
Craft AI MLOps platform Python SDK

Release 0.18.0rc1

Craft AI

May 31, 2023

Contents

1	craft_ai_sdk package	1
1.1	craft_ai_sdk.exceptions module	1
1.2	craft_ai_sdk.sdk module	1
2	CraftAiSdk class	3
2.1	Step	4
2.2	Pipeline	8
2.3	Pipeline Execution	9
2.4	Deployment	11
2.5	Endpoint	13
2.6	Data store	15
2.7	Environment Variables	16
2.8	Pipeline Metrics	17
	Python Module Index	19
	Index	21

craft_ai_sdk package

1.1 craft_ai_sdk.exceptions module

exception `craft_ai_sdk.exceptions.SdkException`(*message, status_code=None, name=None, stack_message=None, request_id=None, additional_data=None*)

Bases: `Exception`

General exception while using this package

1.2 craft_ai_sdk.sdk module

See section *CraftAiSdk class* for more details.

CraftAiSdk class

```
class craft_ai_sdk.CraftAiSdk(sdk_token=None, environment_url=None, control_url=None,  
                             verbose_log=None, warn_on_metric_outside_of_step=True)
```

Main class to instantiate

base_environment_url

Base URL to CraftAI Environment.

Type

str

base_environment_api_url

Base URL to CraftAI Environment API.

Type

str

base_control_url

Base URL to CraftAI authorization server.

Type

str

base_control_api_url

Base URL to CraftAI authorization server API.

Type

str

verbose_log

If True, information during method execution will be printed.

Type

bool

warn_on_metric_outside_of_step

If True, a warning will be printed when a metric is added outside of a step.

Type

bool

```
__init__(sdk_token=None, environment_url=None, control_url=None, verbose_log=None,  
         warn_on_metric_outside_of_step=True)
```

Inits CraftAiSdk.

Parameters

- **sdk_token** (str, optional) – SDK token. You can retrieve it from the website. Defaults to CRAFT_AI_SDK_TOKEN environment variable.
- **environment_url** (str, optional) – URL to CraftAI environment. Defaults to CRAFT_AI_ENVIRONMENT_URL environment variable.

- **control_url** (str, optional) – URL to CraftAI authorization server. You probably don't need to set it. Defaults to CRAFT_AI_CONTROL_URL environment variable, or <https://mlops-platform.craft.ai>.
- **verbose_log** (bool, optional) – If True, information during method execution will be printed. Defaults to True if the environment variable SDK_VERBOSE_LOG is set to true; False if it is set to false; else, defaults to True in interactive mode; False otherwise.
- **warn_on_metric_outside_of_step** (bool, optional) – If True, a warning will be raised when a metric is added outside of a step. Defaults to True.

Raises

- **ValueError** – if the sdk_token or environment_url is not defined and
- **the corresponding environment variable is not set.** –

2.1 Step

```
CraftAiSdk.create_step(step_name, function_path=None, function_name=None,  
                        repository_branch=STEP_PARAMETER.FALLBACK_PROJECT,  
                        description=None, container_config=None, inputs=None,  
                        outputs=None)
```

Create pipeline step from a function located on a remote repository.

Use STEP_PARAMETER to explicitly set a value to null or fall back on project information.

Parameters

- **step_name** (str) – Step name.
- **function_path** (str, optional) – Path to the file that contains the function. This parameter is required if parameter “dockerfile_path” is not specified.
- **function_name** (str, optional) – Name of the function in that file. This parameter is required if parameter “dockerfile_path” is not specified.
- **repository_branch** (str, optional) – Branch name. Defaults to falling back on project information.
- **description** (str, optional) – Description. Defaults to None.
- **container_config** (dict[str, str], optional) – Some step configuration, with the following optional keys:
 - “language” (str): Language and version used for the step. Defaults to falling back on project information.
 - “repository_url” (str): Remote repository url. Defaults to falling back on project information.
 - “repository_deploy_key” (str): Private SSH key of the repository. Defaults to falling back on project information, can be set to null.
 - “requirements_path” (str): Path to the requirements.txt file. Environment variables created through [create_or_update_environment_variable\(\)](#) can be used in requirements.txt, as in “\${ENV_VAR}”. Defaults to falling back on project information, can be set to null.
 - “included_folders” (list[str]): List of folders and files in the repository required for the step execution. Defaults to falling back on project information, can be set to null.

- "system_dependencies" (list[str]): List of system dependencies. Defaults to falling back on project information, can be set to null.
- "dockerfile_path" (str): Path to the Dockerfile. This parameter should only be used as a last resort and for advanced use. When specified, the following parameters should be set to null: "function_path", "function_name", "language", "requirements_path" and "system_dependencies".
- **inputs** (list of instances of *Input*) – List of inputs. Each parameter of the step function should be specified as an instance of *Input* via this parameter *inputs*. During the execution of the step, the value of the inputs would be passed as function arguments.
- **outputs** (list of instances of *Output*) – List of the step outputs. For the step to have outputs, the function should return a dict with the name of the output as keys and the value of the output as values. Each output should be specified as an instance of *Output* via this parameter *outputs*.

Returns

List of steps represented as dict (with either keys "id" and "name" if the creation succeeded or keys "name" and "error" if the creation failed).

Return type

list of dict[str, str]

CraftAiSdk.get_step(step_name)

Get a single step if it exists.

Parameters

step_name (str) – The name of the step to get.

Returns

None if the step does not exist; otherwise the step information, with the following keys:

- "name" (str): Name of the step.
- "status" (str): either "Pending" or "Ready".
- "created_at" (str): The creation date in ISO format.
- "updated_at" (str): The last update date in ISO format.
- "inputs" (dict): The inputs of the step.
- "outputs" (dict): The outputs of the step.

Return type

dict

CraftAiSdk.update_step(step_name, function_path=None, function_name=None, repository_branch=STEP_PARAMETER.FALLBACK_PROJECT, description=None, container_config=None)

Update a pipeline step from a source code located on a remote repository.

The current step configuration will be **replaced** by the provided options. Use STEP_PARAMETER to explicitly set a value to null or fall back on project information.

Warning: This feature is experimental and may behave unexpectedly. It is your responsibility to check the state and behavior of the step after an update. If the creation of the new step configuration takes more than roughly a minute, this function will return (time out) before the update terminated and you will need to check the step status manually at regular intervals. If several step updates are applies

at the same time, in some rare cases the code used for the step may not match its configuration.

Parameters

- **step_name** (*str*) – Name of the step to update.
- **function_path** (*str*, optional) – Path to the file that contains the function. This parameter is required if parameter “dockerfile_path” is not specified.
- **function_name** (*str*, optional) – Name of the function in that file. This parameter is required if parameter “dockerfile_path” is not specified.
- **repository_branch** (*str*, optional) – Branch name. Defaults to falling back on project information.
- **description** (*str*, optional) – Description. Defaults to None.
- **container_config** (*dict*[*str*, *str*], optional) – Some step configuration, with the following optional keys:
 - “language” (*str*): Language and version used for the step. Defaults to falling back on project information.
 - “repository_url” (*str*): Remote repository url. Defaults to falling back on project information.
 - “repository_deploy_key” (*str*): Private SSH key of the repository. Defaults to falling back on project information, can be set to null.
 - “requirements_path” (*str*): Path to the requirements.txt file. Environment variables created through `create_or_update_environment_variable()` can be used in requirements.txt, as in “\${ENV_VAR}”. Defaults to falling back on project information, can be set to null.
 - “included_folders” (*list*[*str*]): List of folders and files in the repository required for the step execution. Defaults to falling back on project information, can be set to null.
 - “system_dependencies” (*list*[*str*]): List of system dependencies. Defaults to falling back on project information, can be set to null.
 - “dockerfile_path” (*str*): Path to the Dockerfile. This parameter should only be used as a last resort and for advanced use. When specified, the following parameters should be set to null: “function_path”, “function_name”, “language”, “requirements_path” and “system_dependencies”.

Returns

Information of the updated step represented as dict (with either keys “id” and “name” if the creation succeeded or keys “name” and “error” if the creation failed).

Return type

dict[*str*, *str*]

`CraftAiSdk.list_steps()`

Get the list of all steps.

Returns

List of steps represented as dict (with keys “name”, “status”, “created_at”, “updated_at”).

Return type

list of dict

`CraftAiSdk.delete_step(step_name, force_dependents_deletion=False)`

Delete one step.

Parameters

- **step_name** (*str*) – Name of the step to delete as defined in the config. yaml configuration file.
- **force_dependents_deletion** (bool, optional) – if True the associated step's dependencies will be deleted too (pipeline, pipeline executions, deployments). Defaults to False.

Returns

Deleted step represented as dict (with keys "id" and "name").

Return type

dict[str, str]

```
class craft_ai_sdk.Input(name, data_type, description=None, is_required=None,
                        default_value=None)
```

Class to specify a step input when creating a step (cf. `CraftAiSdk.create_step()`).**Parameters**

- **name** (*str*) – Name of the input. This corresponds to the name of a parameter of a step function.
- **data_type** (*str*) – Type of the input: It could be one of "string", "number", "boolean", "json", "array" or "file". For convenience, members of the enumeration `INPUT_OUTPUT_TYPES` could be used too.
- **description** (*str*, optional) – Description. Defaults to None.
- **is_required** (bool, optional) – Specify if an value should be provided at execution time. Defaults to None.
- **default_value** (Any, optional) – A default value for the step input at execution time. The type for *default_value* should match the type specified by *data_type*. Defaults to None.

```
class craft_ai_sdk.Output(name, data_type, description=None)
```

Class to specify a step output when creating a step (cf. `CraftAiSdk.create_step()`).**Parameters**

- **name** (*str*) – Name of the output. This corresponds to the key of the *dict* returned by the step function.
- **data_type** (Any) – Type of the output. It could be one of "string", "number", "boolean", "json", "array" or "file". For convenience, members of the enumeration `INPUT_OUTPUT_TYPES` could be used too.
- **description** (*str*, optional) – Description. Defaults to None.

```
class craft_ai_sdk.INPUT_OUTPUT_TYPES(strenum.LowercaseStrEnum)
```

Enumeration for Input and Output data types.

`ARRAY = 'array'``BOOLEAN = 'boolean'``FILE = 'file'``JSON = 'json'`

```
NUMBER = 'number'

STRING = 'string'
```

2.2 Pipeline

`CraftAiSdk.create_pipeline(pipeline_name, step_name)`

Create a pipeline containing a single step.

Parameters

- **pipeline_name** (*str*) – Name of the pipeline to be created.
- **step_name** (*str*) – Name of the step to be included in the pipeline. Note that the step should have the status "Ready" to create the pipeline.

Returns

Created pipeline represented as dict (with key "id").

Return type

dict[str, str]

`CraftAiSdk.get_pipeline(pipeline_name)`

Get a single pipeline if it exists.

Parameters

pipeline_name (*str*) – Name of the pipeline to get.

Returns

The pipeline information or None if the pipeline does not exist.

- "id" (*str*): Pipeline id.
- "name" (*str*): Pipeline name.
- "created_at" (*str*): Pipeline date of creation.
- "steps" (*list*): List of step names.

Return type

dict

`CraftAiSdk.delete_pipeline(pipeline_name, force_deployments_deletion=False)`

Delete a pipeline identified by its name and id.

Parameters

- **pipeline_name** (*str*) – Name of the pipeline.
- **force_deployments_deletion** (*bool*, optional) – if True the associated endpoints will be deleted too. Defaults to False.

Returns

The deleted pipeline and its associated deleted endpoints. The returned dict contains two keys:

- "pipeline" (*dict*): Deleted pipeline represented as dict (with keys "id" and "name").
- "endpoints" (*list*): List of deleted endpoints represented as dict (with keys "id" and "name").

Return type

dict

`CraftAiSdk.list_pipelines()`

Get the list of all pipelines.

Returns

List of pipelines represented as dict (with keys "pipeline_name", "created_at").

Return type

list of dict

2.3 Pipeline Execution

`CraftAiSdk.execute_pipeline(pipeline_name)`

Execute a pipeline.

Warning: Deprecated since version 0.8.1: This function is deprecated and its usage is not supported. Prefer creating an endpoint and triggering your pipeline with it.

Parameters

pipeline_name (*str*) – Name of an existing pipeline.

Returns

Created pipeline execution represented as dict (with key "execution_id").

Return type

dict[str, str]

`CraftAiSdk.run_pipeline(pipeline_name, inputs=None, inputs_mapping=None)`

Run a pipeline.

Parameters

- **pipeline_name** (*str*) – Name of an existing pipeline.
- **inputs** (*dict, optional*) – Dictionary of inputs to pass to the pipeline with input names as keys and corresponding values as values. For files, the value should be the path to the file or a file content in an instance of `io.IOBBase`. Defaults to None.
- **inputs_mapping** (*list of instances of [InputSource](#)*) – List of input mappings, to map pipeline inputs to different sources (such as environment variables). See [InputSource](#) for more details.

Returns

Created pipeline execution represented as dict with output_names as keys and corresponding values as values.

Return type

dict[str, str]

`CraftAiSdk.list_pipeline_executions(pipeline_name)`

Get a list of executions for the given pipeline

Parameters

pipeline_name (*str*) – Name of an existing pipeline.

Returns

A list of information on the pipeline execution represented as dict (with keys "execution_id", "status", "created_at", "steps", "pipeline_name"). In particular the keys:

- "steps" is a list of the execution steps represented as dict (with keys "name", "status").
- "pipeline_name" is the executed pipeline name.

Return type

list

`CraftAiSdk.get_pipeline_execution(pipeline_name, execution_id)`

Get the status of one pipeline execution identified by its name.

Parameters

- **pipeline_name** (*str*) – Name of an existing pipeline.
- **execution_id** (*str*) – Name of the pipeline execution.

Returns

Information on the pipeline execution with id `execution_id` represented as dict (with keys "execution_id", "status", "created_at", "steps", "pipeline_name", "created_by"). In particular the keys:

- "steps" is a list of the execution steps represented as dict (with keys "name", "status").
- "pipeline_name" is the executed pipeline name.

Return type

dict

`CraftAiSdk.get_pipeline_execution_logs(pipeline_name, execution_id, from_datetime=None, to_datetime=None, limit=None)`

Get the logs of an executed pipeline identified by its name.

Parameters

- **pipeline_name** (*str*) – Name of an existing pipeline.
- **execution_id** (*str*) – ID of the pipeline execution.
- **from_datetime** (`datetime.time`, optional) – Datetime from which the logs are collected.
- **to_datetime** (`datetime.time`, optional) – Datetime until which the logs are collected.
- **limit** (`int`, optional) – Maximum number of logs that are collected.

Returns

List of collected logs represented as dict (with keys "message", "timestamp" and "stream").

Return type

list

2.4 Deployment

`CraftAiSdk.create_deployment(pipeline_name, deployment_name, execution_rule, schedule=None, inputs_mapping=None, outputs_mapping=None)`

Create a custom deployment associated to a given pipeline.

Parameters

- **pipeline_name** (*str*) – Name of the pipeline.
- **deployment_name** (*str*) – Name of the deployment.
- **execution_rule** (*str*) – Execution rule of the deployment. Must be “endpoint” or “periodic”. For convenience, members of the enumeration `DEPLOYMENT_EXECUTION_RULES` could be used too.
- **schedule** (*str*, optional) – Schedule of the deployment. Only required if `execution_rule` is “periodic”.
- **inputs_mapping** (*list* of instances of `InputSource`) – List of input mappings, to map pipeline inputs to different sources (such as constant values, endpoint inputs, or environment variables). See `InputSource` for more details. For endpoint rules, if an input of the step in the pipeline is not explicitly mapped, it will be automatically mapped to an endpoint input with the same name. For periodic rules, all inputs of the step in the pipeline must be explicitly mapped.
- **outputs_mapping** (*list* of instances of `OutputDestination`) – List of output mappings, to map pipeline outputs to different destinations. See `OutputDestination` for more details. Each output of the step should be explicitly mapped.

Returns

Created endpoint represented as dict (with key "id" and "name").

Return type

dict[str, str]

`CraftAiSdk.list_deployments()`

Get the list of all deployments.

Returns

List of deployments represented as dict (with keys "id", "name" and "pipeline").

Return type

list of dict

`CraftAiSdk.get_deployment(deployment_name)`

Get information of a deployment.

Parameters

- **deployment_name** (*str*) – Name of the deployment.

Returns

Deployment information represented as dict (with keys "id", "name" and "pipeline").

Return type

dict

`CraftAiSdk.delete_deployment(deployment_name)`

Delete a deployment identified by its name.

Parameters

deployment_name (*str*) – Name of the deployment.

Returns

Deleted deployment represented as dict (with keys "id", "name").

Return type

dict

```
class craft_ai_sdk.InputSource(step_input_name, endpoint_input_name=None,
                               environment_variable_name=None, is_required=None,
                               default_value=None, constant_value=None,
                               is_null=None, datastore_path=None)
```

Class to specify to which source a step input should be mapped when creating a deployment (cf. [*CraftAiSdk.create_deployment\(\)*](#)). The different sources can be one of:

- **endpoint_input_name** (*str*): An endpoint input with the provided name.
- **constant_value**: A constant value.
- **environment_variable_name**: The value of the provided environment variable.
- **is_null**: Nothing.

If the execution rule of the deployment is endpoint and the input is directly mapped to an endpoint input, two more parameters can be specified:

- **default_value**
- **is_required**

Parameters

- **step_input_name** (*str*) – Name of the step input to be mapped.
- **endpoint_input_name** (*str*, optional) – Name of the endpoint input to which the input is mapped.
- **environment_variable_name** (*str*, optional) – Name of the environment variable to which the input is mapped.
- **constant_value** (*Any*, optional) – A constant value.
- **is_null** (*True*, optional) – If specified, the input is not provided any value at execution time.
- **datastore_path** (*str*, optional) – Path of the input file in the datastore.
- **default_value** (*Any*, optional) – This parameter could only be specified if the parameter *endpoint_input_name* is specified.
- **is_required** (*bool*, optional) – This parameter could only be specified if the parameter *endpoint_input_name* is specified. If set to *True*, the corresponding endpoint input should be provided at execution time.

```
class craft_ai_sdk.OutputDestination(step_output_name, endpoint_output_name=None,
                                     is_null=None, datastore_path=None)
```

Class to specify to which destination a step output should be mapped when creating a deployment (cf. [*CraftAiSdk.create_deployment\(\)*](#)). If the execution rule of the deployment is endpoint, an output could either be exposed as an output of the endpoint (via *endpoint_output_name* parameter) or not (via *is_null* parameter).

Parameters

- **step_output_name** (*str*) – Name of the step output to be mapped.

- **endpoint_output_name** (*str*, optional) – Name of the endpoint output to which the output is mapped.
- **is_null** (*True*, optional) – If specified, the output is not exposed as a deployment output.
- **datastore_path** (*str*, optional) – Path of the output file in the datastore.

class craft_ai_sdk.DEPLOYMENT_EXECUTION_RULES(*value*)

Enumeration for deployments execution rules.

2.5 Endpoint

CraftAiSdk.create_endpoint(*pipeline_name*, *endpoint_name*, *inputs_mapping=None*, *outputs_mapping=None*)

Create a custom endpoint associated to a given pipeline.

Warning: Deprecated since version 0.8.1: This function is deprecated and its usage is not supported. Prefer using `create_deployment` instead.

Parameters

- **pipeline_name** (*str*) – Name of the pipeline.
- **endpoint_name** (*str*) – Name of the endpoint.
- **inputs_mapping** (*list* of instances of *InputSource*) – List of input mappings, to map pipeline inputs to different sources (such as constant values, endpoint inputs, or environment variables). See *InputSource* for more details. If an input of the step in the pipeline is not explicitly mapped, it will be automatically mapped to an endpoint input with the same name.
- **outputs_mapping** (*list* of instances of *OutputDestination*) – List of output mappings, to map pipeline outputs to different destinations (either exposed by the endpoint, or not). See *OutputDestination* for more details. Each output of the step should be explicitly mapped.

Returns

Created endpoint represented as dict (with key "id" and "name").

Return type

dict[str, str]

CraftAiSdk.list_endpoints()

Get the list of all endpoints.

Warning: Deprecated since version 0.12.0: This function is deprecated and its usage is not supported. Prefer using `list_deployments` instead.

Returns

List of endpoints represented as dict (with keys "id", "name" and "pipeline").

Return type

list of dict

CraftAiSdk.**get_endpoint**(*endpoint_name*)

Get information of an endpoint.

Warning: Deprecated since version 0.12.0: This function is deprecated and its usage is not supported. Prefer using `get_deployment` instead.

Parameters

endpoint_name (*str*) – Name of the endpoint.

Returns

Endpoint information represented as dict (with keys "id", "name" and "pipeline").

Return type

dict

CraftAiSdk.**trigger_endpoint**(*endpoint_name*, *endpoint_token*, *inputs*={},
wait_for_results=True)

Trigger an endpoint.

Parameters

- **endpoint_name** (*str*) – Name of the endpoint.
- **endpoint_token** (*str*) – Token to access endpoint.
- **inputs** (dict, optional) – Dictionary of inputs to pass to the endpoint with input names as keys and corresponding values as values. For files, the value should be an instance of `io.IOBase`. Defaults to {}.
- **wait_for_results** (bool, optional) – Automatically call `retrieve_endpoint_results` and returns the execution result. Defaults to `True`.

Returns

Created pipeline execution represented as dict (with key "execution_id" if "wait_for_results" is `False`, otherwise with key "status", "execution" & "results" or "error" depending on "status" if "wait_for_results" is `True` or unspecified).

Return type

dict[str, str]

CraftAiSdk.**delete_endpoint**(*endpoint_name*)

Delete an endpoint identified by its name.

Warning: Deprecated since version 0.12.0: This function is deprecated and its usage is not supported. Prefer using `delete_deployment` instead.

Parameters

endpoint_name (*str*) – Name of the endpoint.

Returns

Deleted endpoint represented as dict (with keys "id", "name").

Return type

dict

`CraftAiSdk.generate_new_endpoint_token(endpoint_name)`

Generate a new endpoint token for an endpoint.

Parameters

endpoint_name (*str*) – Name of the endpoint.

Returns

New endpoint token represented as dict (with keys "endpoint_token").

Return type

dict

2.6 Data store

`CraftAiSdk.upload_data_store_object(filepath_or_buffer, object_path_in_datastore)`

Upload a file as an object into the data store.

Parameters

- **filepath_or_buffer** (*str*, or file-like object) – String, path to the file to be uploaded ; or file-like object implementing a `read()` method (e.g. via builtin open function). The file object must be opened in binary mode, not text mode.
- **object_path_in_datastore** (*str*) – Destination of the uploaded file.

`CraftAiSdk.download_data_store_object(object_path_in_datastore, filepath_or_buffer)`

Download an object in the data store and save it into a file.

Parameters

- **object_path_in_datastore** (*str*) – Location of the object to download from the data store.
- **filepath_or_buffer** (*str* or file-like object) – String, filepath to save the file to ; or a file-like object implementing a `write()` method, (e.g. via builtin open function). The file object must be opened in binary mode, not text mode.

Returns

content of the file

Return type

str

`CraftAiSdk.get_data_store_object_information(object_path_in_datastore)`

Get information about a single object in the data store.

Parameters

object_path_in_datastore (*str*) – Location of the object in the data store.

Returns

Object information, with the following keys:

- "path" (*str*): Location of the object in the data store.
- "last_modified" (*str*): The creation date or last modification date in ISO format.
- "size" (*int*): The size of the object in bytes.

Return type

dict

`CraftAiSdk.list_data_store_objects()`

Get the list of the objects stored in the data store.

Returns

List of objects in the data store represented as dict (with keys "path", "last_modified", and "size").

Return type

list of dict

`CraftAiSdk.delete_data_store_object(object_path_in_datastore)`

Delete an object on the datastore.

Parameters

object_path_in_datastore (*str*) – Location of the object to be deleted in the data store.

Returns

Deleted object represented as dict (with key "path").

Return type

dict

2.7 Environment Variables

`CraftAiSdk.create_or_update_environment_variable(environment_variable_name, environment_variable_value)`

Create or update an environment variable available for all pipelines executions.

Parameters

- **environment_variable_name** (*str*) – Name of the environment variable to create.
- **environment_variable_value** (*str*) – Value of the environment variable to create.

Returns

An object containing the id of environment variable (with keys "id")

Return type

dict

`CraftAiSdk.list_environment_variables()`

Get a list of all environments variables.

Returns

List of environment variable as dict (with keys "name" and "value")

Return type

list of dict

`CraftAiSdk.delete_environment_variable(environment_variable_name)`

Delete the specified environment variable

Parameters

environment_variable_name (*str*) – Name of the environment variable to delete.

Returns

dict (with keys "name" and "value") of the deleted environment variable

2.8 Pipeline Metrics

`CraftAiSdk.record_metric_value(label, value)`

Create or update a pipeline metric. Note that this function can only be used inside a step code.

Parameters

- **label** (*str*) – Label of the metric to store.
- **value** (*float*) – Value of the metric to store.

`CraftAiSdk.get_metrics(label=None, pipeline_name=None, deployment_name=None, execution_id=None)`

Get a list of pipeline metrics. Note that only one of the parameters (`pipeline_name`, `deployment_name`, `execution_id`) can be set.

Parameters

- **label** (*str, optional*) – Label of the metric to retrieve.
- **pipeline_name** (*str, optional*) – Filter metrics by pipeline, defaults to all the pipelines.
- **deployment_name** (*str, optional*) – Filter metrics by deployment, defaults to all the deployments.
- **execution_id** (*str, optional*) – Filter metrics by execution, defaults to all the executions.

Returns

List of execution metrics as dict (with keys "label", "value", "created_at", "execution_id", "deployment_name", "pipeline_name").

Return type

list of dict

Python Module Index

C

`craft_ai_sdk.exceptions`, [1](#)
`craft_ai_sdk.sdk`, [1](#)

Symbols

`__init__()` (*craft_ai_sdk.CraftAiSdk* method), 3

A

`ARRAY` (*craft_ai_sdk.INPUT_OUTPUT_TYPES* attribute), 7

B

`base_control_api_url` (*craft_ai_sdk.CraftAiSdk* attribute), 3

`base_control_url` (*craft_ai_sdk.CraftAiSdk* attribute), 3

`base_environment_api_url` (*craft_ai_sdk.CraftAiSdk* attribute), 3

`base_environment_url` (*craft_ai_sdk.CraftAiSdk* attribute), 3

`BOOLEAN` (*craft_ai_sdk.INPUT_OUTPUT_TYPES* attribute), 7

C

`craft_ai_sdk.exceptions`
module, 1

`craft_ai_sdk.sdk`
module, 1

`CraftAiSdk` (class in *craft_ai_sdk*), 3

`create_deployment()` (*craft_ai_sdk.CraftAiSdk* method), 11

`create_endpoint()` (*craft_ai_sdk.CraftAiSdk* method), 13

`create_or_update_environment_variable()`
(*craft_ai_sdk.CraftAiSdk* method), 16

`create_pipeline()` (*craft_ai_sdk.CraftAiSdk* method), 8

`create_step()` (*craft_ai_sdk.CraftAiSdk* method), 4

D

`delete_data_store_object()`
(*craft_ai_sdk.CraftAiSdk* method), 16

`delete_deployment()` (*craft_ai_sdk.CraftAiSdk* method), 11

`delete_endpoint()` (*craft_ai_sdk.CraftAiSdk* method), 14

`delete_environment_variable()`
(*craft_ai_sdk.CraftAiSdk* method), 16

`delete_pipeline()` (*craft_ai_sdk.CraftAiSdk* method), 8

`delete_step()` (*craft_ai_sdk.CraftAiSdk* method), 7

`DEPLOYMENT_EXECUTION_RULES` (class in *craft_ai_sdk*), 13

`download_data_store_object()`
(*craft_ai_sdk.CraftAiSdk* method), 15

E

`execute_pipeline()` (*craft_ai_sdk.CraftAiSdk* method), 9

F

`FILE` (*craft_ai_sdk.INPUT_OUTPUT_TYPES* attribute), 7

G

`generate_new_endpoint_token()`
(*craft_ai_sdk.CraftAiSdk* method), 14

`get_data_store_object_information()`
(*craft_ai_sdk.CraftAiSdk* method), 15

`get_deployment()` (*craft_ai_sdk.CraftAiSdk* method), 11

`get_endpoint()` (*craft_ai_sdk.CraftAiSdk* method), 13

`get_metrics()` (*craft_ai_sdk.CraftAiSdk* method), 17

`get_pipeline()` (*craft_ai_sdk.CraftAiSdk* method), 8

`get_pipeline_execution()`
(*craft_ai_sdk.CraftAiSdk* method), 10

`get_pipeline_execution_logs()`
(*craft_ai_sdk.CraftAiSdk* method), 10

`get_step()` (*craft_ai_sdk.CraftAiSdk* method), 5

I

`Input` (class in *craft_ai_sdk*), 7

`INPUT_OUTPUT_TYPES` (class in *craft_ai_sdk*), 7

`InputSource` (class in *craft_ai_sdk*), 12

J

`JSON` (*craft_ai_sdk.INPUT_OUTPUT_TYPES* attribute), 7

L

`list_data_store_objects()`
(*craft_ai_sdk.CraftAiSdk* method), 15

`list_deployments()` (*craft_ai_sdk.CraftAiSdk method*), 11
`list_endpoints()` (*craft_ai_sdk.CraftAiSdk method*), 13
`list_environment_variables()` (*craft_ai_sdk.CraftAiSdk method*), 16
`list_pipeline_executions()` (*craft_ai_sdk.CraftAiSdk method*), 9
`list_pipelines()` (*craft_ai_sdk.CraftAiSdk method*), 8
`list_steps()` (*craft_ai_sdk.CraftAiSdk method*), 6

M

module

`craft_ai_sdk.exceptions`, 1
`craft_ai_sdk.sdk`, 1

N

`NUMBER` (*craft_ai_sdk.INPUT_OUTPUT_TYPES attribute*), 7

O

`Output` (*class in craft_ai_sdk*), 7

`OutputDestination` (*class in craft_ai_sdk*), 12

R

`record_metric_value()` (*craft_ai_sdk.CraftAiSdk method*), 17

`run_pipeline()` (*craft_ai_sdk.CraftAiSdk method*), 9

S

`SdkException`, 1

`STRING` (*craft_ai_sdk.INPUT_OUTPUT_TYPES attribute*), 8

T

`trigger_endpoint()` (*craft_ai_sdk.CraftAiSdk method*), 14

U

`update_step()` (*craft_ai_sdk.CraftAiSdk method*), 5

`upload_data_store_object()` (*craft_ai_sdk.CraftAiSdk method*), 15

V

`verbose_log` (*craft_ai_sdk.CraftAiSdk attribute*), 3

W

`warn_on_metric_outside_of_step` (*craft_ai_sdk.CraftAiSdk attribute*), 3