False)`
Submit a query to the CE.

Parameters

- **querystr** – The query to be submitted
- **auth_required** – If `True`, send an authentication key with this request. Required if the CE has authentication enabled and you are sending a mutation. Not required for queries. The request is not authenticated if the `auth.required` config item is `False`

```
from trompace.mutations import person
from trompace.connection import submit_query
query = person.mutation_create_person(name="Gustav Mahler", ...)
submit_query(query, auth_required=True)
```

1.2 Demos

We provide demo scripts which show how to use the library and how to structure the Trompa CE data model.

The scripts can optionally output sample graphql queries so that they can be copied and modified, and can also submit the sample queries to a CE instance. The scripts also include some comments explaining specific data modelling decisions that were made for some types.

Use the `--print` flag to output the graphql queries that are being executed, and `--submit` to send them to the configured CE:

```
TROMPACE_CLIENT_CONFIG=trompace.ini python -m demo.person --print --submit
```

1.2.1 DefinedTerm

Shows how to create a fixed vocabulary of folksonomy tags/categories, or a fixed vocabulary of annotation motivations for use in Web Annotations using `schema:DefinedTermSet` and `schema:DefinedTerm`.

```
python -m demo.definedterm --print --submit
```

1.2.2 MediaObject

Shows how to create `schema:MediaObjects` which refer to files in the CE ecosystem (scores, images, audio, etc), as well as the required links between `MediaObjects` which are related to each other.

```
python -m demo.mediaobject --print --submit
```


1.2.3 MusicComposition

Shows how to create a `schema:MusicComposition`, referring to a musical work. Also shows how to indicate movements of larger works, and composers.

```
python -m demo.musiccomposition --print --submit
```

1.2.4 Person

Shows how to create `schema:Person` objects, referring to natural people. Shows how to link together references to the same `Person` on different websites.

```
python -m demo.person --print --submit
```

1.2.5 Rating

Shows how to create `schema:Rating` objects and rating templates, for use in Web annotations,

```
python -m demo.rating --print --submit
```

1.2.6 Annotation Body

Shows how to create `oa:Annotation` objects and different types of values for `oa:hasBody` (external URLs, `oa:TextualBody`, and links to existing nodes in the CE).

```
python -m demo.annotation.annotationbody --print --submit
```

1.2.7 Annotation Fixed Vocabulary

Shows how to create `oa:Annotation` objects and the use of a fixed `schema:DefinedTerm` as an `oa:hasBody`.

```
python -m demo.annotation.annotationfixedvocab --print --submit
```

1.2.8 Annotation Linking

Shows how to create multiple `oa:Annotation` objects and link them together into a “meta” annotation.

```
python -m demo.annotation.annotationlinking --print --submit
```

1.2.9 Annotation Motivation

Shows how to create `oa:Annotation` objects and the use of a fixed `schema:DefinedTerm` as an `oa:motivatedBy`.

```
python -m demo.annotation.annotationmotivation --print --submit
```

1.2.10 Annotation Rating

Shows how to create `oa:Annotation` objects and the use of a `schema:Rating` as an `oa:hasBody`, along with links to the Rating definition that the rating was derived from (`prov:wasDerivedFrom`).

```
python -m demo.annotation.annotationrating --print --submit
```

1.2.11 Annotation Tag

Shows how to create `oa:Annotation` objects and a freeform text tag.

```
python -m demo.annotation.annotationtarget --print --submit
```

1.2.12 Annotation Target

Shows how to create `oa:Annotation` objects and different types of values for `oa:hasTarget` (external URLs, existing nodes in the CE, URLs specified by fields of nodes in the CE).

```
python -m demo.annotation.annotationbody --print --submit
```

1.2.13 Annotation Session

Shows how to add many `oa:Annotation` objects to a single logical session.

```
python -m demo.annotation.session --print --submit
```

1.2.14 Annotation Toolkit

Shows how to join together a collection of Rating templates, Fixed Vocabularies, and Motivations into an annotation toolkit.

```
python -m demo.annotation.toolkit --print --submit
```

1.3 Mutations

Mutations are queries used to create or update data in the CE.

1.3.1 AudioObject

```
trompace.mutations.audioobject.mutation_create_audioobject (*, title: str, contributor: str, creator: str, source: str, format_: str, name: str = None, description: str = None, date: str = None, encodingformat: str = None, embedurl: str = None, url: str = None, contenturl: str = None, language: str = None, inlanguage: str = None, license: str = None, subject: str = None)
```

Returns a mutation for creating a AudioObject.

Parameters

- **name** – The name of the audio object.
- **description** – An account of the audio object.
- **date** – The date associated with the audio object.
- **creator** – The person, organization or service who created the thing the web resource is about.
- **contributor** – A person, an organization, or a service responsible for contributing the audio object to the web resource. This can be either a name or a base URL.
- **format_** – A MimeType of the format of the page describing the audio object.
- **encodingformat** – A MimeType of the format of object encoded by the audio object.
- **source** – The URL of the web resource about this audio object. If no such resource is available, use the same value as contentUrl.
- **subject** – The subject of the audio object.
- **contenturl** – The URL of the content encoded by the audio object.
- **url** –
- **license** –
- **language** – The language the metadata is written in. Currently supported languages are en,es,ca,nl,de,fr
- **inlanguage** – The language of the audio object.
- **title** – The title of the resource indicated by *source*

Returns The string for the mutation for creating the AudioObject.

Raises

- **UnsupportedLanguageException** if the input language is not one of the supported languages. –
- **ValueError** if a required argument is None –

`trompace.mutations.audioobject.mutation_delete_audioobject` (*identifier: str*)

Returns a mutation for deleting a AudioObject based on the identifier.

Parameters *identifier* – The unique identifier of the AudioObject.

Returns The string for the mutation for deleting the AudioObject based on the identifier.

`trompace.mutations.audioobject.mutation_merge_audioobject_encoding` (*audioobject_identifier: str, audiob-
ject_derivative_identifier: str*)

Returns a mutation for indicating that a derivative AudioObject *encodes* a primary AudioObject (<https://schema.org/encoding>). For example a transcription of a score is an *encoding* of that score.

Parameters

- **audioobject_identifier** – The unique identifier of the “main” AudioObject.
- **audioobject_derivative_identifier** – The unique identifier of the AudioObject which is the encoding.

Returns A GraphQL mutation for MergeAudioObjectEncoding.

`trompace.mutations.audioobject.mutation_merge_audioobject_exampleofwork` (*audioobject_identifier: str, work_identifier: str*)

Returns a mutation for indicating that a AudioObject is an example of a work (<https://schema.org/exampleOfWork>).

Parameters

- **audioobject_identifier** – The unique identifier of the AudioObject.
- **work_identifier** – The unique identifier of the work that the AudioObject is an example of.

Returns A GraphQL mutation for MergeAudioObjectExampleOfWork.

`trompace.mutations.audioobject.mutation_remove_audioobject_encoding` (*audioobject_identifier: str, audiob-
ject_derivative_identifier: str*)

Returns a mutation for removing that a derivative AudioObject *encodes* a primary AudioObject (<https://schema.org/encoding>). For example a transcription of a score is an *encoding* of that score.

Parameters

- **audioobject_identifier** – The unique identifier of the “main” AudioObject.
- **audioobject_derivative_identifier** – The unique identifier of the AudioObject which is the encoding.

Returns A GraphQL mutation for RemoveAudioObjectEncoding.

```
trompace.mutations.audioobject.mutation_remove_audioobject_exampleofwork (audioobject_identifier:
                                                                    str,
                                                                    work_identifier:
                                                                    str)
```

Returns a mutation for removing that a AudioObject is an example of a work (<https://schema.org/exampleOfWork>).

Parameters

- **audio_object_identifier** – The unique identifier of the AudioObject.
- **work_identifier** – The unique identifier of the work that the AudioObject is an example of.

Returns A GraphQL mutation for RemoveAudioObjectExampleOfWork.

```
trompace.mutations.audioobject.mutation_update_audioobject (identifier: str, *,
name: str = None,
title: str = None,
description: str = None, date: str = None, creator: str = None, contributor: str = None, format_: str = None, encoding-format: str = None, source: str = None, license: str = None, subject: str = None, contenturl: str = None, language: str = None, inlanguage: str = None)
```

Returns a mutation for updating a AudioObject.

Parameters

- **identifier** – The identifier of the AudioObject in the CE to be updated.
- **name** – The name of the audio object.
- **description** – An account of the audio object.
- **date** – The date associated with the audio object.
- **creator** – The person, organization or service who created the thing the web resource is about.
- **contributor** – A person, an organization, or a service responsible for contributing the audio object to the web resource. This can be either a name or a base URL.
- **format_** – A MimeType of the format of the page describing the audio object.
- **encodingformat** – A MimeType of the format of object encoded by the audio object.
- **source** – The URL of the web resource about this audio object. If no such resource is available, use the same value as contentUrl.
- **subject** – The subject of the audio object.
- **contenturl** – The URL of the content encoded by the audio object.
- **url** –

- **license** –
- **language** – The language the metadata is written in. Currently supported languages are en,es,ca,nl,de,fr
- **inlanguage** – The language of the audio object.
- **title** – The title of the resource indicated by *source*

Returns The string for the mutation for updating the AudioObject.

Raises Assertion error if the input language or inlanguage is not one of the supported languages. –

1.3.2 ControlAction

```
trompace.mutations.controlaction.mutation_add_actioninterface_result (controlaction_id:  
                                                                    str,  
                                                                    thing_interface_id:  
                                                                    str)
```

Returns a mutation for linking a control action object and a thing interface with the result relation :param controlaction_id: The unique identifier of the control action. :param thing_interface_id: the unique identifier of a thing interface object

Returns The string for the mutation for adding a control action to an object.

```
trompace.mutations.controlaction.mutation_add_controlaction_additionalproperty (controlaction_id:  
                                                                    str,  
                                                                    prop-  
                                                                    erty_id:  
                                                                    str)
```

Returns a mutation for adding a control action to a property value specification. :param controlaction_id: The unique identifier of the control action. :param property_id: The unique identifier of the property.

Returns The string for the mutation for adding a control action to a property.

```
trompace.mutations.controlaction.mutation_add_controlaction_object (controlaction_id:  
                                                                    str, ob-  
                                                                    ject_id:  
                                                                    str)
```

Returns a mutation for adding a control action to an object, either a property value specification or a property :param controlaction_id: The unique identifier of the control action. :param object_id: The unique identifier of the object (property / property value specification).

Returns The string for the mutation for adding a control action to an object.

```
trompace.mutations.controlaction.mutation_add_entrypoint_controlaction (entrypoint_id:  
                                                                    str,  
                                                                    con-  
                                                                    tro-  
                                                                    lac-  
                                                                    tion_id:  
                                                                    str)
```

Returns a mutation for adding an entry point object to a control action :param entrypoint_id: The unique identifier of the entry point. :param controlaction_id: The unique identifier of the control action.

Returns The string for the mutation for adding an entry point object to a control action

`trompace.mutations.controlaction.mutation_add_propertyvaluespecification_potentialaction` (*pr*

Returns a mutation for adding a control action to a property value specification. :param controlaction_id: The unique identifier of the control action. :param propertyvaluespecification_id: The unique identifier of the property value specification.

Returns The string for the mutation for adding a control action to a property value specification.

`trompace.mutations.controlaction.mutation_create_controlaction` (*name:* *str*,
actionstatus:
trompace.constants.ActionStatusType
= <ActionSta-
tusType.PotentialActionStatus:
4>, description:
str = None)

Returns a mutation for creating a ControlAction. If an action status is not set, defaults to a PotentialActionStatus :param name: The name of the ControlAction. :param description: An account of the ControlAction. :param actionstatus: The status of the ControlAction.

Returns The string for the mutation for creating the ControlAction object.

`trompace.mutations.controlaction.mutation_modify_controlaction` (*controlaction_id:*
str, actionstatus:
trompace.constants.ActionStatusType,
error: str =
None)

Returns a mutation for modifying the status and errors of the ControlAction :param controlaction_id: The unique identifier of the ControlAction. :param actionstatus: the status to update to. :param error: An error to set if the actionstatus is FailedActionStatus

Returns The string for the mutation for modifying a ControlAction.

`trompace.mutations.controlaction.mutation_request_controlaction` (*controlaction_id:*
str, entry-
point_id: str,
properties:
list, prop-
ertyValues:
list)

Returns a mutation for requesting a control action controlaction_id: The unique identifier of the control action. entypoint_id: The unique identifier of the entry point linked to this control action properties: a list of the properties objects of the control action propertyValues: a list of the properties objects of the control action

Returns The string for the mutation for requesting a control action.

`trompace.mutations.controlaction.mutation_update_controlaction_status` (*controlaction_id:*
str,
ac-
tion_status:
trompace.constants.ActionStat

Returns a mutation for updating the status of a control action to an object :param controlaction_id: The unique identifier of the control action. :param action_status: the action status enum object.

Returns The string for the mutation for adding a control action to an object.

1.3.3 DefinedTerm

```
trompace.mutations.definedterm.create_defined_term(*, creator: str, term-  
code: str, additionaltype:  
List[str], broader_url: str  
= None, broader_schema:  
trompace.mutations.annotation.AnnotationSchemaMotivati  
= None, image: str = None)
```

Return a mutation for making a DefinedTerm. A DefinedTerm (<https://schema.org/DefinedTerm>) is a word, name, acronym, phrase, etc. with a formal definition. It is part of a DefinedTermSet.

Parameters

- **creator** – a URI to the identity of the user who created this DefinedTerm
- **termcode** – The name of this term
- **additionaltype** – A list of schema.org additionalTypes used to categorise this DefinedTerm
- **broader_url** (*optional*) – a type to be related with skos:broader
- **broader_schema** (*optional*) – an annotation motivation to be related with skos:broader
- **image** (*optional*) – an image to describe this DefinedTerm

Returns A GraphQL Mutation to create a DefinedTerm in the Trompa CE

```
trompace.mutations.definedterm.create_defined_term_set(*, creator: str, name: str,  
additionaltype: List[str],  
image: str = None)
```

Return a mutation for making a DefinedTermSet. A DefinedTermSet (<https://schema.org/DefinedTermSet>) is a group of defined terms, e.g. categories or labels.

Parameters

- **creator** – a URI to the identity of the user who created this DefinedTermSet
- **name** – the name of this DefinedTermSet
- **additionaltype** – A list of schema.org additionalTypes used to categorise this DefinedTermSet
- **image** (*optional*) – an image to describe this DefinedTermSet

Returns A GraphQL Mutation to create a DefinedTermSet in the Trompa CE

```
trompace.mutations.definedterm.defined_term_add_to_defined_term_set(*, de-  
fined_term_set:  
str, de-  
fined_term:  
str)
```

Return a mutation for adding a DefinedTerm to a DefinedTermSet.

Parameters

- **defined_term_set** – The identifier of the DefinedTermSet to add to
- **defined_term** – The identifier of the DefinedTermSet to add

Returns A GraphQL Mutation to add a DefinedTerm to the DefinedTermSet


```
trompace.mutations.definedterm.delete_defined_term(identifier: str)
```

Return a mutation for deleting a DefinedTermSet.

Parameters *identifier* – The identifier of the DefinedTerm to delete

Returns A GraphQL Mutation to delete a DefinedTerm from the Trompa CE

```
trompace.mutations.definedterm.delete_defined_term_set(identifier: str)
```

Return a mutation for deleting a DefinedTermSet.

Parameters *identifier* – The identifier of the DefinedTermSet to delete

Returns A GraphQL Mutation to delete a DefinedTermSet from the Trompa CE

```
trompace.mutations.definedterm.update_defined_term(identifier: str, *, creator:
                                                    str = None, termcode: str =
                                                    None, additionaltype: List[str]
                                                    = None, broader_url: str
                                                    = None, broader_schema:
                                                    trompace.mutations.annotation.AnnotationSchemaMotivati
                                                    = None, image: str = None)
```

Return a mutation for updating a DefinedTerm.

TODO: Copy arguments from **create_**

Returns A GraphQL Mutation to update a DefinedTerm in the Trompa CE

```
trompace.mutations.definedterm.update_defined_term_set(identifier: str, *, creator: str
                                                         = None, name: str = None,
                                                         additionaltype: List[str] =
                                                         None, image: str = None)
```

Return a mutation for updating a DefinedTermSet.

TODO: Copy arguments from **create_**

Returns A GraphQL Mutation to update a DefinedTermSet in the Trompa CE

1.3.4 DigitalDocument

```
trompace.mutations.digitaldocument.mutation_create_digitaldocument(*, title:
                                                                    str, con-
                                                                    tributor:
                                                                    str, cre-
                                                                    ator: str,
                                                                    source:
                                                                    str, for-
                                                                    mat_: str,
                                                                    subject:
                                                                    str =
                                                                    None, lan-
                                                                    guage: str
                                                                    = None,
                                                                    descrip-
                                                                    tion: str =
                                                                    None)
```

Returns a mutation for creating a digital document object.

Parameters

- **title** – The title of the resource indicated by *source*

- **contributor** – The main URL of the site where the information about this DigitalDocument was taken from
- **creator** – The person, organization or service who is creating this DigitalDocument (e.g. URL of the software)
- **source** – The URL of the web resource where information about this DigitalDocument is taken from
- **format_** – The mimetype of the resource indicated by *source*
- **subject** (*optional*) – The subject of the music composition.
- **language** (*optional*) – The language the metadata is written in. Currently supported languages are en,es,ca,nl,de,fr
- **description** (*optional*) – A description about the item referred to by source.

Returns The string for the mutation for creating the digital document.

Raises

- **UnsupportedLanguageException** if the input language is not one of the supported languages. –
- **NotAMimeTypeException** – if *format_* is not a valid mimetype.

`trompace.mutations.digitaldocument.mutation_delete_digitaldocument` (*identifier:*
str)

Returns a mutation for deleting a digital document object based on the identifier.

Parameters *identifier* – The unique identifier of the digital document object.

Returns The string for the mutation for deleting the digital document object based on the identifier.

`trompace.mutations.digitaldocument.mutation_update_digitaldocument` (*identifier:*
str,
title: *str*
= None,
contributor: *str* *= None*,
creator: *str* *= None*,
source: *str* *= None*,
format_: *str* *= None*,
subject: *str* *= None*,
language: *str* *= None*,
description: *str* *= None*)

Returns a mutation for updating a digital document object.

Parameters

- **identifier** – The identifier of the media object in the CE to be updated.

- **title** – The title of the resource indicated by *source*
- **contributor** – The main URL of the site where the information about this DigitalDocument was taken from
- **creator** – The person, organization or service who is creating this DigitalDocument (e.g. URL of the software)
- **source** – The URL of the web resource where information about this DigitalDocument is taken from
- **format_** – The mimetype of the resource indicated by *source*
- **subject** (*optional*) – The subject of the music composition.
- **language** (*optional*) – The language the metadata is written in. Currently supported languages are en,es,ca,nl,de,fr
- **description** (*optional*) – A description about the item referred to by source.

Returns The string for the mutation for creating the artist.

Raises Assertion error if the input language is not one of the supported languages. –

1.3.5 EntryPoint

```
trompace.mutations.entrypoint.mutation_create_entry_point(*, name: str, contributor: str, subject: str, creator: str, source: str, language: str, actionPlatform: str, contentType: List, encodingType: list, formatin='text/html', identifier=None, description: str = None)
```

Returns a mutation for creating an entry point object :param name: The name of the entry point. :param contributor: A person, an organization, or a service responsible for contributing the aentry point to the web resource. This can be either a name or a base URL. :param creator: The person, organization or service who created the thing the entry point is about. :param source: The URL of the web resource to be represented by the node. :param description: An account of the entry point.. :param language: The language the metadata is written in. Currently supported languages are en,es,ca,nl,de,fr. :param actionPlatform: The action platform. :param contentType: The content type associated with the entry point, should be a mimetype. :param encodingType: The encoding type associated with the entry point, should be a mimetype.

Returns The string for the mutation for creating the artist.

Raises Assertion error if the input language is not one of the supported languages. –

1.3.6 ItemList

`trompace.mutations.itemlist.mutation_add_itemlist_itemlist_element` (*itemlist_id:*
str, element_id:
str)

Returns a mutation for adding a ThingInterface in an ItemList object based on the identifier. (<https://schema.org/itemListElement>)

Parameters

- **itemlist_id** – The unique identifier of the ItemList object.
- **element_id** – The unique identifier of the ThingInterface object.

Returns The string for the mutation for adding an ThingInterface in an ItemList object based on the identifier,

`trompace.mutations.itemlist.mutation_add_listitem_item` (*listitem_id:* *str, item_id:*
str)

Returns a mutation for adding an Item to a ListItem object based on the identifier. (<https://schema.org/item>)

Parameters

- **listitem_id** – The unique identifier of the ListItem object.
- **item_id** – The unique identifier of the Item object.

Returns The string for the mutation for adding a Item to a ListItem object based on the identifier.

`trompace.mutations.itemlist.mutation_add_listitem_nextitem` (*listitem_id:* *str, nextitem_id:* *str*)

Returns a mutation for adding a NextItem to a ListItem object based on the identifier. (<https://schema.org/nextItem>)

Parameters

- **listitem_id** – The unique identifier of the ListItem object.
- **nextitem_id** – The unique identifier of the NextItem object.

Returns The string for the mutation for adding a NextItem to a ListItem object based on the identifier.

`trompace.mutations.itemlist.mutation_create_itemlist` (*name:* *str, creator:*
str, itemlistorder:
trompace.constants.ItemListOrderType
= <ItemListOrderType.ItemListUnordered:
3>, description: *str = None,*
contributor: *str = None,*
additionaltype: *List[str] =*
None)

Returns a mutation for creating an ItemList object. (<https://schema.org/ItemList>)

Parameters

- **name** – The name of the ItemList object.
- **creator** – The person, organization or service who created the ItemList.

- **itemlistorder** – The type of ordering for the list (ascending, descending, unordered, ordered)
- **contributor** (*optional*) – A person, an organization, or a service responsible for contributing the ItemList to the web resource. This can be either a name or a base URL.
- **description** (*optional*) – The description of the ItemList object
- **additionaltype** (*optional*) – A list of schema.org additionalTypes used to categorise this ItemList

Returns The string for the mutation for creating the ItemList.

```
trompace.mutations.itemlist.mutation_create_listitem(creator: str, name: str = None,
                                                       contributor: str = None, de-
                                                       scription: str = None, ite-
                                                       murl: str = None, position: Op-
                                                       tional[int] = None)
```

Returns a mutation for creating a ListItem object. (<https://schema.org/ListItem>)

Parameters

- **name** – The name of the ListItem object.
- **creator** – The person, organization or service who created the ListItem.
- **contributor** – A person, an organization, or a service responsible for contributing the ListItem to the web resource. This can be either a name or a base URL.
- **description** – The description of the ListItem object
- **itemurl** – If the item of this ListItem points to a URL outside of the CE, the item URL
- **position** – the position of the ListItem

Returns The string for the mutation for creating the ListItem.

```
trompace.mutations.itemlist.mutation_delete_itemlist(identifier: str)
```

Returns a mutation for deleting an ItemList object based on the identifier. (<https://schema.org/ItemList>)

Parameters **identifier** – The unique identifier of the ItemList object.

Returns The string for the mutation for deleting the ItemList object based on the identifier.

```
trompace.mutations.itemlist.mutation_delete_listitem(identifier: str)
```

Returns a mutation for deleting a ListItem object based on the identifier. (<https://schema.org/ListItem>)

Parameters **identifier** – The unique identifier of the ListItem object.

Returns The string for the mutation for deleting the ListItem object based on the identifier.

```
trompace.mutations.itemlist.mutation_remove_itemlist_itemlist_element(itemlist_id:
                                                                           str,
                                                                           ele-
                                                                           ment_id:
                                                                           str)
```

Returns a mutation for removing a ThingInterface from an ItemList object based on the identifier. (<https://schema.org/itemListElement>)

Parameters

- **itemlist_id** – The unique identifier of the ItemList object.
- **element_id** – The unique identifier of the ThingInterface object.

Returns The string for the mutation for removing an ThingInterface from an ItemList object based on the identifier.

```
trompace.mutations.itemlist.mutation_remove_listitem_item(listitem_id: str,  
                                                           item_id: str)
```

Returns a mutation for removing a Item from a ListItem object based on the identifier. (<https://schema.org/item>)

Parameters

- **listitem_id** – The unique identifier of the ListItem object.
- **item_id** – The unique identifier of the Item object.

Returns The string for the mutation for removing a Item from a ListItem object based on the identifier.

```
trompace.mutations.itemlist.mutation_remove_listitem_nextitem(listitem_id: str,  
                                                                nextitem_id: str)
```

Returns a mutation for removing a NextItem from a ListItem object based on the identifier. (<https://schema.org/nextItem>)

Parameters

- **listitem_id** – The unique identifier of the ListItem object.
- **nextitem_id** – The unique identifier of the NextItem object.

Returns The string for the mutation for removing a NextItem from a ListItem object based on the identifier.

```
trompace.mutations.itemlist.mutation_sequence_add_itemlist_itemlist_element(itemlist_id:  
                                                                              str,  
                                                                              el-  
                                                                              e-  
                                                                              ment_ids:  
                                                                              list)
```

Returns a mutation for adding a sequence of ThingInterface in an ItemList object based on the identifiers. (<https://schema.org/itemListElement>)

Parameters

- **itemlist_id** – The unique identifier of the ItemList object.
- **element_ids** – The list of unique identifiers of the ThingInterface objects.

Returns The string for the mutation for adding a sequence of ThingInterface in an ItemList object based on the identifiers.

```
trompace.mutations.itemlist.mutation_sequence_add_listitem_item(listitem_ids:  
                                                                  list, item_ids:  
                                                                  list)
```

Returns a mutation for adding a sequence of ThingInterface in an ListItem objects based on the identifiers. (<https://schema.org/itemListElement>)

Parameters

- **listitem_ids** – The list of unique identifiers of the ListItem objects.
- **item_ids** – The list of unique identifiers of the ThingInterface objects.

Returns The string for the mutation for adding a sequence of ThingInterface in an ListItem objects based on the identifiers.

```
trompace.mutations.itemlist.mutation_sequence_add_listitem_nextitem(listitem_ids: list)
```

Returns a mutation for adding a sequence of NextItem to an ListItem objects based on the identifiers. (<https://schema.org/itemListElement>)

Parameters **listitem_ids** – The list of unique identifier of the ListItem objects.

Returns The string for the mutation for adding a sequence of NextItem to an ListItem objects based on the identifiers.

```
trompace.mutations.itemlist.mutation_sequence_create_listitem(listitems: list,
                                                                contributor: str,
                                                                creator: str = None,
                                                                name: str = None,
                                                                description: list = None)
```

Returns a mutation for creating a sequence of ListItem objects (<https://schema.org/itemListElement>)

Parameters

- **listitems** – the ListItem objects to create
- **creator** – The person, organization or service who created the items.
- **contributor** – A person, an organization, or a service responsible for contributing the ListItem to the web resource. This can be either a name or a base URL.
- **name** – The name of the ListItem object.

Returns The string for the mutation for creating a sequence of ListItem objects

```
trompace.mutations.itemlist.mutation_update_itemlist(identifier: str, name: str = None,
                                                       creator: str = None,
                                                       itemlistorder: trompace.constants.ItemListOrderType = None,
                                                       description: str = None,
                                                       contributor: str = None,
                                                       additionaltype: List[str] = None)
```

Returns a mutation for updating an ItemList object. (<https://schema.org/ItemList>)

Parameters

- **identifier** – The identifier of the ItemList in the CE to be updated.
- **name** – The name of the ItemList object.
- **creator** – The person, organization or service who created the ItemList.
- **itemlistorder** – The type of ordering for the list (ascending, descending, unordered, ordered)
- **contributor** (*optional*) – A person, an organization, or a service responsible for contributing the ItemList to the web resource. This can be either a name or a base URL.
- **description** (*optional*) – The description of the ItemList object
- **additionaltype** (*optional*) – A list of schema.org additionalTypes used to categorise this ItemList

Returns The string for the mutation for updating the ItemList.

```
trompace.mutations.itemlist.mutation_update_listitem(identifier: str, creator: str =  
None, name: str = None, con-  
tributor: str = None, descrip-  
tion: str = None, itemurl: str  
= None, position: int = None)
```

Returns a mutation for updating a ListItem object. (<https://schema.org/ListItem>)

Parameters

- **identifier** – The identifier of the ListItem in the CE to be updated.
- **name** – The name of the ListItem object.
- **creator** – The person, organization or service who created the ListItem.
- **contributor** – A person, an organization, or a service responsible for contributing the ListItem to the web resource. This can be either a name or a base URL.
- **description** – The description of the ListItem object
- **itemurl** – If the item of this ListItem points to a URL outside of the CE, the item URL
- **position** – the position of the ListItem

Returns The string for the mutation for updating the ListItem.

1.3.7 MediaObject

```
trompace.mutations.mediaobject.mutation_add_media_object_used(mediaobject_identifier:  
str,  
thing_identifier:  
str)
```

Returns a mutation for indicating that a MediaObject *used* some other thing to create it (<http://www.w3.org/ns/prov#used>). For example an MEI file covered from musicxml might use the verovio software.

Parameters

- **mediaobject_identifier** – The unique identifier of a MediaObject.
- **thing_identifier** – The unique identifier of some Thing that was used to create the MediaObject.

Returns A GraphQL mutation for MergeMediaObjectUsed.

```
trompace.mutations.mediaobject.mutation_create_media_object(*, title: str, con-  
tributor: str, cre-  
ator: str, source: str,  
format_: str, name:  
str = None, descrip-  
tion: str = None,  
date: str = None, en-  
codingformat: str =  
None, embedurl: str  
= None, url: str =  
None, contenturl: str  
= None, language:  
str = None, inlan-  
guage: str = None,  
license: str = None)
```

Returns a mutation for creating a media object object.

Parameters

- **name** – The name of the media object.
- **description** – An account of the media object.
- **date** – The date associated with the media object.
- **creator** – The person, organization or service who created the thing the web resource is about.
- **contributor** – A person, an organization, or a service responsible for contributing the media object to the web resource. This can be either a name or a base URL.
- **format_** – A MimeType of the format of the page describing the media object.
- **encodingformat** – A MimeType of the format of object encoded by the media object.
- **source** – The URL of the web resource about this media object. If no such resource is available, use the same value as `contentUrl`.
- **subject** – The subject of the media object.
- **contenturl** – The URL of the content encoded by the media object.
- **url** –
- **license** –
- **language** – The language the metadata is written in. Currently supported languages are en,es,ca,nl,de,fr
- **inLanguage** – The language of the media object.
- **title** – The title of the resource indicated by *source*

Returns The string for the mutation for creating the media object.

Raises `UnsupportedLanguageException` if the input language is not one of the supported languages. –

`trompace.mutations.mediaobject.mutation_delete_media_object` (*identifier: str*)

Returns a mutation for deleting a media object object based on the identifier.

Parameters **identifier** – The unique identifier of the media object object.

Returns The string for the mutation for deleting the media object object based on the identifier.

`trompace.mutations.mediaobject.mutation_merge_media_object_encoding` (*mediaobject_identifier: str, mediaobject_derivative_identifier: str*)

Returns a mutation for indicating that a derivative `MediaObject` *encodes* a primary `MediaObject` (<https://schema.org/encoding>). For example a transcription of a score is an *encoding* of that score.

Parameters

- **mediaobject_identifier** – The unique identifier of the “main” `MediaObject`.
- **mediaobject_derivative_identifier** – The unique identifier of the `MediaObject` which is the encoding.

Returns A GraphQL mutation for `MergeMediaObjectEncoding`.

```
trompace.mutations.mediaobject.mutation_merge_media_object_wasderivedfrom(mediaobject_derivative_  
str,  
me-  
diaob-  
ject_source_identifier:  
str)
```

Returns a mutation for indicating that a MediaObject *was derived from* a primary MediaObject (<http://www.w3.org/ns/prov#wasDerivedFrom>). For example a PDF generated from a MusicXML source file was derived from that source file.

Parameters

- **mediaobject_derivative_identifier** – The unique identifier of the MediaObject which is the derivative.
- **mediaobject_source_identifier** – The unique identifier of the “source” MediaObject.

Returns A GraphQL mutation for MergeMediaObjectWasDerivedFrom.

```
trompace.mutations.mediaobject.mutation_merge_mediaobject_example_of_work(mediaobject_identifier:  
str,  
work_identifier:  
str)
```

Returns a mutation for indicating that a MediaObject is an example of a work (<https://schema.org/exampleOfWork>).

Parameters

- **mediaobject_identifier** – The unique identifier of the media object.
- **work_identifier** – The unique identifier of the work that the media object is an example of.

Returns A GraphQL mutation for MergeMediaObjectExampleOfWork.

```
trompace.mutations.mediaobject.mutation_remove_media_object_encoding(mediaobject_identifier:  
str, me-  
diaob-  
ject_derivative_identifier:  
str)
```

Returns a mutation for removing that a derivative MediaObject *encodes* a primary MediaObject (<https://schema.org/encoding>).

Parameters

- **mediaobject_identifier** – The unique identifier of the “main” MediaObject.
- **mediaobject_derivative_identifier** – The unique identifier of the MediaObject which is the encoding.

Returns A GraphQL mutation for RemoveMediaObjectEncoding.

```
trompace.mutations.mediaobject.mutation_remove_media_object_wasderivedfrom(mediaobject_derivative_  
str,  
me-  
diaob-  
ject_source_identifier:  
str)
```

Returns a mutation for removing that a MediaObject *was derived from* a primary MediaObject (<http://www.w3.org/ns/prov#wasDerivedFrom>).

Parameters

- **mediaobject_derivative_identifier** – The unique identifier of the MediaObject which is the derivative.
- **mediaobject_source_identifier** – The unique identifier of the “source” MediaObject.

Returns A GraphQL mutation for RemoveMediaObjectWasDerivedFrom.

```
trompace.mutations.mediaobject.mutation_remove_mediaobject_example_of_work (mediaobject_identifier:
                                                                    str,
                                                                    work_identifier:
                                                                    str)
```

Returns a mutation for removing that a MediaObject is an example of a work (<https://schema.org/exampleOfWork>).

Parameters

- **mediaobject_identifier** – The unique identifier of the media object.
- **work_identifier** – The unique identifier of the work that the media object is an example of.

Returns A GraphQL mutation for RemoveMediaObjectExampleOfWork.

```
trompace.mutations.mediaobject.mutation_update_media_object (identifier: str, *,
                                                                name: str = None,
                                                                title: str = None,
                                                                description: str =
                                                                None, date: str =
                                                                None, creator: str
                                                                = None, contribu-
                                                                tor: str = None, for-
                                                                mat_: str = None,
                                                                encodingformat: str
                                                                = None, source: str
                                                                = None, license: str
                                                                = None, subject: str
                                                                = None, url: str =
                                                                None, contenturl: str
                                                                = None, language:
                                                                str = None, inlan-
                                                                guage: str = None)
```

Returns a mutation for updating a media object object.

Parameters

- **identifier** – The identifier of the media object in the CE to be updated.
- **name** – The name of the media object.
- **description** – An account of the media object.
- **date** – The date associated with the media object.
- **creator** – The person, organization or service who created the thing the web resource is about.
- **contributor** – A person, an organization, or a service responsible for contributing the media object to the web resource. This can be either a name or a base URL.
- **format_** – A MimeType of the format of the page describing the media object.
- **encodingformat** – A MimeType of the format of object encoded by the media object.

- **source** – The URL of the web resource about this media object. If no such resource is available, use the same value as `contentUrl`.
- **subject** – The subject of the media object.
- **contenturl** – The URL of the content encoded by the media object.
- **url** –
- **license** –
- **language** – The language the metadata is written in. Currently supported languages are `en,es,ca,nl,de,fr`
- **inLanguage** – The language of the media object.
- **title** – The title of the resource indicated by *source*

Returns The string for the mutation for updating the media object.

Raises **Assertion error if the input language or inLanguage is not one of the supported languages.** –

1.3.8 MusicComposition

```
trompace.mutations.musiccomposition.mutation_create_music_composition(*,
                                                                    title:
                                                                    str,
                                                                    con-
                                                                    trib-
                                                                    utor:
                                                                    str,
                                                                    cre-
                                                                    ator:
                                                                    str,
                                                                    source:
                                                                    str,
                                                                    for-
                                                                    mat_:
                                                                    str,
                                                                    sub-
                                                                    ject:
                                                                    str =
                                                                    None,
                                                                    lan-
                                                                    guage:
                                                                    str =
                                                                    None,
                                                                    inlan-
                                                                    guage:
                                                                    str =
                                                                    None,
                                                                    name:
                                                                    str =
                                                                    None,
                                                                    de-
                                                                    scrip-
                                                                    tion:
                                                                    str =
                                                                    None,
                                                                    posi-
                                                                    tion:
                                                                    int =
                                                                    None)
```

Returns a mutation for creating a music composition object

Parameters

- **title** – The title of the resource indicated by *source*
- **contributor** – The main URL of the site where the information about this MusicComposition was taken from
- **creator** – The person, organization or service who is creating this MusicComposition (e.g. URL of the software)
- **source** – The URL of the web resource where information about this MusicComposition is taken from
- **format_** – The mimetype of the resource indicated by *source*

- **subject** – The subject of the music composition.
- **language** – The language the metadata is written in. Currently supported languages are en,es,ca,nl,de,fr
- **inlanguage** – The language of the music composition.
- **name** – The name of the music composition.
- **description** – An account of the music composition.
- **position** – In the case that this is a movement of a larger work (e.g. a Symphony), the position of this MusicComposition in the larger one.

Returns The string for the mutation for creating the music composition.

Raises

- **UnsupportedLanguageException** – if language is not one of the supported languages.
- **NotAMimeTypeException** – if format_ is not a valid mimetype

trompace.mutations.musiccomposition.mutation_delete_music_composition(*identifier: str*)

Returns a mutation for deleting a MusicComposition.

Parameters *identifier* – The identifier of the MusicComposition.

Returns The string for the mutation for deleting the music composition object based on the identifier.

trompace.mutations.musiccomposition.mutation_merge_music_composition_composer(*composition_identifier, person_identifier*)

Returns a mutation for adding a Person as the composer of a MusicComposition (<https://schema.org/composer>).

Parameters

- **composition_identifier** – The identifier of a MusicComposition.
- **person_identifier** – The identifier of a Person who composed the MusicComposition.

trompace.mutations.musiccomposition.mutation_merge_music_composition_exact_match(*from_identifier, to_identifier*)

Returns a mutation for indicating that two MusicComposition objects represent the same composition (<http://www.w3.org/2004/02/skos/core#exactMatch>).

Parameters

- **from_identifier** – The identifier of one MusicComposition.
- **to_identifier** – The identifier of another MusicComposition.

trompace.mutations.musiccomposition.mutation_merge_music_composition_has_part(*main_identifier, part_identifier*)

Returns a mutation for adding a MusicComposition as a part of another MusicComposition (<https://schema.org/hasPart>). For example, the first movement of a composition could be represented as a hasPart of a main symphony.

Parameters

- **main_identifier** – The identifier of a main MusicComposition.
- **part_identifier** – The identifier of a MusicComposition which is a part of the main MusicComposition.

`trompace.mutations.musiccomposition.mutation_merge_music_composition_included_composition` (*music_composition_id*, *included_composition_id*)

Returns a mutation for adding a MusicComposition as an included composition of another MusicComposition (<https://schema.org/includedComposition>). For example, the first movement of a composition could be represented as an includedComposition of a main symphony.

Parameters

- **main_identifier** – The identifier of a main MusicComposition.
- **part_identifier** – The identifier of a MusicComposition which is an included part of the main MusicComposition.

`trompace.mutations.musiccomposition.mutation_merge_music_composition_recorded_as` (*music_composition_id*, *music_recording_id*)

Returns a mutation for adding a MusicRecording as an recording of a MusicComposition (<https://schema.org/recordedAs>).

Parameters

- **music_composition_id** – The identifier of a MusicComposition.
- **music_recording_id** – The identifier of a MusicRecording which is a recording of the MusicComposition.

`trompace.mutations.musiccomposition.mutation_merge_music_composition_work_example` (*music_composition_id*, *creative_work_id*)

Returns a mutation for adding a CreativeWork as an example of a MusicComposition (<https://schema.org/workExample>).

Parameters

- **music_composition_id** – The identifier of a MusicComposition.
- **creativework_id** – The identifier of a CreativeWork which is an example of the MusicComposition.

`trompace.mutations.musiccomposition.mutation_remove_music_composition_composer` (*composition_id*, *person_id*)

Returns a mutation for removing a Person as the composer of a MusicComposition (<https://schema.org/composer>).

Parameters

- **composition_identifier** – The identifier of a MusicComposition.
- **person_identifier** – The identifier of a Person who composed the MusicComposition.

`trompace.mutations.musiccomposition.mutation_remove_music_composition_exact_match` (*from_identifier*, *to_identifier*)

Returns a mutation for removing two MusicComposition representing the same composition (<http://www.w3.org/2004/02/skos/core#exactMatch>).

Parameters

- **from_identifier** – The identifier of one MusicComposition.
- **to_identifier** – The identifier of another MusicComposition.

`trompace.mutations.musiccomposition.mutation_remove_music_composition_has_part` (*main_identifier, part_identifier*)

Returns a mutation for removing a MusicComposition as a part of another MusicComposition (<https://schema.org/hasPart>).

Parameters

- **main_identifier** – The identifier of a main MusicComposition.
- **part_identifier** – The identifier of a MusicComposition which is a part of the main MusicComposition.

`trompace.mutations.musiccomposition.mutation_remove_music_composition_included_composition`

Returns a mutation for removing a MusicComposition as an included composition of another MusicComposition (<https://schema.org/includedComposition>).

Parameters

- **main_identifier** – The identifier of a main MusicComposition.
- **part_identifier** – The identifier of a MusicComposition which is an included part of the main MusicComposition.

`trompace.mutations.musiccomposition.mutation_remove_music_composition_recorded_as` (*music_composition_id, music_recording_id*)

Returns a mutation for removing a MusicRecording as a recording of a MusicComposition. (<https://schema.org/recordedAs>).

Parameters

- **music_composition_id** – The identifier of a MusicComposition.
- **music_recording_id** – The identifier of a MusicRecording which is a recording of the MusicComposition.

`trompace.mutations.musiccomposition.mutation_remove_music_composition_work_example` (*music_composition_id, creative_work_id*)

Returns a mutation for removing a CreativeWork as an example of a MusicComposition (<https://schema.org/workExample>).

Parameters

- **music_composition_id** – The identifier of a MusicComposition.
- **creativework_id** – The identifier of a CreativeWork which is an example of the MusicComposition.


```

trompace.mutations.musiccomposition.mutation_update_music_composition(identifier:
                                                                    str, *,
                                                                    title:
                                                                    str =
                                                                    None,
                                                                    con-
                                                                    trib-
                                                                    utor:
                                                                    str =
                                                                    None,
                                                                    cre-
                                                                    ator:
                                                                    str =
                                                                    None,
                                                                    source:
                                                                    str =
                                                                    None,
                                                                    for-
                                                                    mat_:
                                                                    str =
                                                                    None,
                                                                    sub-
                                                                    ject:
                                                                    str =
                                                                    None,
                                                                    lan-
                                                                    guage:
                                                                    str =
                                                                    None,
                                                                    inlan-
                                                                    guage:
                                                                    str =
                                                                    None,
                                                                    name:
                                                                    str =
                                                                    None,
                                                                    de-
                                                                    scrip-
                                                                    tion:
                                                                    str =
                                                                    None,
                                                                    posi-
                                                                    tion:
                                                                    int =
                                                                    None)

```

Returns a mutation for updating a MusicComposition object.

Parameters

- **identifier** – The identifier of the MusicComposition in the CE to be updated
- **title** – The title of the resource indicated by *source*
- **contributor** – The main URL of the site where the information about this MusicComposition was taken from
- **creator** – The person, organization or service who is creating this MusicComposition

(e.g. URL of the software)

- **source** – The URL of the web resource where information about this MusicComposition is taken from
- **format_** – The mimetype of the resource indicated by *source*
- **subject** – The subject of the music composition.
- **language** – The language the metadata is written in. Currently supported languages are en,es,ca,nl,de,fr
- **inlanguage** – The language of the music composition.
- **name** – The name of the music composition.
- **description** – An account of the music composition.
- **position** – In the case that this is a movement of a larger work (e.g. a Symphony), the position of this MusicComposition in the larger one.

Returns The string for the mutation for updating the music composition.

Raises

- **UnsupportedLanguageException** – if *language* is not one of the supported languages.
- **NotAMimeTypeException** – if *format_* is not a valid mimetype

1.3.9 Person

A `schema:Person` object represents a web resource about a person. For example, this could be a composer or performer.

Creating objects

To create a Person object, use the `CreatePerson` mutation

```
trompace.mutations.person.mutation_create_person(*, title: str, contributor: str, creator: str, source: str, format_: str, language: str = None, name: str = None, family_name: str = None, given_name: str = None, gender: str = None, birth_date: str = None, death_date: str = None, description: str = None, image: str = None, publisher: str = None, honorific_prefix: str = None, honorific_suffix: str = None, job_title: str = None)
```

Returns a mutation for creating a Person

Parameters

- **title** – The title of the resource indicated by *source* (`dcterms:title`)
- **contributor** – The main URL of the site where the information about this Person was taken from (`dcterms:contributor`)
- **creator** – The person, organization or service who is creating this Person (e.g. URL of the software) (`dcterms:creator`)

- **source** – The URL of the web resource where information about this Person is taken from (`dcterms:source`)
- **language** – The language the metadata is written in. Currently supported languages are en,es,ca,nl,de,fr (`dcterms:language`)
- **format_** – The mimetype of the resource indicated by *source* (`dcterms:format`)
- **name** – The name of the person (`schema:name`)
- **family_name** (*optional*) – The family name of the person (`schema:familyName`)
- **given_name** (*optional*) – The given name of the person (`schema:givenName`)
- **gender** (*optional*) – The person's gender (`schema:gender`)
- **birth_date** (*optional*) – The birth date of the person, formatted as yyyy, yyyy-mm or yyyy-mm-dd (`schema:birthDate`)
- **death_date** (*optional*) – The date of death of the person , formatted as yyyy, yyyy-mm or yyyy-mm-dd (`schema:deathDate`)
- **description** (*optional*) – A biographical description of the person (`dc-terms:description`)
- **image** (*optional*) – URL to an image associated with the person (`schema:image`)
- **publisher** (*optional*) – An entity responsible for making the resource available (`dc-terms:publisher`)
- **honorific_prefix** (*optional*) – An honorific prefix (`schema:honorificPrefix`)
- **honorific_suffix** (*optional*) – An honorific suffix (`schema:honorificSuffix`)
- **job_title** (*optional*) – The person's job title. (`schema:jobTitle`)

Returns The string for the mutation for creating the person.

Raises

- **UnsupportedLanguageException** – if language is not one of the supported languages.
- **ValueError** – if gender is not a value supported by the CE
- **NotAMimeTypeException** – if *format_* is not a valid mimetype.

To update a Person object, use the UpdatePerson mutation

```
trompace.mutations.person.mutation_update_person (identifier: str, *, title: str = None,
                                                    contributor: str = None, creator: str
                                                    = None, source: str = None, lan-
                                                    guage: str = None, format_: str
                                                    = None, name: str = None, fam-
                                                    ily_name: str = None, given_name:
                                                    str = None, gender: str = None,
                                                    birth_date: str = None, death_date:
                                                    str = None, description: str = None,
                                                    image: str = None, publisher: str
                                                    = None, honorific_prefix: str =
                                                    None, honorific_suffix: str = None,
                                                    job_title: str = None)
```

Returns a mutation for updating a Person

Parameters

- **identifier** – The identifier of the person in the CE to be updated
- **title** – The title of the resource indicated by *source* (`dcterms:title`)
- **contributor** – The main URL of the site where the information about this Person was taken from (`dcterms:contributor`)
- **creator** – The person, organization or service who is creating this Person (e.g. URL of the software) (`dcterms:creator`)
- **source** – The URL of the web resource where information about this Person is taken from (`dcterms:source`)
- **language** – The language the metadata is written in. Currently supported languages are en,es,ca,nl,de,fr (`dcterms:language`)
- **format_** – The mimetype of the resource indicated by *source* (`dcterms:format`)
- **name** – The name of the person (`schema:name`)
- **family_name** (*optional*) – The family name of the person (`schema:familyName`)
- **given_name** (*optional*) – The given name of the person (`schema:givenName`)
- **gender** (*optional*) – The person's gender (`schema:gender`)
- **birth_date** (*optional*) – The birth date of the person, formatted as yyyy, yyyy-mm or yyyy-mm-dd (`schema:birthDate`)
- **death_date** (*optional*) – The date of death of the person , formatted as yyyy, yyyy-mm or yyyy-mm-dd (`schema:deathDate`)
- **description** (*optional*) – A biographical description of the person (`dcterms:description`)
- **image** (*optional*) – URL to an image associated with the person (`schema:image`)
- **publisher** (*optional*) – An entity responsible for making the resource available (`dcterms:publisher`)
- **honorific_prefix** (*optional*) – An honorific prefix (`schema:honorificPrefix`)
- **honorific_suffix** (*optional*) – An honorific suffix (`schema:honorificSuffix`)
- **job_title** (*optional*) – The person's job title. (`schema:jobTitle`)

Returns The string for the mutation for updating the person.

Raises

- **UnsupportedLanguageException** – if language is not one of the supported languages.
- **ValueError** – if gender is not a value supported by the Ce
- **NotAMimeTypeException** – if format_ is not a valid mimetype.

To delete a Person object, use the DeletePerson mutation

```
trompace.mutations.person.mutation_delete_person (identifier: str)
```

Returns a mutation for deleting a Person with the given identifier.

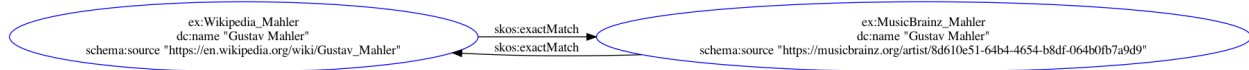
Parameters **identifier** – The identifier of the Person to delete.

Returns A mutation string to delete a Person

Relations

exactMatch

The `skos:exactMatch` relationship is used to identify that two `Person` nodes refer to the exact same natural person.



Use the `MergePersonExactMatch` mutation to create a one-way link between two `Person` objects

```
trompace.mutations.person.mutation_person_add_exact_match_person (identifier_from:
                                                                    str, identifier_to:
                                                                    str)
```

Returns a mutation for linking two `Person` objects with `skos:exactMatch`.

Parameters

- **identifier_from** – the identifier of the `Person` to match to
- **identifier_to** – the identifier of the `Person` that is an exact match of `identifier_from`

Returns: a mutation to make an `exactMatch` relationship between the `Person` objects

Use the `RemovePersonExactMatch` mutation to remove link between two `Person` objects

```
trompace.mutations.person.mutation_person_remove_exact_match_person (identifier_from:
                                                                        str, identifier_to:
                                                                        str)
```

Returns a mutation for removing the `skos:exactMatch` relation between two `Person` objects

Parameters

- **identifier_from** – the identifier of the `Person` to match to
- **identifier_to** – the identifier of the `Person` that is an exact match of `identifier_from`

Returns: a mutation to remove the `exactMatch` relationship from the `Person` objects

1.3.10 Place

```
trompace.mutations.place.mutation_create_place (*, title: str, contributor: str, creator: str,
                                                  source: str, format_: str, name: str =
                                                  None, language: str = None)
```

Returns a mutation for creating a `place` object.

Parameters

- **name** – The name of the `place` object.
- **creator** – The person, organization or service who created the thing the web resource is about.
- **contributor** – A person, an organization, or a service responsible for contributing the `place` to the web resource. This can be either a name or a base URL.
- **format_** – A `MimeType` of the format of the page describing the `place`.
- **source** – The URL of the web resource about this `place`.
- **title** – The title of the resource indicated by `source`

Returns The string for the mutation for creating the place.

`trompace.mutations.place.mutation_delete_place(identifier: str)`

Returns a mutation for deleting a place object based on the identifier.

Parameters `identifier` – The unique identifier of the place object.

Returns The string for the mutation for deleting the place object based on the identifier.

`trompace.mutations.place.mutation_merge_person_birthplace(person_identifier: str, place_identifier: str)`

Returns a mutation for adding a place as the place of birth of a person.

Parameters

- `person_identifier` – The unique identifier of the person object.
- `place_identifier` – The unique identifier of the place where the person was born.

Returns The string for the mutation for merging a place as the birth place of a person.

`trompace.mutations.place.mutation_merge_person_deathplace(person_identifier: str, place_identifier: str)`

Returns a mutation for adding a place as the place of death of a person.

Parameters

- `person_identifier` – The unique identifier of the person.
- `place_identifier` – The unique identifier of the place where the person died.

Returns The string for the mutation for merging a place as the place of death of a person.

`trompace.mutations.place.mutation_remove_person_birthplace(person_identifier: str, place_identifier: str)`

Returns a mutation for removing a place as the place of birth of a person.

Parameters

- `person_identifier` – The unique identifier of the person object.
- `place_identifier` – The unique identifier of the place where the person was born.

Returns The string for the mutation for removing a place as the birth place of a person.

`trompace.mutations.place.mutation_remove_person_deathplace(person_identifier: str, place_identifier: str)`

Returns a mutation for removing a place as the place of death of a person.

Parameters

- `person_identifier` – The unique identifier of the person.
- `place_identifier` – The unique identifier of the place where the person died.

Returns The string for the mutation for removing a place as the place of death of a person.

`trompace.mutations.place.mutation_update_place(identifier: str, *, name: str = None, title: str = None, creator: str = None, contributor: str = None, format_: str = None, source: str = None, language: str = None)`

Returns a mutation for updating a place object.

Parameters

- `identifier` – The identifier of the place in the CE to be updated.

- **name** – The name of the place object.
- **creator** – The person, organization or service who created the thing the web resource is about.
- **contributor** – A person, an organization, or a service responsible for contributing the place to the web resource. This can be either a name or a base URL.
- **format_** – A MimeType of the format of the page describing the place.
- **source** – The URL of the web resource about this place.
- **title** – The title of the resource indicated by *source*

Returns The string for the mutation for updating the place.

1.3.11 Property

```
trompace.mutations.property.mutation_create_property (title: str, name: str, rangeIncludes: List[trompace.StringConstant], description: str = None)
```

Returns a mutation for creating a property :param title: The title of the property :param name: The name of the property :param description: An account of the property :param rangeIncludes: accepts an array of possible node types for this content reference and can be used by the Component developer to limit the type of nodes (content types) that can be selected.

Returns The string for the mutation for creating the property.

```
trompace.mutations.property.mutation_create_propertyvaluespecification (name:
                                                                    str,
                                                                    de-
                                                                    fault-
                                                                    Value:
                                                                    str,
                                                                    val-
                                                                    ue-
                                                                    MaxLength:
                                                                    int,
                                                                    val-
                                                                    ueMinLength:
                                                                    int,
                                                                    mul-
                                                                    ti-
                                                                    ple-
                                                                    Val-
                                                                    ues:
                                                                    bool,
                                                                    val-
                                                                    ue-
                                                                    Name:
                                                                    str,
                                                                    val-
                                                                    uePat-
                                                                    tern:
                                                                    str,
                                                                    val-
                                                                    ueRequired:
                                                                    bool,
                                                                    de-
                                                                    scrip-
                                                                    tion:
                                                                    str
                                                                    =
                                                                    None)
```

Returns a mutation for creating a property value specification Each PropertyValueSpecification defines a scalar input parameter that the Component user should be prompted with when preparing the request for an algorithm process job. There are numerous properties that can be used to set requirements, type and limits for a scalar parameter. With these properties, a Component developer can set up the input field for this parameter. :param name: The name of the property value specification. :param description: An account of the property :param defaultValue: the default value to use for the property value specification. :param valueMaxLength: The maximum length of the value. :param valueMinLength: The minimum length of the value. :param multipleValues: A boolean that states if multiple values are accepted or not. :param valueName: The name of the value. :param valuePattern: The format of the value. :param valueRequired: A boolean stating if the value is required or not.

Returns The string for the mutation for creating the property value specification.

1.3.12 Rating

```
trompace.mutations.rating.create_rating(*, creator: str, bestrating: int, ratingvalue: int =
                                         None, worstrating: int = None, ratingexplanation:
                                         str = None, additionaltype: str = None)
```

Return a mutation for making a Rating. A Rating (<https://schema.org/Rating>) is an evaluation on a numeric scale.

Parameters

- **creator** – a URI to the identity of the user who created this DefinedTerm
- **bestrating** – The highest value allowed in this rating system
- **ratingvalue** – The rating for the content.
- **worstrating** (*optional*) – The lowest value allowed in this rating system. If worstRating is omitted, 1 is assumed.
- **ratingexplanation** (*optional*) – A freeform text box describing why this rating was given
- **additionaltype** (*optional*) – A schema.org additionalType used to categorise this Rating

Returns A GraphQL Mutation to create a Rating in the Trompa CE

```
trompace.mutations.rating.create_rating_definition(*, creator: str, bestrating: int,
                                                    worstrating: int, name: str =
                                                    None)
```

Return a mutation for making a Rating definition. A Rating (<https://schema.org/Rating>) is an evaluation on a numeric scale. A Rating definition describes the structure that a rating can take. It is used so that an annotation tool can give a title and present an input according to the correct scale (e.g. 1-5 or 1-100).

This is a helper method that requires a bestrating and worstrating value and automatically sets additionaltype to RATING_DEFINITION_ADDITIONAL_TYPE

A name is recommended if multiple ratings are going to be shown on the same annotator, but isn't necessary.

```
trompace.mutations.rating.delete_rating(identifier: str)
```

Return a mutation for deleting a Rating.

Parameters **identifier** – The identifier of the Rating to delete

Returns A GraphQL Mutation to delete a Rating from the Trompa CE

```
trompace.mutations.rating.rating_add_was_derived_from_rating(rating_id: str,
                                                             original_rating_id:
                                                             str)
```

Generate a mutation to link two ratings with the prov:wasDerivedFrom relation

Parameters

- **rating_id** – the id of the new rating
- **original_rating_id** – the id of the original rating from which rating_id was derived

Returns A GraphQL Mutation for MergeRatingWasDerivedFrom

```
trompace.mutations.rating.rating_remove_was_derived_from_rating(rating_id:
                                                                str, origi-
                                                                nal_rating_id:
                                                                str)
```

Generate a mutation to remove the prov:wasDerivedFrom relation between two ratings

Parameters

- **rating_id** – the id of the new rating
- **original_rating_id** – the id of the original rating from which rating_id was derived

Returns A GraphQL Mutation for RemoveRatingWasDerivedFrom

1.4 Queries

Queries are used to retrieve information from the CE

1.4.1 Person

```
trompace.queries.person.query_person (identifier: str = None, contributor: str = None, creator:  
                                       str = None, source: str = None, language: str = None,  
                                       format_: str = None, name: str = None, family_name:  
                                       str = None, given_name: str = None, return_items_list:  
                                       list = ['identifier', 'name'])
```

Returns a query for retrieving a person or people.

Parameters

- **identifier** – The identifier of the person in the CE.
- **contributor** – The main URL of the site where the information about the Person was taken from
- **creator** – The person, organization or service who is creating this Person (e.g. URL of the software)
- **source** – The URL of the web resource where information about this Person is taken from
- **language** – The language the metadata is written in.
- **format_** – The mimetype of the resource indicated by *source*
- **name** – The name of the person
- **family_name** – The family name of the person
- **given_name** – The given name of the person
- **return_items_list** – A list of item fields that the query must return.

Returns The string for a person query.

1.4.2 MusicComposition

`trompace.queries.musiccomposition.query_musiccomposition` (*identifier: str = None, title: str = None, contributor: str = None, creator: str = None, source: str = None, inlanguage: str = None, name: str = None, position: int = None, filter_: dict = None, return_items: Union[list, str] = None*)

Returns a query for querying the database for a music composition. :param identifier: The identifier of the music composition in the CE. :param title: The title of the resource indicated by *source* :param creator: The person, organization or service who created the thing the web resource is about. :param contributor: A person, an organization, or a service responsible for contributing the music composition to the web resource. This can be either a name or a base URL. :param source: The source URL that an item comes from :param inlanguage: The language of the music composition. Currently supported languages are en,es,ca,nl,de,fr :param name: The name of the music composition. :param position: In the case that this is a movement of a larger work (e.g. a Symphony), the position of this

MusicComposition in the larger one.

Parameters

- **filter_** – return nodes with this custom filter
- **return_items** – return these items in the response

Returns The string for the quereing the music composition.

Raises `UnsupportedLanguageException` if the input language is not one of the supported languages. –

1.4.3 MediaObject

`trompace.queries.mediaobject.query_mediaobject` (*identifier: str = None, creator: str = None, contributor: str = None, encodingformat: str = None, source: str = None, contenturl: str = None, inlanguage: str = None, filter_: dict = None, return_items: Union[list, str] = None*)

Returns a query for querying the database for a media object. :param identifier: The identifier of the media object in the CE. :param creator: The person, organization or service who created the thing the web resource is about. :param contributor: A person, an organization, or a service responsible for contributing the media object to the web resource. This can be either a name or a base URL. :param encodingformat: A MimeType of the format of object encoded by the media object. :param source: The URL of the web resource to be represented by the node. :param contenturl: The URL of the content encoded by the media object. :param inlanguage: The language of the media object. Currently supported languages are en,es,ca,nl,de,fr. :param filter_: return nodes with this custom filter :param return_items: return these items in the response

Returns The string for the quereing the media object.

Raises `UnsupportedLanguageException` if the input language is not one of the supported languages. –

1.5 Algorithms

The algorithms module allows for the realtime interaction between external algorithms and the CE.

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

PYTHON MODULE INDEX

t

- `trompace.mutations.audioobject`, [7](#)
- `trompace.mutations.controlaction`, [10](#)
- `trompace.mutations.definedterm`, [12](#)
- `trompace.mutations.digitaldocument`, [13](#)
- `trompace.mutations.entrypoint`, [15](#)
- `trompace.mutations.itemlist`, [16](#)
- `trompace.mutations.mediaobject`, [20](#)
- `trompace.mutations.musiccomposition`, [25](#)
- `trompace.mutations.place`, [33](#)
- `trompace.mutations.property`, [35](#)
- `trompace.mutations.rating`, [37](#)
- `trompace.queries.mediaobject`, [39](#)
- `trompace.queries.musiccomposition`, [39](#)
- `trompace.queries.person`, [38](#)

INDEX

C

create_defined_term() (in module *trompace.mutations.definedterm*), 12
 create_defined_term_set() (in module *trompace.mutations.definedterm*), 12
 create_rating() (in module *trompace.mutations.rating*), 37
 create_rating_definition() (in module *trompace.mutations.rating*), 37

D

defined_term_add_to_defined_term_set() (in module *trompace.mutations.definedterm*), 12
 delete_defined_term() (in module *trompace.mutations.definedterm*), 13
 delete_defined_term_set() (in module *trompace.mutations.definedterm*), 13
 delete_rating() (in module *trompace.mutations.rating*), 37

M

module

trompace.mutations.audioobject, 7
trompace.mutations.controlaction, 10
trompace.mutations.definedterm, 12
trompace.mutations.digitaldocument, 13
trompace.mutations.entrypoint, 15
trompace.mutations.itemlist, 16
trompace.mutations.mediaobject, 20
trompace.mutations.musiccomposition, 25
trompace.mutations.place, 33
trompace.mutations.property, 35
trompace.mutations.rating, 37
trompace.queries.mediaobject, 39
trompace.queries.musiccomposition, 39
trompace.queries.person, 38
 mutation_add_actioninterface_result() (in module *trompace.mutations.controlaction*),

10
 mutation_add_controlaction_additionalproperty() (in module *trompace.mutations.controlaction*), 10
 mutation_add_controlaction_object() (in module *trompace.mutations.controlaction*), 10
 mutation_add_entrypoint_controlaction() (in module *trompace.mutations.controlaction*), 10
 mutation_add_itemlist_itemlist_element() (in module *trompace.mutations.itemlist*), 16
 mutation_add_listitem_item() (in module *trompace.mutations.itemlist*), 16
 mutation_add_listitem_nextitem() (in module *trompace.mutations.itemlist*), 16
 mutation_add_media_object_used() (in module *trompace.mutations.mediaobject*), 20
 mutation_add_propertyvaluespecification_potentiala (in module *trompace.mutations.controlaction*), 10
 mutation_create_audioobject() (in module *trompace.mutations.audioobject*), 7
 mutation_create_controlaction() (in module *trompace.mutations.controlaction*), 11
 mutation_create_digitaldocument() (in module *trompace.mutations.digitaldocument*), 13
 mutation_create_entry_point() (in module *trompace.mutations.entrypoint*), 15
 mutation_create_itemlist() (in module *trompace.mutations.itemlist*), 16
 mutation_create_listitem() (in module *trompace.mutations.itemlist*), 17
 mutation_create_media_object() (in module *trompace.mutations.mediaobject*), 20
 mutation_create_music_composition() (in module *trompace.mutations.musiccomposition*), 25
 mutation_create_person() (in module *trompace.mutations.person*), 30
 mutation_create_place() (in module *trompace.mutations.place*), 33

mutation_create_property()	(in module trompace.mutations.property), 35	mutation_merge_person_deathplace()	(in module trompace.mutations.place), 34
mutation_create_propertyvaluespecification()	(in module trompace.mutations.property), 35	mutation_modify_controlaction()	(in module trompace.mutations.controlaction), 11
mutation_delete_audioobject()	(in module trompace.mutations.audioobject), 8	mutation_person_add_exact_match_person()	(in module trompace.mutations.person), 33
mutation_delete_digitaldocument()	(in module trompace.mutations.digitaldocument), 14	mutation_person_remove_exact_match_person()	(in module trompace.mutations.person), 33
mutation_delete_itemlist()	(in module trompace.mutations.itemlist), 17	mutation_remove_audioobject_encoding()	(in module trompace.mutations.audioobject), 8
mutation_delete_listitem()	(in module trompace.mutations.itemlist), 17	mutation_remove_audioobject_exampleofwork()	(in module trompace.mutations.audioobject), 8
mutation_delete_media_object()	(in module trompace.mutations.mediaobject), 21	mutation_remove_itemlist_itemlist_element()	(in module trompace.mutations.itemlist), 17
mutation_delete_music_composition()	(in module trompace.mutations.musiccomposition), 26	mutation_remove_listitem_item()	(in module trompace.mutations.itemlist), 18
mutation_delete_person()	(in module trompace.mutations.person), 32	mutation_remove_listitem_nextitem()	(in module trompace.mutations.itemlist), 18
mutation_delete_place()	(in module trompace.mutations.place), 34	mutation_remove_media_object_encoding()	(in module trompace.mutations.mediaobject), 22
mutation_merge_audioobject_encoding()	(in module trompace.mutations.audioobject), 8	mutation_remove_media_object_wasderivedfrom()	(in module trompace.mutations.mediaobject), 22
mutation_merge_audioobject_exampleofwork()	(in module trompace.mutations.audioobject), 8	mutation_remove_mediaobject_example_of_work()	(in module trompace.mutations.mediaobject), 23
mutation_merge_media_object_encoding()	(in module trompace.mutations.mediaobject), 21	mutation_remove_music_composition_composer()	(in module trompace.mutations.musiccomposition), 27
mutation_merge_media_object_wasderivedfrom()	(in module trompace.mutations.mediaobject), 21	mutation_remove_music_composition_exact_match()	(in module trompace.mutations.musiccomposition), 27
mutation_merge_mediaobject_example_of_work()	(in module trompace.mutations.mediaobject), 22	mutation_remove_music_composition_has_part()	(in module trompace.mutations.musiccomposition), 28
mutation_merge_music_composition_composer()	(in module trompace.mutations.musiccomposition), 26	mutation_remove_music_composition_included_composition()	(in module trompace.mutations.musiccomposition), 28
mutation_merge_music_composition_exact_match()	(in module trompace.mutations.musiccomposition), 26	mutation_remove_music_composition_recorded_as()	(in module trompace.mutations.musiccomposition), 28
mutation_merge_music_composition_has_part()	(in module trompace.mutations.musiccomposition), 26	mutation_remove_music_composition_work_example()	(in module trompace.mutations.musiccomposition), 28
mutation_merge_music_composition_included_composition()	(in module trompace.mutations.musiccomposition), 26	mutation_remove_person_birthplace()	(in module trompace.mutations.place), 34
mutation_merge_music_composition_recorded_as()	(in module trompace.mutations.musiccomposition), 27	mutation_remove_person_deathplace()	(in module trompace.mutations.place), 34
mutation_merge_music_composition_work_example()	(in module trompace.mutations.musiccomposition), 27	mutation_request_controlaction()	(in module trompace.mutations.controlaction), 11
mutation_merge_person_birthplace()	(in module trompace.mutations.place), 34	mutation_sequence_add_itemlist_itemlist_element()	(in module trompace.mutations.itemlist), 18

`mutation_sequence_add_listitem_item()` (in module *trompace.mutations.itemlist*), 18
`mutation_sequence_add_listitem_nextitem()` (in module *trompace.mutations.itemlist*), 18
`mutation_sequence_create_listitem()` (in module *trompace.mutations.itemlist*), 19
`mutation_update_audioobject()` (in module *trompace.mutations.audioobject*), 9
`mutation_update_controlaction_status()` (in module *trompace.mutations.controlaction*), 11
`mutation_update_digitaldocument()` (in module *trompace.mutations.digitaldocument*), 14
`mutation_update_itemlist()` (in module *trompace.mutations.itemlist*), 19
`mutation_update_listitem()` (in module *trompace.mutations.itemlist*), 19
`mutation_update_media_object()` (in module *trompace.mutations.mediaobject*), 23
`mutation_update_music_composition()` (in module *trompace.mutations.musiccomposition*), 28
`mutation_update_person()` (in module *trompace.mutations.person*), 31
`mutation_update_place()` (in module *trompace.mutations.place*), 34
`module`, 13
`trompace.mutations.entrypoint` module, 15
`trompace.mutations.itemlist` module, 16
`trompace.mutations.mediaobject` module, 20
`trompace.mutations.musiccomposition` module, 25
`trompace.mutations.place` module, 33
`trompace.mutations.property` module, 35
`trompace.mutations.rating` module, 37
`trompace.queries.mediaobject` module, 39
`trompace.queries.musiccomposition` module, 39
`trompace.queries.person` module, 38

U

`update_defined_term()` (in module *trompace.mutations.definedterm*), 13
`update_defined_term_set()` (in module *trompace.mutations.definedterm*), 13

Q

`query_mediaobject()` (in module *trompace.queries.mediaobject*), 39
`query_musiccomposition()` (in module *trompace.queries.musiccomposition*), 39
`query_person()` (in module *trompace.queries.person*), 38

R

`rating_add_was_derived_from_rating()` (in module *trompace.mutations.rating*), 37
`rating_remove_was_derived_from_rating()` (in module *trompace.mutations.rating*), 37

S

`submit_query()` (in module *trompace.connection*), 4

T

`trompace.mutations.audioobject` module, 7
`trompace.mutations.controlaction` module, 10
`trompace.mutations.definedterm` module, 12
`trompace.mutations.digitaldocument`