

---

# **Zowe Client Python SDK**

*Release 1.0.0.dev21*

**Contributors to the Zowe Project**

**Sep 25, 2024**



# CONTENTS:

- 1 Getting started** **3**
- 1.1 Installation . . . . . 3
- 1.2 Basic usage . . . . . 3
  
- 2 About the project** **5**
  
- 3 Zowe SDK packages** **7**
- 3.1 Core . . . . . 7
- 3.2 z/OS Console . . . . . 22
- 3.3 z/OS Files . . . . . 23
- 3.4 z/OS Jobs . . . . . 36
- 3.5 z/OS TSO . . . . . 41
- 3.6 z/OSMF . . . . . 44
  
- 4 Classes** **45**
- 4.1 zowe.secrets\_for\_zowe\_sdk . . . . . 45
  
- 5 Contributing** **47**
- 5.1 Git branching model . . . . . 47
- 5.2 Pull requests . . . . . 47
- 5.3 Running Tests . . . . . 48
- 5.4 Building Docs . . . . . 48
- 5.5 Code standards . . . . . 48
  
- Index** **49**



The Zowe Client Python SDK, is a set of Python packages designed to allow programmatic interactions with z/OS REST API interfaces.

Python developers can leverage the Zowe SDK in order to create powerful scripts/applications that can interact with z/OS components.



## GETTING STARTED

### 1.1 Installation

When installing the Zowe Client Python SDK you have two options.

- Install all the Zowe packages
- Install a single sub-package

The choice depends on your intentions. If you chose to install all Zowe SDK packages this means that you will install everything under the *zowe* namespace in PyPi.

Alternatively you can chose to install only a single subpackage for a smaller installation.

To install all Zowe SDK packages using pip:

```
pip install -U --pre zowe-python-sdk-bundle
```

To install only a subpackage using pip:

```
pip install -U --pre zowe.<subpackage>_for_zowe_sdk
```

To see all available sub-packages check the *Zowe SDK packages* section.

### 1.2 Basic usage

After you install the package in your project, import the class for the required sub-package (i.e *Console* class for z/OS Console commands). Create a dictionary to handle communication with the plug-in:

```
from zowe.zos_console_for_zowe_sdk import Console
profile = {
    "host": "<host address>",
    "port": 443, # Include the port if different from the default (443)
    "user": "<user>",
    "password": "<password>",
}
my_console = Console(profile)
```

Alternatively you can use an existing Zowe CLI profile instead:

```
from zowe.zos_console_for_zowe_sdk import Console
from zowe.core_for_zowe_sdk import ProfileManager

# Load the profile using ProfileManager
profile = ProfileManager().load(profile_name="<profile name>")

my_console = Console(profile)
```



## ABOUT THE PROJECT

The Zowe Client Python SDK, is a set of Python packages designed to allow programmatic interactions with z/OS REST API interfaces.

Python developers can leverage the Zowe SDK in order to create powerful scripts/applications that can interact with z/OS components.



## ZOWE SDK PACKAGES

The Zowe Client Python SDK is divided into multiple packages. Each one responsible for a single aspect of the SDK.

### 3.1 Core

Contains the core functionalities for the Zowe Client Python SDK.

#### 3.1.1 Installing

To install this package using pip issue the following command:

```
pip install zowe.zowe_core_for_zowe_sdk
```

#### 3.1.2 Reference

`zowe.core_for_zowe_sdk`

##### ApiConnection

```
class core.zowe.core_for_zowe_sdk.ApiConnection(host_url: str, user: str, password: str,  
ssl_verification: bool = True)
```

Class used to represent a connection with a REST API.

##### Parameters

- **host\_url** (*str*) – The base url of the rest api host
- **user** (*str*) – The user of the rest api
- **password** (*str*) – The password for the user
- **ssl\_verification** (*bool*) – Options for ssl verification. True by default.

##### Raises

*MissingConnectionArgs* – Missing connection argument.

## CredentialManager

**class** `core.zowe.core_for_zowe_sdk.CredentialManager`

A class including static functions for managing credentials.

**static** `load_secure_props()` → None

Load `secure_props` stored for the given config file.

if keyring is not initialized, set empty value

**Raises**

***SecureProfileLoadFailed*** – Fail to load secure profile

**static** `save_secure_props()` → None

Set `secure_props` for the given config file.

## Log

**class** `core.zowe.core_for_zowe_sdk.Log`

Class used to represent a logger.

**dirname**

Path where the log file is saved

**Type**

str

**file\_handler**

Shared FileHandler object for managing log file output

**Type**

logging.FileHandler

**console\_handler**

Shared StreamHandler object for managing log console output

**Type**

logging.StreamHandler

**file\_output**

Specifies whether log messages would be saved to a file. True by default.

**Type**

bool

**console\_output**

Specifies whether log messages would be printed out on console. True by default.

**Type**

bool

**loggers**

The set of all loggers

**Type**

set

**static close**(*logger: Logger*)

Disable a logger.

**Parameters**

**logger** (*logging.Logger*) – The logger to be turned off

**static close\_all**()

Disable all loggers.

**static close\_console\_output**()

Turn off log output to console.

**static close\_file\_output**()

Turn off log output to a file.

**static open**(*logger: Logger*)

Enable a logger.

**Parameters**

**logger** (*logging.Logger*) – The logger to be turned on

**static open\_all**()

Enable all loggers.

**static open\_console\_output**()

Turn on log output to console.

**static open\_file\_output**()

Turn on log output to a file.

**static register\_logger**(*name: str*) → *Logger*

Create and register a logger.

**Parameters**

**name** (*str*) – The name for the logger.

**Returns**

A *Logger* object named after the file where it is created.

**Return type**

*logging.Logger*

**static set\_all\_logger\_level**(*level: int*)

Set display level for all loggers.

**Parameters**

**level** (*int*) – The intended logger level

**static set\_console\_output\_level**(*level: int*)

Set the level for the console handler.

**Parameters**

**level** (*int*) – The intended console output level

**static set\_file\_output\_level**(*level: int*)

Set the level for the file handler.

**Parameters**

**level** (*int*) – The intended file output level

## ProfileManager

```
class core.zowe.core_for_zowe_sdk.ProfileManager(appname: str | None = 'zowe', show_warnings: bool | None = True)
```

Class used to manage profiles.

Profile Manager contains the logic to merge the different properties of profiles (from the Project Config and the Project User Config as well as the Global Config and Global User Config). This class handles all the exceptions raised in the Config File to provide a smooth user experience.

### Parameters

- **appname** (*Optional[str]*) – Name of the app
- **show\_warnings** (*Optional[bool]*) – Indicates whether warnings are shown

**property config\_appname: str**

Return the application name.

### Returns

The name of the application as configured in the current instance.

### Return type

str

**property config\_dir: str | None**

Return the folder path where the Zowe z/OSMF Team Project Config files are stored.

### Returns

The directory path where the main project configuration files are located. This path can be None if not set.

### Return type

Optional[str]

**property config\_filename: str**

Return the filename for the Zowe z/OSMF Team Project Config.

### Returns

The filename of the main project configuration file.

### Return type

str

**property config\_filepath: str | None**

Get the full filepath for the Zowe z/OSMF Team Project Config file.

### Returns

Filepath of configuration file or None if the location or filename is not set.

### Return type

Optional[str]

**static get\_env(cfg: ConfigFile, cwd: str | None = None) → dict**

Map the env variables to the profile properties.

### Parameters

- **cfg** ([ConfigFile](#)) – A config file that contains the schema properties
- **cwd** (*Optional[str]*) – Path of current working directory

**Returns**

Containing profile properties from env variables (prop: value)

**Return type**

dict

**get\_highest\_priority\_layer**(*json\_path*: str) → *ConfigFile* | None

Get the highest priority layer (configuration file) based on the given profile name.

**Parameters**

**json\_path** (str) – The path of the json.

**Raises**

**FileNotFoundError** – File is not found in given path.

**Returns**

The highest priority layer (configuration file) that contains the specified profile, or None if the profile is not found in any layer.

**Return type**

Optional[*ConfigFile*]

**static get\_profile**(*cfg*: *ConfigFile*, *profile\_name*: str | None, *profile\_type*: str | None, *validate\_schema*: bool | None = True) → *Profile*

Retrieve a profile from the configuration file, optionally validating the schema.

**Parameters**

- **cfg** (*ConfigFile*) – The configuration file object which contains the profiles.
- **profile\_name** (*Optional[str]*) – The name of the profile to retrieve. If None, the method attempts to fetch the profile based only on the type.
- **profile\_type** (*Optional[str]*) – The type of the profile to retrieve. If None, the method attempts to fetch the profile based only on the name.
- **validate\_schema** (*Optional[bool]*) – Whether to validate the profile against the schema present in the configuration file.

**Returns**

A NamedTuple containing the profile data, name, and any secure properties not found.

**Return type**

*Profile*

**Raises**

- **ValidationError** – If the instance is invalid under the provided schema.
- **SchemaError** – If the provided schema itself is invalid.
- **UndefinedTypeCheck** – If a type checker is asked to check a type it does not have registered.
- **UnknownType** – If an unknown type is found in the schema.
- **FormatError** – If validating a format in the configuration fails.

**load**(*profile\_name*: str | None = None, *profile\_type*: str | None = None, *check\_missing\_props*: bool = True, *validate\_schema*: bool | None = True, *override\_with\_env*: bool | None = False, *suppress\_config\_file\_warnings*: bool | None = True) → dict

Load connection details from a team config profile.

We will load properties from config files in the following order, from highest to lowest priority: 1. Project User Config (`./zowe.config.user.json`) 2. Project Config (`./zowe.config.json`) 3. Global User Config (`~/zowe.config.user.json`) 4. Global Config (`~/zowe.config.json`)

If `profile_type` is not base, then we will load properties from both `profile_type` and base profiles and merge them together.

### Parameters

- **profile\_name** (*Optional[str]*) – The name of the profile to load. If None, profiles are loaded based only on profile type.
- **profile\_type** (*Optional[str]*) – The type of the profile to load, e.g., ‘zosmf’, ‘zftp’. If None, profiles are loaded based only on name.
- **check\_missing\_props** (*bool*) – Flag to indicate whether to check for missing secure properties.
- **validate\_schema** (*Optional[bool]*) – Whether to validate the loaded profile against the schema defined in the configuration.
- **override\_with\_env** (*Optional[bool]*) – If True, overrides profile properties with values from environment variables.
- **suppress\_config\_file\_warnings** (*Optional[bool]*) – Suppresses warnings from the configuration file loading process.

### Raises

- **ProfileNotFound** – If both `profile_name` and `profile_type` are not provided, indicating which profile to load.
- **SecureValuesNotFound** – If any secure properties are required but not found or loaded.

### Returns

A dictionary containing the merged connection details from all relevant profiles.

### Return type

dict

**save()** → None

Save the layers (configuration files) to disk.

**set\_profile(profile\_path: str, profile\_data: dict)** → None

Set a profile in the highest priority layer (configuration file) based on the given profile name.

### Parameters

- **profile\_path** (*str*) – The path of the profile to be set. eg: profiles.zosmf
- **profile\_data** (*dict*) – The data of the profile to set.

**set\_property(json\_path: str, value: str, secure: bool | None = None)** → None

Set a property in the profile, storing it securely if necessary.

### Parameters

- **json\_path** (*str*) – The JSON path of the property to set.
- **value** (*str*) – The value to be set for the property.
- **secure** (*Optional[bool]*) – If True, the property will be stored securely. Default is None.



**property user\_config\_dir:** str | None

Return the folder path where the Zowe z/OSMF User Project Config files are stored.

**Returns**

The directory path where the user-specific project configuration files are located.

**Return type**

Optional[str]

## RequestHandler

**class** core.zowe.core\_for\_zowe\_sdk.**RequestHandler**(*session\_arguments: dict, logger\_name: str = 'core.zowe.core\_for\_zowe\_sdk.request\_handler'*)

Class used to handle HTTP/HTTPS requests.

**Parameters**

- **session\_arguments** (*dict*) – Zowe SDK session arguments
- **logger\_name** (*str*) – The logger name of the modules calling request handler

**\_\_del\_\_**()

Clean up the REST session object once it is no longer needed anymore.

**perform\_request**(*method: str, request\_arguments: dict, expected\_code: list = [200], stream: bool = False*)  
→ str | bytes | dict | None

Execute an HTTP/HTTPS requests from given arguments and return validated response (JSON).

**Parameters**

- **method** (*str*) – The request method that should be used
- **request\_arguments** (*dict*) – The dictionary containing the required arguments for the execution of the request
- **expected\_code** (*list*) – The list containing the acceptable response codes (default is [200])
- **stream** (*bool*) – The boolean value whether the request is stream

**Returns**

normalized request response in json (dictionary)

**Return type**

Union[str, bytes, dict, None]

## SdkApi

**class** core.zowe.core\_for\_zowe\_sdk.**SdkApi**(*profile: dict, default\_url: str, logger\_name: str = 'core.zowe.core\_for\_zowe\_sdk.sdk\_api', log: bool = True*)

Abstract class used to represent the base SDK API.

**Parameters**

- **profile** (*dict*) – Profile information in json (dict) format
- **default\_url** (*str*) – Default url used for session
- **logger\_name** (*str*) – Name of the logger (same as the filename by default)

- **log** (*bool*) – Flag to disable logger

**\_\_enter\_\_**()

Return the SdkApi instance.

**\_\_exit\_\_**(*exc\_type, exception, traceback*)

Delete the request handler before exit.

## ZosmfProfile

**class** `core.zowe.core_for_zowe_sdk.ZosmfProfile(profile_name: str)`

Class used to represent a Zowe z/OSMF profile.

### Description

This class is only used when there is already a Zowe z/OSMF profile created and the user opted to use the profile instead of passing the credentials directly in the object constructor.

**param profile\_name**

Zowe z/OSMF profile name

**type profile\_name**

str

**load**() → *ApiConnection*

Load z/OSMF connection details from a z/OSMF profile.

**Returns**

z/OSMF connection object

**Return type**

*ApiConnection*

**property profiles\_dir: str**

Return the os path for the Zowe z/OSMF profiles.

**Returns**

the os path for the Zowe z/OSMF profiles

**Return type**

str

## Config File classes

### ConfigFile

**class** `core.zowe.core_for_zowe_sdk.config_file.ConfigFile(type: str, name: str, _location: str | None = None, profiles: dict | None = None, defaults: dict | None = None, schema_property: dict | None = None, secure_props: dict | None = None, jsonc: dict | None = None, _missing_secure_props: list = <factory>)`

Class used to represent a single config file.

**Mainly it will have the following details :**

1. **Type (“User Config” or “Team Config”)**  
User Configs override Team Configs. User Configs are used to have personalised config details that the user don’t want to have in the Team Config.
2. Directory in which the file is located.
3. Name (excluding .config.json or .config.user.json)
4. **Contents of the file.**  
4.1 Profiles 4.2 Defaults 4.3 Schema Property
5. Secure Properties associated with the file.

**autodiscover\_config\_dir()** → None

Autodiscover Zowe z/OSMF Team Config files by going up the path from current working directory.

Sets path if it finds the config directory, Else, it raises an Exception.

**Raises**

**FileNotFoundError** – Cannot find file in directory.

**find\_profile(path: str, profiles: dict)** → dict | None

Find a profile at a specified location from within a set of nested profiles.

**Parameters**

- **path** (*str*) – The location to look for the profile
- **profiles** (*dict*) – A dict of nested profiles

**Returns**

The profile object that was found, or None if not found

**Return type**

Optional[dict]

**get\_profile(profile\_name: str | None = None, profile\_type: str | None = None, validate\_schema: bool | None = True)** → *Profile*

Load given profile including secure properties and excluding values from base profile.

**Parameters**

- **profile\_name** (*Optional[str]*) – Name of the profile
- **profile\_type** (*Optional[str]*) – Type of the profile
- **validate\_schema** (*Optional[bool]*) – True if validation is preferred

**Raises**

**ProfileNotFound** – Cannot find profile

**Returns**

Returns a namedtuple called Profile

**Return type**

*Profile*

**get\_profile\_name\_from\_path**(*path: str*) → str

Get the name of the profile from the given path.

**Parameters**

**path** (*str*) – The location to look for the profile

**Returns**

Returns the profile name

**Return type**

str

**get\_profile\_path\_from\_name**(*short\_path: str*) → str

Get the path of the profile from the given name.

**Parameters**

**short\_path** (*str*) – Partial path of profile

**Returns**

Returns the full profile path

**Return type**

str

**get\_profile\_name\_from\_profile\_type**(*profile\_type: str*) → str

Return profile name from given profile type as defined in the team config profile.

**Parameters**

**profile\_type** (*str*) – Type of the profile

**Returns**

The exact profile name of the profile to load from the mentioned type.

**Return type**

str

**Raises**

**ProfileNotFound** – Cannot find profile

**init\_from\_file**(*validate\_schema: bool | None = True, suppress\_config\_file\_warnings: bool | None = True*) → None

Initialize the class variable after setting filepath (or if not set, autodiscover the file).

**Parameters**

- **validate\_schema** (*Optional [bool]*) – True if validation is preferred, false otherwise
- **suppress\_config\_file\_warnings** (*Optional [bool]*) – True if the property should be stored securely, False otherwise.

**load\_profile\_properties**(*profile\_name: str*) → dict

Load profile properties given profile\_name including secure properties.

Load exact profile properties (without prepopulated fields from base profile) from the profile dict and populate fields from the secure credentials storage

**Parameters**

**profile\_name** (*str*) – Name of the profile

**Returns**

Object containing profile properties

**Return type**

dict

**save**(*update\_secure\_props: bool | None = True*) → None

Save the config file to disk. and secure props to vault.

**Parameters****update\_secure\_props** (*Optional[bool]*) – If True, the secure properties will be stored in the vault. Default is True.**schema\_list**(*cwd: str = None*) → list

Load the schema properties in a sorted order according to the priority.

**Parameters****cwd** (*str*) – current working directory**Returns**

Return the profile properties from schema (prop: value)

**Return type**

list

**set\_profile**(*profile\_path: str, profile\_data: dict*) → None

Set a profile in the config file.

**Parameters**

- **profile\_path** (*str*) – The path of the profile to be set. eg: profiles.zosmf
- **profile\_data** (*dict*) – The data to be set for the profile.

**set\_property**(*json\_path: str, value: str, secure: bool | None = None*) → None

Set a property in the profile, storing it securely if necessary.

**Parameters**

- **json\_path** (*str*) – The JSON path of the property to set.
- **value** (*str*) – The value to be set for the property.
- **secure** (*Optional[bool]*) – If True, the property will be stored securely. Default is None.

**validate\_schema**() → None

Get the \$schema\_property from the config and load the schema.

## Profile

```
class core.zowe.core_for_zowe_sdk.config_file.Profile(data: dict = {}, name: str = "",
                                                    missing_secure_props: list = [])
```

Class to represent a profile.

**data: dict**

Alias for field number 0

**missing\_secure\_props: list**

Alias for field number 2

**name: str**

Alias for field number 1

## Custom Warnings classes

### ConfigNotFoundWarning

**class** `core.zowe.core_for_zowe_sdk.custom_warnings.ConfigNotFoundWarning`(*message: str*)

A warning that is raised when a configuration file is not found.

**Parameters**

**message** (*str*) – A human-readable string describing the warning.

**\_\_str\_\_**() → *str*

Return a string representation of the warning message.

**Returns**

a string representation of the warning message

**Return type**

*str*

### ProfileNotFoundWarning

**class** `core.zowe.core_for_zowe_sdk.custom_warnings.ProfileNotFoundWarning`(*message: str*)

A warning that is raised when a user profile cannot be found.

**Parameters**

**message** (*str*) – A string describing the warning.

**\_\_str\_\_**() → *str*

Return a string representation of the warning message.

**Returns**

a string representation of the warning message

**Return type**

*str*

### ProfileParsingWarning

**class** `core.zowe.core_for_zowe_sdk.custom_warnings.ProfileParsingWarning`(*message: str*)

A warning that is raised when there is an error while parsing a user profile.

**Parameters**

**message** (*str*) – A human-readable string describing the warning.

**\_\_str\_\_**() → *str*

Return a string representation of the warning message.

**Returns**

a string representation of the warning message

**Return type**

*str*

## SecurePropsNotFoundWarning

**class** `core.zowe.core_for_zowe_sdk.custom_warnings.SecurePropsNotFoundWarning`(*message: str*)

A warning that is raised when secure properties are not found.

**Parameters**

**message** (*str*) – A human-readable string describing the warning.

**\_\_str\_\_**() → *str*

Return a string representation of the warning message.

**Returns**

a string representation of the warning message

**Return type**

*str*

## Exceptions classes

### FileNotFound

**class** `core.zowe.core_for_zowe_sdk.exceptions.FileNotFound`(*input\_path: str*)

Class used to represent a file not found exception.

**Parameters**

**input\_path** (*str*) – The invalid input path

### InvalidRequestMethod

**class** `core.zowe.core_for_zowe_sdk.exceptions.InvalidRequestMethod`(*input\_method: str*)

Class used to represent an invalid request method exception.

**Parameters**

**input\_method** (*str*) – The invalid HTTP method used

### MissingConnectionArgs

**class** `core.zowe.core_for_zowe_sdk.exceptions.MissingConnectionArgs`

Class used to represent a missing connection argument exception.

### ProfileNotFound

**class** `core.zowe.core_for_zowe_sdk.exceptions.ProfileNotFound`(*profile\_name: str = 'unknown', error\_msg: str = 'error'*)

Class used to represent a profile load failure exception.

**Parameters**

- **profile\_name** (*str*) – The name of the profile it failed to load
- **error\_msg** (*str*) – The error message received while trying to load the profile

## RequestFailed

```
class core.zowe.core_for_zowe_sdk.exceptions.RequestFailed(status_code: int, request_output: str)
```

Class used to represent a request failure exception.

### Parameters

- **status\_code** (*int*) – The status code from the failed request
- **request\_output** (*str*) – The output from the request

## SecureProfileLoadFailed

```
class core.zowe.core_for_zowe_sdk.exceptions.SecureProfileLoadFailed(profile_name: str =  
                                                                    'unknown', error_msg: str  
                                                                    = 'error')
```

Class used to represent a secure profile load failure exception.

### Parameters

- **profile\_name** (*str*) – The name of the profile it failed to load
- **error\_msg** (*str*) – The error message received while trying to load the profile

## SecureValuesNotFound

```
class core.zowe.core_for_zowe_sdk.exceptions.SecureValuesNotFound(values: set)
```

Class used to represent a profile load failure exception.

### Parameters

- **values** (*set*) – The list of secure values not found

## UnexpectedStatus

```
class core.zowe.core_for_zowe_sdk.exceptions.UnexpectedStatus(expected: int, received: int,  
                                                             request_output: int)
```

Class used to represent an unexpected request response status exception.

### Parameters

- **expected** (*int*) – The expected status code
- **received** (*int*) – The received status code
- **request\_output** (*int*) – The output from the request



## UnsupportedAuthType

```
class core.zowe.core_for_zowe_sdk.exceptions.UnsupportedAuthType(auth_type: str)
```

Class used to represent an unsupported authentication type exception.

### Parameters

**auth\_type** (*str*) – The type of authentication on the session

## Session classes

### ISession

```
class core.zowe.core_for_zowe_sdk.session.ISession(host: str, port: int = 443, reject_unauthorized:
    bool = True, user: str | None = None, password:
    str | None = None, protocol: str = 'https',
    base_path: str | None = None, type: str | None =
    None, token_type: str | None = None, token_value:
    str | None = None, cert: str | None = None)
```

Class to represent session parameters.

### Session

```
class core.zowe.core_for_zowe_sdk.session.Session(props: dict)
```

Class used to set connection details received from a ProfileManager or manually set by passing and ISession object.

### Parameters

**props** (*dict*) – Profile and properties

### Raises

**Exception** – Exception thrown when cert key is not provided

**property host\_url: str**

Return the formatted host URL.

### Returns

the formatted host URL

### Return type

str

**load()** → *ISession*

Load a ISession object.

### Returns

A custom ISession object.

### Return type

*ISession*

## 3.2 z/OS Console

Contains the z/OSMF Console REST API functionalities.

### 3.2.1 Installing

```
pip install zowe.zos_console_for_zowe_sdk
```

### 3.2.2 Reference

#### zowe.zos\_console\_for\_zowe\_sdk

##### Console

**class** `zos_console.zowe.zos_console_for_zowe_sdk.Console`(*connection: dict, log: bool = True*)

Class used to represent the base z/OSMF Console API.

##### Parameters

- **connection** (*dict*) – A profile in dict (json) format
- **log** (*bool*) – Flag to disable logger

**get\_response**(*response\_key: str, console: str | None = None*) → `ConsoleResponse`

Collect outstanding synchronous z/OS Console response messages.

##### Parameters

- **response\_key** (*str*) – The command response key from the Issue Command request.
- **console** (*Optional[str]*) – The console that should be used to get the command response.

##### Returns

A JSON containing the response to the command

##### Return type

`ConsoleResponse`

**issue\_command**(*command: str, console: str | None = None*) → `IssueCommandResponse`

Issues a command on z/OS Console.

##### Parameters

- **command** (*str*) – The z/OS command to be executed
- **console** (*Optional[str]*) – Name of the console that should be used to execute the command (default is None)

##### Returns

A JSON containing the response from the console command

##### Return type

`IssueCommandResponse`

## 3.3 z/OS Files

Contains the z/OSMF Files REST API functionalities.

### 3.3.1 Installing

```
pip install zowe.zos_files_for_zowe_sdk
```

### 3.3.2 Reference

`zowe.zos_files_for_zowe_sdk`

**FileType**

**FileSystems**

**class** `zos_files.zowe.zos_files_for_zowe_sdk.FileSystems`(*connection: dict, log: bool = True*)

Class used to represent the base z/OSMF FileSystems API.

It includes all operations related to file systems.

#### Parameters

- **connection** (*dict*) – A profile for connection in dict (json) format
- **log** (*bool*) – Flag to disable logger

**create**(*file\_system\_name: str, options: dict = {}*) → *dict*

Create a z/OS UNIX zFS Filesystem.

#### Parameters

- **file\_system\_name** (*str*) – Name of the file system
- **options** (*dict*) – Specifies file system attributes

#### Raises

- ***MaxAllocationQuantityExceeded*** – Thrown when file system exceeds max allocation quantity
- ***InvalidPermsOption*** – Thrown when invalid permission option is provided

#### Returns

A JSON containing the result of the operation

#### Return type

*dict*

**delete**(*file\_system\_name: str*) → *dict*

Delete a zFS Filesystem.

#### Parameters

**file\_system\_name** (*str*) – Name of the file system

#### Returns

A JSON containing the result of the operation

**Return type**

dict

**list**(*file\_path\_name: str | None = None, file\_system\_name: str | None = None*) → FileSystemListResponse

List all mounted filesystems.

It could also list the specific filesystem mounted at a given path, or the filesystem with a given Filesystem name.

**Parameters**

- **file\_path\_name** (*Optional[str]*) – USS directory that contains the files and directories to be listed
- **file\_system\_name** (*Optional[str]*) – Name of the file system to be listed

**Returns**

A JSON containing the result of the operation

**Return type**

FileSystemListResponse

**mount**(*file\_system\_name: str, mount\_point: str, options: dict = {}, encoding: str = 'utf-8'*) → dict

Mount a z/OS UNIX file system on a specified directory.

**Parameters**

- **file\_system\_name** (*str*) – Name for the file system
- **mount\_point** (*str*) – Mount point to be used for mounting the UNIX file system
- **options** (*dict*) – A JSON of request body options
- **encoding** (*str*) – Specifies optional encoding name (e.g. IBM-1047)

**Returns**

A JSON containing the result of the operation

**Return type**

dict

**unmount**(*file\_system\_name: str, options: dict = {}, encoding: str = 'utf-8'*) → dict

Unmount a z/OS UNIX file system on a specified directory.

**Parameters**

- **file\_system\_name** (*str*) – Name for the file system
- **options** (*dict*) – A JSON of request body options
- **encoding** (*str*) – Specifies optional encoding name (e.g. IBM-1047)

**Returns**

A JSON containing the result of the operation

**Return type**

dict

## Files

**class** `zos_files.zowe.zos_files_for_zowe_sdk.Files`(*connection: dict, log: bool = True*)

Class used to represent the base z/OSMF Files API.

### **ds**

A Datasets class object

#### **Type**

*Datasets*

### **uss**

An USSFiles class object

#### **Type**

*USSFiles*

### **fs**

A FileSystems class object

#### **Type**

*FileSystems*

### **Parameters**

- **connection** (*dict*) – The z/OSMF connection object (generated by the ZoweSDK object)
- **log** (*bool*) – Flag to disable logger

**copy\_data\_set\_or\_member**(*from\_dataset\_name, to\_dataset\_name, from\_member\_name=None, volser=None, alias=None, to\_member\_name=None, enq=None, replace=False*)

Use `ds.copy_data_set_or_member()` instead of this deprecated function.

**copy\_uss\_to\_data\_set**(*from\_filename, to\_dataset\_name, to\_member\_name=None, type=FileType.TEXT, replace=False*)

Use `ds.copy_uss_to_data_set()` instead of this deprecated function.

**create\_data\_set**(*dataset\_name, options: DatasetOption | None = None*)

Use `ds.create()` instead of this deprecated function.

**create\_default\_data\_set**(*dataset\_name: str, default\_type: str*)

Use `ds.create_default()` instead of this deprecated function.

**create\_uss**(*file\_path, type, mode=None*)

Use `uss.create()` instead of this deprecated function.

**create\_zfs\_file\_system**(*file\_system\_name, options={}*)

Use `fs.create()` instead of this deprecated function.

**delete\_data\_set**(*dataset\_name, volume=None, member\_name=None*)

Use `ds.delete()` instead of this deprecated function.

**delete\_migrated\_data\_set**(*dataset\_name: str, purge=False, wait=False*)

Use `ds.delete_migrated()` instead of this deprecated function.

**delete\_uss**(*filepath\_name, recursive=False*)

Use `uss.delete()` instead of this deprecated function.

**delete\_zfs\_file\_system**(*file\_system\_name*)

Use fs.delete() instead of this deprecated function.

**download\_binary\_dsn**(*dataset\_name*, *output\_file*, *with\_prefixes=False*)

Use ds.download\_binary() instead of this deprecated function.

**download\_dsn**(*dataset\_name*, *output\_file*)

Use ds.download() instead of this deprecated function.

**download\_uss**(*file\_path*, *output\_file*, *binary=False*)

Use uss.download() instead of this deprecated function.

**get\_dsn\_binary\_content**(*dataset\_name*, *with\_prefixes=False*)

Use ds.get\_binary\_content() instead of this deprecated function.

**get\_dsn\_binary\_content\_streamed**(*dataset\_name*, *with\_prefixes=False*)

Use ds.get\_binary\_content() instead of this deprecated function.

**get\_dsn\_content**(*dataset\_name*)

Use ds.get\_content() instead of this deprecated function.

**get\_dsn\_content\_streamed**(*dataset\_name*)

Use ds.get\_content() instead of this deprecated function.

**get\_file\_content**(*filepath\_name*)

Use uss.get\_content() instead of this deprecated function.

**get\_file\_content\_streamed**(*file\_path*, *binary=False*)

Use uss.get\_content\_streamed() instead of this deprecated function.

**list\_dsn**(*name\_pattern*, *return\_attributes=False*)

Use ds.list() instead of this deprecated function.

**list\_dsn\_members**(*dataset\_name*, *member\_pattern=None*, *member\_start=None*, *limit=1000*,  
*attributes='member'*)

Use ds.list\_members() instead of this deprecated function.

**list\_files**(*path*)

Use uss.list() instead of this deprecated function.

**list\_unix\_file\_systems**(*file\_path\_name=None*, *file\_system\_name=None*)

Use fs.list() instead of this deprecated function.

**migrate\_data\_set**(*dataset\_name: str*, *wait=False*)

Use ds.migrate() instead of this deprecated function.

**mount\_file\_system**(*file\_system\_name*, *mount\_point*, *options={}*, *encoding='utf-8'*)

Use fs.mount() instead of this deprecated function.

**recall\_migrated\_data\_set**(*dataset\_name: str*, *wait=False*)

Use ds.recall\_migrated() instead of this deprecated function.

**rename\_data\_set**(*before\_dataset\_name: str*, *after\_dataset\_name: str*)

Use ds.rename() instead of this deprecated function.

**rename\_data\_set\_member**(*dataset\_name: str*, *before\_member\_name: str*, *after\_member\_name: str*,  
*enq=""*)

Use ds.rename\_member() instead of this deprecated function.

**unmount\_file\_system**(*file\_system\_name*, *options*={}, *encoding*='utf-8')

Use fs.unmount() instead of this deprecated function.

**upload\_file\_to\_dsn**(*input\_file*, *dataset\_name*, *encoding*='utf-8', *binary*=False)

Use ds.upload\_file() instead of this deprecated function.

**upload\_file\_to\_uss**(*input\_file*, *filepath\_name*, *encoding*='utf-8')

Use uss.upload() instead of this deprecated function.

**write\_to\_dsn**(*dataset\_name*, *data*, *encoding*='utf-8')

Use ds.write() instead of this deprecated function.

**write\_to\_uss**(*filepath\_name*, *data*, *encoding*='utf-8')

Use uss.write() instead of this deprecated function.

## USSFiles

**class** zos\_files.zowe.zos\_files\_for\_zowe\_sdk.USSFiles(*connection*: dict, *log*: bool = True)

Class used to represent the base z/OSMF USSFiles API.

It includes all operations related to USS files.

### Parameters

- **connection** (dict) – The z/OSMF connection object (generated by the ZoweSDK object)
- **log** (bool) – Flag to disable logger

**create**(*file\_path*: str, *type*: str, *mode*: str | None = None) → dict

Add a file or directory.

### Parameters

- **file\_path** (str) – Path of the file to add
- **type** (str) – Specify either “file” or “dir”
- **mode** (Optional[str]) – Unix permission string (e.g. *rwxr-xr-x*)

### Returns

A JSON containing the operation results

### Return type

dict

**delete**(*filepath\_name*: str, *recursive*: bool = False) → dict

Delete a file or directory.

### Parameters

- **filepath\_name** (str) – Path of the file to be deleted
- **recursive** (bool) – If specified as True, all the files and sub-directories will be deleted.

### Returns

A JSON containing the operation results

### Return type

dict

**download**(*file\_path: str, output\_file: str, binary: bool = False*)

Retrieve the contents of a USS file and saves it to a local file.

**Parameters**

- **file\_path** (*str*) – Path of the file to be downloaded
- **output\_file** (*str*) – Name of the file to be saved locally
- **binary** (*bool*) – Specifies whether the contents are binary

**get\_content**(*filepath\_name: str*) → dict

Retrieve the content of a filename. The complete path must be specified.

**Parameters**

**filepath\_name** (*str*) – Path of the file

**Returns**

A JSON with the contents of the specified USS file

**Return type**

dict

**get\_content\_streamed**(*file\_path: str, binary: bool = False*) → dict

Retrieve the contents of a given USS file streamed.

**Parameters**

- **file\_path** (*str*) – Path of the file
- **binary** (*bool*) – Specifies whether the contents are binary

**Returns**

A JSON response with results of the operation

**Return type**

dict

**list**(*path: str*) → USSListResponse

Retrieve a list of USS files based on a given pattern.

**Parameters**

**path** (*str*) – Path to retrieve the list

**Returns**

A JSON with a list of file names matching the given pattern

**Return type**

USSListResponse

**upload**(*input\_file: str, filepath\_name: str, encoding: str = 'utf-8'*)

Upload contents of a given file and saves it to a file at the given USS path.

**Parameters**

- **input\_file** (*str*) – Name of the file to be uploaded
- **filepath\_name** (*str*) – Path of the file where it will be created
- **encoding** (*str*) – Specifies encoding schema

**Raises**

**FileNotFound** – Thrown when specific file is not found.



**write**(*filepath\_name: str, data: str, encoding: str = 'utf-8'*) → dict

Write content to an existing UNIX file.

#### Parameters

- **filepath\_name** (*str*) – Path of the file
- **data** (*str*) – Contents to be written
- **encoding** (*str*) – Specifies the encoding name (e.g. IBM-1047)

#### Returns

A JSON containing the result of the operation

#### Return type

dict

## Datasets classes

### DatasetOption

```
class zos_files.zowe.zos_files_for_zowe_sdk.datasets.DatasetOption(like: str | None = None,
volser: str | None = None,
unit: str | None = None,
dsorg: str | None = None,
alcunit: str | None = None,
primary: int | None = None,
secondary: int | None =
None, dirblk: int | None =
None, avgbk: int | None =
None, recfm: str | None =
None, blksize: int | None =
None, lrecl: int | None =
None, storclass: str | None =
None, mgmtclass: str | None =
None, dataclass: str |
None = None, dsntype: str |
None = None)
```

A dataclass that represents options for creating a dataset.

#### Parameters

- **like** (*Optional[str]*) – The dataset name to copy attributes from
- **volser** (*Optional[str]*) – The volume serial number that identifies where the dataset resides or should be allocated
- **unit** (*Optional[str]*) – Specifies the type of device on which the dataset is to be stored
- **dsorg** (*Optional[str]*) – Defines the organization of the dataset (PS for sequential, PO for partitioned, DA for direct access)
- **alcunit** (*Optional[str]*) – Specifies the unit of space allocation for the dataset (CYL for cylinders, TRK for tracks, BLK for blocks)
- **primary** (*Optional[int]*) – The amount of primary space to allocate for the dataset
- **secondary** (*Optional[int]*) – The amount of secondary space to allocate if the primary space is exhausted

- **dirblk** (*Optional[int]*) – The number of directory blocks to allocate for a partitioned dataset
- **avgblk** (*Optional[int]*) – The average block size for the dataset
- **recfm** (*Optional[str]*) – The format of the records in the dataset
- **blksize** (*Optional[int]*) – The physical block size used for the dataset
- **lrecl** (*Optional[int]*) – The length of the logical records in the dataset
- **storclass** (*Optional[str]*) – Specifies the storage class to be used for the dataset
- **mgmtclass** (*Optional[str]*) – Specifies the management class
- **dataclass** (*Optional[str]*) – Specifies the data class for the dataset
- **dsntype** (*Optional[str]*) – Specifies the type of dataset

**property alcunit:** `str | None`

Get the unit of space allocation.

**property avgblk:** `int | None`

Get the average block size.

**property blksize:** `int | None`

Get the physical block size.

**property dataclass:** `str | None`

Get the data class.

**property dirblk:** `int | None`

Get the number of directory blocks.

**property dsntype:** `str | None`

Get the type of dataset.

**property dsorg:** `str | None`

Get the organization of the dataset.

**property like:** `str | None`

Get the dataset name to copy attributes from.

**property lrecl:** `int | None`

Get the length of logical records.

**property mgmtclass:** `str | None`

Get the management class.

**property primary:** `int | None`

Get the primary space allocation.

**property recfm:** `str | None`

Get the record format.

**property secondary:** `int | None`

Get the secondary space allocation.

**property storclass:** `str | None`

Get the storage class.

**to\_dict()** → dict

Return the DatasetOption as a dict.

**property unit:** str | None

Get the type of device.

**property volser:** str | None

Get the volume serial number.

## Datasets

**class** zos\_files.zowe.zos\_files\_for\_zowe\_sdk.datasets.Datasets(*connection: dict, log: bool = True*)

Class used to represent the base z/OSMF Datasets API.

It includes all operations related to datasets.

### Parameters

- **connection** (*dict*) – A profile for connection in dict (json) format
- **log** (*bool*) – Flag to disable logger

**copy\_data\_set\_or\_member**(*from\_dataset\_name: str, to\_dataset\_name: str, from\_member\_name: str | None = None, volser: str | None = None, alias: bool | None = None, to\_member\_name: str | None = None, enq: str | None = None, replace: bool = False*) → dict

Copy a dataset or member to another dataset or member.

### Parameters

- **from\_dataset\_name** (*str*) – Name of the dataset to copy from
- **to\_dataset\_name** (*str*) – Name of the dataset to copy to
- **from\_member\_name** (*Optional[str]*) – Name of the member to copy from
- **volser** (*Optional[str]*) – Volume serial number of the dataset to copy from
- **alias** (*Optional[bool]*) – Alias of the dataset to copy from
- **to\_member\_name** (*Optional[str]*) – Name of the member to copy to
- **enq** (*Optional[str]*) – Enqueue type for the dataset to copy from
- **replace** (*bool*) – If true, members in the target data set are replaced

### Raises

**ValueError** – Thrown when enq has an invalid value

### Returns

A JSON containing the result of the operation

### Return type

dict

**copy\_uss\_to\_data\_set**(*from\_filename: str, to\_dataset\_name: str, to\_member\_name: str | None = None, type: str = FileType.TEXT, replace: bool = False*) → dict

Copy a USS file to dataset.

### Parameters

- **from\_filename** (*str*) – Name of the file to copy from.

- **to\_dataset\_name** (*str*) – Name of the dataset to copy to.
- **to\_member\_name** (*Optional*[*str*]) – Name of the member to copy to.
- **type** (*str*) – Type of the file to copy from. Default is `FileType.TEXT`.
- **replace** (*bool*) – If true, members in the target dataset are replaced.

**Returns**

A JSON containing the result of the operation.

**Return type**

dict

**create**(*dataset\_name: str, options: DatasetOption | None = None*) → dict

Create a sequential or partitioned dataset.

**Parameters**

- **dataset\_name** (*str*) – Name of the dataset to be created
- **options** (*Optional*[*DatasetOption*]) – A `DatasetOption` class with property options of the dataset

**Raises**

**ValueError** – Thrown when a parameter has an invalid value

**Returns**

A JSON containing the result of the operation

**Return type**

dict

**create\_default**(*dataset\_name: str, default\_type: str*) → dict

Create a dataset with default options set.

Default options depend on the requested type.

**Parameters**

- **dataset\_name** (*str*) – The name of the dataset
- **default\_type** (*str*) – The type of the dataset: “partitioned”, “sequential”, “classic”, “c” or “binary”

**Raises**

**ValueError** – Thrown when a parameter is invalid

**Returns**

A JSON containing the result of the operation

**Return type**

dict

**delete**(*dataset\_name: str, volume: str | None = None, member\_name: str | None = None*) → dict

Delete a sequential or partitioned data.

**Parameters**

- **dataset\_name** (*str*) – The name of the dataset
- **volume** (*Optional*[*str*]) – The optional volume serial number
- **member\_name** (*Optional*[*str*]) – The name of the member to be deleted

**Returns**

A JSON containing the result of the operation

**Return type**

dict

**delete\_migrated**(*dataset\_name: str, purge: bool = False, wait: bool = False*) → dict

Delete migrated data set.

**Parameters**

- **dataset\_name** (*str*) – Name of the data set
- **purge** (*bool*) – If true, the function uses the PURGE=YES on ARCHDEL request, otherwise it uses the PURGE=NO.
- **wait** (*bool*) – If true, the function waits for completion of the request, otherwise the request is queued.

**Returns**

A JSON containing the result of the operation

**Return type**

dict

**download**(*dataset\_name: str, output\_file: str*) → None

Retrieve the contents of a dataset and saves it to a given file.

**Parameters**

- **dataset\_name** (*str*) – Name of the dataset to be downloaded
- **output\_file** (*str*) – Name of the file to be saved locally

**download\_binary**(*dataset\_name: str, output\_file: str, with\_prefixes: bool = False*) → None

Retrieve the contents of a binary dataset and saves it to a given file.

**Parameters**

- **dataset\_name** (*str*) – Name of the dataset to download
- **output\_file** (*str*) – Name of the local file to create
- **with\_prefixes** (*bool*) – If true, include a four big endian bytes record length prefix. Default: False

**get\_binary\_content**(*dataset\_name: str, stream: bool = False, with\_prefixes: bool = False*) → dict

Retrieve the contents of a given dataset as a binary bytes object.

**Parameters**

- **dataset\_name** (*str*) – Name of the dataset to retrieve
- **stream** (*bool*) – Specifies whether the request is streaming
- **with\_prefixes** (*bool*) – If True include a 4 byte big endian record len prefix. Default: False

**Returns**

A JSON with the contents of a given dataset

**Return type**

dict

**get\_content**(*dataset\_name: str, stream: bool = False*) → dict

Retrieve the contents of a given dataset.

**Parameters**

- **dataset\_name** (*str*) – The name of the dataset
- **stream** (*bool*) – Specifies whether the response is streamed. Default: False

**Returns**

A JSON with the contents of a given dataset

**Return type**

dict

**list**(*name\_pattern: str, return\_attributes: bool = False*) → DatasetListResponse

Retrieve a list of datasets based on a given pattern.

**Parameters**

- **name\_pattern** (*str*) – The pattern to match dataset names.
- **return\_attributes** (*bool*) – Whether to return dataset attributes along with the names. Defaults to False.

**Returns**

A JSON with a list of dataset names (and attributes if specified) matching the given pattern.

**Return type**

DatasetListResponse

**list\_members**(*dataset\_name: str, member\_pattern: str | None = None, member\_start: str | None = None, limit: int = 1000, attributes: str = 'member'*) → MemberListResponse

Retrieve the list of members on a given PDS/PDSE.

**Parameters**

- **dataset\_name** (*str*) – Name of the dataset
- **member\_pattern** (*Optional[str]*) – Filters members by name pattern
- **member\_start** (*Optional[str]*) – The starting point for listing members
- **limit** (*int*) – The maximum number of members returned
- **attributes** (*str*) – The member attributes to retrieve

**Returns**

A JSON with a list of members from a given PDS/PDSE

**Return type**

MemberListResponse

**migrate**(*dataset\_name: str, wait: bool = False*) → dict

Migrate the data set.

**Parameters**

- **dataset\_name** (*str*) – Name of the data set
- **wait** (*bool*) – If true, the function waits for completion of the request, otherwise the request is queued.

**Returns**

A JSON containing the result of the operation

**Return type**

dict

**recall\_migrated**(*dataset\_name: str, wait: bool = False*) → dict

Recall a migrated data set.

**Parameters**

- **dataset\_name** (*str*) – Name of the data set
- **wait** (*bool*) – If true, the function waits for completion of the request, otherwise the request is queued

**Returns**

A JSON containing the result of the operation

**Return type**

dict

**rename**(*before\_dataset\_name: str, after\_dataset\_name: str*) → dict

Rename the data set.

**Parameters**

- **before\_dataset\_name** (*str*) – The source data set name.
- **after\_dataset\_name** (*str*) – New name for the source data set.

**Returns**

A JSON containing the result of the operation

**Return type**

dict

**rename\_member**(*dataset\_name: str, before\_member\_name: str, after\_member\_name: str, enq: str = ""*) → dict

Rename the data set member.

**Parameters**

- **dataset\_name** (*str*) – Name of the data set.
- **before\_member\_name** (*str*) – The source member name.
- **after\_member\_name** (*str*) – New name for the source member.
- **enq** (*str*) – Values can be SHRW or EXCLU. SHRW is the default for PDS members, EXCLU otherwise.

**Raises****ValueError** – Thrown when a parameter is invalid**Returns**

A JSON containing the result of the operation

**Return type**

dict

**upload\_file**(*input\_file: str, dataset\_name: str, encoding: str = 'utf-8', binary: bool = False*) → None

Upload contents of a given file and uploads it to a dataset.

**Parameters**

- **input\_file** (*str*) – Name of the file to be uploaded

- **dataset\_name** (*str*) – Name of the dataset to be created
- **encoding** (*str*) – Specifies the encoding name (e.g. IBM-1047)
- **binary** (*bool*) – specifies whether the file is binary

**Raises**

**FileNotFound** – Thrown when a file is not found at provided location

**write**(*dataset\_name: str, data: str, encoding: str = 'utf-8'*) → dict

Write content to an existing dataset.

**Parameters**

- **dataset\_name** (*str*) – Name of the dataset to retrieve
- **data** (*str*) – Content to be written
- **encoding** (*str*) – Specifies encoding name (e.g. IBM-1047)

**Returns**

A JSON containing the result of the operation

**Return type**

dict

## Exceptions classes

### InvalidPermsOption

**class** `zos_files.zowe.zos_files_for_zowe_sdk.exceptions.InvalidPermsOption`(*value: int*)

Class used to represent an invalid permission value.

**Parameters**

**value** (*int*) – The value of the permission option

### MaxAllocationQuantityExceeded

**class** `zos_files.zowe.zos_files_for_zowe_sdk.exceptions.MaxAllocationQuantityExceeded`

Class used to represent an invalid allocation quantity.

## 3.4 z/OS Jobs

Contains the z/OSMF Jobs REST API functionalities.



### 3.4.1 Installing

```
pip install zowe.zos_jobs_for_zowe_sdk
```

### 3.4.2 Reference

#### zowe.zos\_jobs\_for\_zowe\_sdk

#### Jobs

**class** `zos_jobs.zowe.zos_jobs_for_zowe_sdk.Jobs`(*connection: dict, log: bool = True*)

Class used to represent the base z/OSMF Jobs API.

It includes all operations related to datasets.

#### Parameters

- **connection** (*dict*) – A profile for connection in dict (json) format
- **log** (*bool*) – Flag to disable logger

**cancel\_job**(*jobname: str, jobid: str, modify\_version: str = '2.0'*) → `StatusResponse`

Cancel a job.

#### Parameters

- **jobname** (*str*) – The name of the job
- **jobid** (*str*) – The job id on JES
- **modify\_version** (*str*) – Default (“2.0”) specifies that the request is to be processed synchronously. For asynchronous processing - change the value to “1.0”

#### Raises

**ValueError** – Thrown if the `modify_version` is invalid

#### Returns

A JSON object containing the result of the request execution

#### Return type

`StatusResponse`

**change\_job\_class**(*jobname: str, jobid: str, class\_name: str, modify\_version: str = '2.0'*) →

`StatusResponse`

Change the job class.

#### Parameters

- **jobname** (*str*) – The name of the job
- **jobid** (*str*) – The job id on JES
- **class\_name** (*str*) – The name of class to be set to
- **modify\_version** (*str*) – Default (“2.0”) specifies that the request is to be processed synchronously. For asynchronous processing - change the value to “1.0”

#### Raises

**ValueError** – Thrown if the `modify_version` is invalid

**Returns**

A JSON object containing the result of the request execution

**Return type**

StatusResponse

**delete\_job**(*jobname: str, jobid: str, modify\_version: str = '2.0'*) → StatusResponse

Delete the given job on JES.

**Parameters**

- **jobname** (*str*) – The name of the job
- **jobid** (*str*) – The job id on JES
- **modify\_version** (*str*) – Default (“2.0”) specifies that the request is to be processed synchronously. For asynchronous processing - change the value to “1.0”

**Raises**

**ValueError** – Thrown if the modify\_version is invalid

**Returns**

A JSON object containing the result of the request execution

**Return type**

StatusResponse

**get\_jcl\_text**(*correlator: str*) → str

Retrieve the input JCL text for job with specified correlator.

**Parameters**

**correlator** (*str*) – The correlator of the job. This is the value of the key ‘job-correlator’ in the status json

**Returns**

A str object containing the result of the request execution

**Return type**

str

**get\_job\_output\_as\_files**(*status: dict, output\_dir: str*)

Get all spool files and submitted jcl text in separate files in the specified output directory.

The structure will be as follows: –<output directory> | file: jcl.txt | dir: jobname

dir: jobid | dir: stepname | file: spool file <nn> ...

**Parameters**

- **status** (*dict*) – The response json describing the job to be used. (i.e. from the last get\_status call)
- **output\_dir** (*str*) – The output directory where the output files will be stored. The directory does not have to exist yet

**get\_job\_status**(*jobname: str, jobid: str*) → JobResponse

Retrieve the status of a given job on JES.

**Parameters**

- **jobname** (*str*) – The name of the job
- **jobid** (*str*) – The job id on JES

**Returns**

A JSON object containing the status of the job on JES

**Return type**

JobResponse

**get\_spool\_file\_contents**(*correlator: str, id: str*) → str

Retrieve the contents of a single spool file from a job.

**Parameters**

- **correlator** (*str*) – The correlator of the job. This is the value of the key ‘job-correlator’ in the status json
- **id** (*str*) – The id number of the spool file. This is returned in the get\_spool\_files return json

**Returns**

The contents of the spool file

**Return type**

str

**get\_spool\_files**(*correlator: str*) → List[SpoolResponse]

Retrieve the spool files for a job identified by the correlator.

**Parameters**

**correlator** (*str*) – The correlator of the job. This is the value of the key ‘job-correlator’ in the status json

**Returns**

A JSON object containing the result of the request execution

**Return type**

List[SpoolResponse]

**hold\_job**(*jobname: str, jobid: str, modify\_version: str = '2.0'*) → StatusResponse

Hold the given job on JES.

**Parameters**

- **jobname** (*str*) – The name of the job
- **jobid** (*str*) – The job id on JES
- **modify\_version** (*str*) – Default (“2.0”) specifies that the request is to be processed synchronously. For asynchronous processing - change the value to “1.0”

**Raises**

**ValueError** – Thrown if the modify\_version is invalid

**Returns**

A JSON object containing the result of the request execution

**Return type**

StatusResponse

**list\_jobs**(*owner: str | None = None, prefix: str = '\*', max\_jobs: int = 1000, user\_correlator: str | None = None*) → List[JobResponse]

Retrieve list of jobs on JES based on the provided arguments.

**Parameters**

- **owner** (*Optional[str]*) – The job owner (default is zosmf user)

- **prefix** (*str*) – The job name prefix (default is \*)
- **max\_jobs** (*int*) – The maximum number of jobs in the output (default is 1000)
- **user\_correlator** (*Optional [str]*) – The z/OSMF user correlator attribute (default is None)

**Returns**

A list of jobs on JES queue based on the given parameters

**Return type**

List[JobResponse]

**release\_job**(*jobname: str, jobid: str, modify\_version: str = '2.0'*) → StatusResponse

Release the given job on JES.

**Parameters**

- **jobname** (*str*) – The name of the job
- **jobid** (*str*) – The job id on JES
- **modify\_version** (*str*) – Default (“2.0”) specifies that the request is to be processed synchronously. For asynchronous processing - change the value to “1.0”

**Raises**

**ValueError** – Thrown if the modify\_version is invalid

**Returns**

A JSON object containing the result of the request execution

**Return type**

StatusResponse

**submit\_from\_local\_file**(*jcl\_path: str*) → JobResponse

Submit a job from local file.

This function will internally call the *submit\_plaintext* function in order to submit the contents of the given input file

**Parameters**

**jcl\_path** (*str*) – Path to the local file where the JCL is located

**Raises**

**FileNotFoundError** – If the local file provided is not found

**Returns**

A JSON object containing the result of the request execution

**Return type**

JobResponse

**submit\_from\_mainframe**(*jcl\_path: str*) → JobResponse

Submit a job from a given dataset.

**Parameters**

**jcl\_path** (*str*) – The dataset where the JCL is located

**Returns**

A JSON object containing the result of the request execution

**Return type**

JobResponse

**submit\_plaintext**(*jcl: str*) → JobResponse

Submit a job from plain text input.

**Parameters**

**jcl** (*str*) – The plain text JCL to be submitted

**Returns**

A JSON object containing the result of the request execution

**Return type**

JobResponse

## 3.5 z/OS TSO

Contains the z/OSMF TSO REST API functionalities.

### 3.5.1 Installing

```
pip install zowe.zos_tso_for_zowe_sdk
```

### 3.5.2 Reference

**zowe.zos\_tso\_for\_zowe\_sdk**

**Tso**

```
class zos_tso.zowe.zos_tso_for_zowe_sdk.Tso(connection: dict, tso_profile: dict | None = None, log: bool = True)
```

Class used to represent the base z/OSMF TSO API.

**Parameters**

- **connection** (*dict*) – Connection object
- **tso\_profile** (*Optional[dict]*) – Profile used for tso connection
- **log** (*bool*) – Flag to disable logger

**end**(*session\_key: str*) → EndResponse

Terminates an existing TSO session.

**Parameters**

**session\_key** (*str*) – The session key of an existing TSO session

**Returns**

A string informing if the session was terminated successfully or not

**Return type**

EndResponse

**end\_tso\_session**(*session\_key: str*) → str

Terminates an existing TSO session.

**Parameters**

**session\_key** (*str*) – The session key of an existing TSO session

**Returns**

A string informing if the session was terminated successfully or not

**Return type**

str

**issue\_command**(*command: str*) → IssueResponse

Issue a TSO command.

This function will first initiate a TSO session, retrieve the session key, send the command and finally terminate the session

**Parameters**

**command** (*str*) – TSO command to be executed

**Returns**

A list containing the output from the TSO command

**Return type**

IssueResponse

**parse\_message\_ids**(*response\_json: dict*) → list

Parse TSO response and retrieve only the message ids.

**Parameters**

**response\_json** (*dict*) – The JSON containing the TSO response

**Returns**

A list containing the TSO response message ids

**Return type**

list

**ping\_tso\_session**(*session\_key: str*) → str

Ping an existing TSO session and returns if it is still available.

**Parameters**

**session\_key** (*str*) – The session key of an existing TSO session

**Returns**

A string informing if the ping was successful or not. Where the options are: 'Ping successful' or 'Ping failed'

**Return type**

str

**retrieve\_tso\_messages**(*response\_json: dict*) → list

Parse the TSO response and retrieve all messages.

**Parameters**

**response\_json** (*dict*) – The JSON containing the TSO response

**Returns**

A list containing the TSO response messages

**Return type**

list

**send**(*session\_key: str, message: str*) → SendResponse

Send a command to an existing TSO session.

**Parameters**

- **session\_key** (*str*) – The session key of an existing TSO session
- **message** (*str*) – The message/command to be sent to the TSO session

**Returns**

A non-normalized list from TSO containing the result from the command

**Return type**

SendResponse

**send\_tso\_message**(*session\_key: str, message: str*) → list

Send a command to an existing TSO session.

**Parameters**

- **session\_key** (*str*) – The session key of an existing TSO session
- **message** (*str*) – The message/command to be sent to the TSO session

**Returns**

A non-normalized list from TSO containing the result from the command

**Return type**

list

**start**(*proc: str | None = None, chset: str | None = None, cpage: str | None = None, rows: str | None = None, cols: str | None = None, rsize: str | None = None, acct: str | None = None*) → StartResponse

Start a TSO session.

**Parameters**

- **proc** (*Optional[str]*) – Proc parameter for the TSO session (default is “IZUFPROC”)
- **chset** (*Optional[str]*) – Chset parameter for the TSO session (default is “697”)
- **cpage** (*Optional[str]*) – Cpage parameter for the TSO session (default is “1047”)
- **rows** (*Optional[str]*) – Rows parameter for the TSO session (default is “204”)
- **cols** (*Optional[str]*) – Cols parameter for the TSO session (default is “160”)
- **rsize** (*Optional[str]*) – Rsize parameter for the TSO session (default is “4096”)
- **acct** (*Optional[str]*) – Acct parameter for the TSO session (default is “DEFAULT”)

**Returns**

The ‘servletKey’ key for the created session (if successful)

**Return type**

StartResponse

**start\_tso\_session**(*proc: str | None = None, chset: str | None = None, cpage: str | None = None, rows: str | None = None, cols: str | None = None, rsize: str | None = None, acct: str | None = None*) → str

Start a TSO session.

**Parameters**

- **proc** (*Optional[str]*) – Proc parameter for the TSO session (default is “IZUFPROC”)
- **chset** (*Optional[str]*) – Chset parameter for the TSO session (default is “697”)
- **cpage** (*Optional[str]*) – Cpage parameter for the TSO session (default is “1047”)
- **rows** (*Optional[str]*) – Rows parameter for the TSO session (default is “204”)
- **cols** (*Optional[str]*) – Cols parameter for the TSO session (default is “160”)

- **rsize** (*Optional[str]*) – Rsize parameter for the TSO session (default is “4096”)
- **acct** (*Optional[str]*) – Acct parameter for the TSO session (default is “DEFAULT”)

**Returns**

The ‘servletKey’ key for the created session (if successful)

**Return type**

str

## 3.6 z/OSMF

Contains the z/OSMF REST API functionalities.

### 3.6.1 Installing

```
pip install zowe.zosmf_for_zowe_sdk
```

### 3.6.2 Reference

#### zowe.zosmf\_for\_zowe\_sdk

#### Zosmf

**class** zosmf.zowe.zosmf\_for\_zowe\_sdk.**Zosmf**(*connection: dict, log: bool = True*)

Class used to represent the base z/OSMF API.

**Parameters**

- **connection** (*dict*) – The z/OSMF connection object (generated by the ZoweSDK object)
- **log** (*bool*) – Flag to disable logger

**get\_info()** → ZosmfResponse

Return a JSON response from the GET request to z/OSMF info endpoint.

**Returns**

A JSON containing the z/OSMF Info REST API data

**Return type**

ZosmfResponse

**list\_systems()** → ZosmfResponse

Return a JSON response from the GET request to z/OSMF info endpoint.

**Returns**

Return a list of the systems that are defined to a z/OSMF instance

**Return type**

ZosmfResponse



**CLASSES**

**4.1 zowe.secrets\_for\_zowe\_sdk**



## CONTRIBUTING

This document contains the contribution guidelines for the Zowe Client Python SDK.

Notice that the Python SDK is still in early development stages, meaning that major architectural changes might be made by the development team at any given moment. **For this reason, only bug fixes and documentation changes are being accepted as contribution at this moment** . This document will be updated as soon as the stable release of the SDK is published (v1.0.0)

- *Git branching model*
- *Pull requests*
- *Running Tests*
- *Building Docs*
- *Code standards*

### 5.1 Git branching model

This project follows the [Git flow](#) branching model.

### 5.2 Pull requests

Consider the following when you interact with pull requests:

- Pull request reviewers should be assigned to a same-team member.
- Pull requests should remain open for at least 24 hours, or until close of the business next business day (accounting for weekends and holidays).
- Anyone can comment on a pull request to request delay on merging or to get questions answered.

## 5.3 Running Tests

The project's test suite can be run with the python test runner, *pytest*.

All test and regular dependencies are included here:

```
pip install -r requirements.txt
```

Commands for running all unit/integration tests from their respective folder:

```
pytest tests/unit
```

```
pytest tests/integration
```

More information on *pytest*'s usage can be found [here](#).

In order to run integration tests, you will need to have a mainframe account and team profile configuration files properly set up.

Information on creating team profile configuration files can be found [here](#).

You will also need to update the `zowe.config.json` file with the necessary information.

## 5.4 Building Docs

The project's documentation is published on [ReadTheDocs.io](#).

To build the docs from source locally, you need these prerequisites:

- [Sphinx](#) - Python Documentation Generator
- Python packages - `pip install -r docs/requirements.txt`

Run `make html` in the docs directory to generate HTML files in “docs/build/html” that can be previewed in your browser.

Docs are generated from reStructuredText (.rst) files in “docs/source” and Python docstrings in the source code which also use reST markup. A quick reference for reStructuredText markup can be found [here](#).

## 5.5 Code standards

This project follows the [PEP 8](#) style guide.

# INDEX

## Symbols

- `__del__()` (`core.zowe.core_for_zowe_sdk.RequestHandler` method), 13
  - `__enter__()` (`core.zowe.core_for_zowe_sdk.SdkApi` method), 14
  - `__exit__()` (`core.zowe.core_for_zowe_sdk.SdkApi` method), 14
  - `__str__()` (`core.zowe.core_for_zowe_sdk.custom_warnings.ConfigNotFoundWarning` method), 18
  - `__str__()` (`core.zowe.core_for_zowe_sdk.custom_warnings.ProfileNotFoundWarning` method), 18
  - `__str__()` (`core.zowe.core_for_zowe_sdk.custom_warnings.ProfileParsingWarning` method), 18
  - `__str__()` (`core.zowe.core_for_zowe_sdk.custom_warnings.SecurePropertyNotFoundWarning` method), 19
- A**
- `alcunit` (`zos_files.zowe.zos_files_for_zowe_sdk.datasets.DatasetOption` property), 30
  - `ApiConnection` (class in `core.zowe.core_for_zowe_sdk`), 7
  - `autodiscover_config_dir()` (`core.zowe.core_for_zowe_sdk.config_file.ConfigFile` method), 15
  - `avgblk` (`zos_files.zowe.zos_files_for_zowe_sdk.datasets.DatasetOption` property), 30
- B**
- `blksize` (`zos_files.zowe.zos_files_for_zowe_sdk.datasets.DatasetOption` property), 30
- C**
- `cancel_job()` (`zos_jobs.zowe.zos_jobs_for_zowe_sdk.Jobs` method), 37
  - `change_job_class()` (`zos_jobs.zowe.zos_jobs_for_zowe_sdk.Jobs` method), 37
  - `close()` (`core.zowe.core_for_zowe_sdk.Log` static method), 8
  - `close_all()` (`core.zowe.core_for_zowe_sdk.Log` static method), 9
  - `close_console_output()` (`core.zowe.core_for_zowe_sdk.Log` static method), 9
  - `close_file_output()` (`core.zowe.core_for_zowe_sdk.Log` static method), 9
  - `config_appname` (`core.zowe.core_for_zowe_sdk.ProfileManager` property), 10
  - `config_dir` (`core.zowe.core_for_zowe_sdk.ProfileManager` property), 10
  - `config_filename` (`core.zowe.core_for_zowe_sdk.ProfileManager` property), 10
  - `config_filepath` (`core.zowe.core_for_zowe_sdk.ProfileManager` property), 10
  - `ConfigFile` (class in `core.zowe.core_for_zowe_sdk.config_file`), 14
  - `ConfigNotFoundWarning` (class in `core.zowe.core_for_zowe_sdk.custom_warnings`), 18
  - `Console` (class in `zos_console.zowe.zos_console_for_zowe_sdk`), 22
  - `console_handler` (`core.zowe.core_for_zowe_sdk.Log` attribute), 8
  - `console_output` (`core.zowe.core_for_zowe_sdk.Log` attribute), 8
  - `copy_data_set_or_member()` (`zos_files.zowe.zos_files_for_zowe_sdk.datasets.Datasets` method), 31
  - `copy_data_set_or_member()` (`zos_files.zowe.zos_files_for_zowe_sdk.Files` method), 25
  - `copy_uss_to_data_set()` (`zos_files.zowe.zos_files_for_zowe_sdk.datasets.Datasets` method), 31
  - `copy_uss_to_data_set()` (`zos_files.zowe.zos_files_for_zowe_sdk.Files` method), 25
  - `create()` (`zos_files.zowe.zos_files_for_zowe_sdk.datasets.Datasets` method), 32
  - `create()` (`zos_files.zowe.zos_files_for_zowe_sdk.FileSystems` method), 23
  - `create()` (`zos_files.zowe.zos_files_for_zowe_sdk.USSFiles` method), 23

*method*), 27  
 create\_data\_set() (*zos\_files.zowe.zos\_files\_for\_zowe\_sdk.Files* *method*), 33  
*method*), 25  
 create\_default() (*zos\_files.zowe.zos\_files\_for\_zowe\_sdk.datasets.Datasets* *method*), 32  
 create\_default\_data\_set() (*zos\_files.zowe.zos\_files\_for\_zowe\_sdk.Files* *method*), 25  
 create\_uss() (*zos\_files.zowe.zos\_files\_for\_zowe\_sdk.Files* *method*), 25  
 create\_zfs\_file\_system() (*zos\_files.zowe.zos\_files\_for\_zowe\_sdk.Files* *method*), 25  
 CredentialManager (class in *core.zowe.core\_for\_zowe\_sdk*), 8  
**D**  
 data (*core.zowe.core\_for\_zowe\_sdk.config\_file.Profile* *attribute*), 17  
 dataclass (*zos\_files.zowe.zos\_files\_for\_zowe\_sdk.datasets.DatasetOption* *property*), 30  
 DatasetOption (class in *zos\_files.zowe.zos\_files\_for\_zowe\_sdk.datasets*), 29  
 Datasets (class in *zos\_files.zowe.zos\_files\_for\_zowe\_sdk.datasets*), 31  
 delete() (*zos\_files.zowe.zos\_files\_for\_zowe\_sdk.datasets.Datasets* *tribute*), 8  
*method*), 32  
 delete() (*zos\_files.zowe.zos\_files\_for\_zowe\_sdk.FileSystems* *method*), 23  
 delete() (*zos\_files.zowe.zos\_files\_for\_zowe\_sdk.USSFiles* *method*), 27  
 delete\_data\_set() (*zos\_files.zowe.zos\_files\_for\_zowe\_sdk.Files* *method*), 25  
 delete\_job() (*zos\_jobs.zowe.zos\_jobs\_for\_zowe\_sdk.Jobs* *method*), 38  
 delete\_migrated() (*zos\_files.zowe.zos\_files\_for\_zowe\_sdk.datasets.Datasets* *method*), 33  
 delete\_migrated\_data\_set() (*zos\_files.zowe.zos\_files\_for\_zowe\_sdk.Files* *method*), 25  
 delete\_uss() (*zos\_files.zowe.zos\_files\_for\_zowe\_sdk.Files* *method*), 25  
 delete\_zfs\_file\_system() (*zos\_files.zowe.zos\_files\_for\_zowe\_sdk.Files* *method*), 25  
 dirblk (*zos\_files.zowe.zos\_files\_for\_zowe\_sdk.datasets.DatasetOption* *property*), 30  
 dirname (*core.zowe.core\_for\_zowe\_sdk.Log* *attribute*), 8  
 download() (*zos\_files.zowe.zos\_files\_for\_zowe\_sdk.datasets.Datasets* *method*), 33  
 download() (*zos\_files.zowe.zos\_files\_for\_zowe\_sdk.USSFiles* *method*), 27  
 download\_binary() (*zos\_files.zowe.zos\_files\_for\_zowe\_sdk.datasets.Datasets* *method*), 33  
 download\_binary\_dsn() (*zos\_files.zowe.zos\_files\_for\_zowe\_sdk.Files* *method*), 26  
 download\_dsn() (*zos\_files.zowe.zos\_files\_for\_zowe\_sdk.Files* *method*), 26  
 download\_uss() (*zos\_files.zowe.zos\_files\_for\_zowe\_sdk.Files* *method*), 26  
 ds (*zos\_files.zowe.zos\_files\_for\_zowe\_sdk.Files* *attribute*), 25  
 dsntype (*zos\_files.zowe.zos\_files\_for\_zowe\_sdk.datasets.DatasetOption* *property*), 30  
 dsorg (*zos\_files.zowe.zos\_files\_for\_zowe\_sdk.datasets.DatasetOption* *property*), 30  
**E**  
 end() (*zos\_tso.zowe.zos\_tso\_for\_zowe\_sdk.Tso* *method*), 41  
 end\_session() (*zos\_tso.zowe.zos\_tso\_for\_zowe\_sdk.Tso* *method*), 41  
**F**  
 file\_handler (*core.zowe.core\_for\_zowe\_sdk.Log* *tribute*), 8  
 file\_output (*core.zowe.core\_for\_zowe\_sdk.Log* *tribute*), 8  
 FileNotFoundError (class in *core.zowe.core\_for\_zowe\_sdk.exceptions*), 19  
 Files (class in *zos\_files.zowe.zos\_files\_for\_zowe\_sdk*), 25  
 FileSystems (class in *zos\_files.zowe.zos\_files\_for\_zowe\_sdk*), 23  
 find\_profile() (*core.zowe.core\_for\_zowe\_sdk.config\_file.ConfigFile* *method*), 15  
 is (*zos\_files.zowe.zos\_files\_for\_zowe\_sdk.Files* *tribute*), 25  
**G**  
 get\_binary\_content() (*zos\_files.zowe.zos\_files\_for\_zowe\_sdk.datasets.Datasets* *method*), 33  
 get\_content() (*zos\_files.zowe.zos\_files\_for\_zowe\_sdk.datasets.Datasets* *method*), 33  
 get\_content() (*zos\_files.zowe.zos\_files\_for\_zowe\_sdk.USSFiles* *method*), 28  
 get\_content\_streamed() (*zos\_files.zowe.zos\_files\_for\_zowe\_sdk.USSFiles* *method*), 28  
 get\_dsn\_binary\_content() (*zos\_files.zowe.zos\_files\_for\_zowe\_sdk.Files* *method*), 26

`get_dsn_binary_content_streamed()`  
 (`zos_files.zowe.zos_files_for_zowe_sdk.Files`  
 method), 26

`get_dsn_content()` (`zos_files.zowe.zos_files_for_zowe_sdk.Files`  
 method), 26

`get_dsn_content_streamed()`  
 (`zos_files.zowe.zos_files_for_zowe_sdk.Files`  
 method), 26

`get_env()` (`core.zowe.core_for_zowe_sdk.ProfileManager`  
 static method), 10

`get_file_content()` (`zos_files.zowe.zos_files_for_zowe_sdk.Files`  
 method), 26

`get_file_content_streamed()`  
 (`zos_files.zowe.zos_files_for_zowe_sdk.Files`  
 method), 26

`get_highest_priority_layer()`  
 (`core.zowe.core_for_zowe_sdk.ProfileManager`  
 method), 11

`get_info()` (`zosmf.zowe.zosmf_for_zowe_sdk.Zosmf`  
 method), 44

`get_jcl_text()` (`zos_jobs.zowe.zos_jobs_for_zowe_sdk.Jobs`  
 method), 38

`get_job_output_as_files()`  
 (`zos_jobs.zowe.zos_jobs_for_zowe_sdk.Jobs`  
 method), 38

`get_job_status()` (`zos_jobs.zowe.zos_jobs_for_zowe_sdk.Jobs`  
 method), 38

`get_profile()` (`core.zowe.core_for_zowe_sdk.config_file.ConfigFile`  
 method), 15

`get_profile()` (`core.zowe.core_for_zowe_sdk.ProfileManager`  
 static method), 11

`get_profile_name_from_path()`  
 (`core.zowe.core_for_zowe_sdk.config_file.ConfigFile`  
 method), 15

`get_profile_path_from_name()`  
 (`core.zowe.core_for_zowe_sdk.config_file.ConfigFile`  
 method), 16

`get_profilename_from_profiletype()`  
 (`core.zowe.core_for_zowe_sdk.config_file.ConfigFile`  
 method), 16

`get_response()` (`zos_console.zowe.zos_console_for_zowe_sdk.Console`  
 method), 22

`get_spool_file_contents()`  
 (`zos_jobs.zowe.zos_jobs_for_zowe_sdk.Jobs`  
 method), 39

`get_spool_files()` (`zos_jobs.zowe.zos_jobs_for_zowe_sdk.Jobs`  
 method), 39

**H**

`hold_job()` (`zos_jobs.zowe.zos_jobs_for_zowe_sdk.Jobs`  
 method), 39

`host_url` (`core.zowe.core_for_zowe_sdk.session.Session`  
 property), 21

**I**

`init_from_file()` (`core.zowe.core_for_zowe_sdk.config_file.ConfigFile`  
 method), 16

`InvalidPermsOption` (class in  
`zos_files.zowe.zos_files_for_zowe_sdk.exceptions`),  
 36

`InvalidRequestMethod` (class in  
`core.zowe.core_for_zowe_sdk.exceptions`),  
 19

`ISession` (class in `core.zowe.core_for_zowe_sdk.session`),  
 21

`issue_command()` (`zos_console.zowe.zos_console_for_zowe_sdk.Console`  
 method), 22

`issue_command()` (`zos_tso.zowe.zos_tso_for_zowe_sdk.Tso`  
 method), 42

**J**

`Jobs` (class in `zos_jobs.zowe.zos_jobs_for_zowe_sdk`), 37

**L**

`like` (`zos_files.zowe.zos_files_for_zowe_sdk.datasets.DatasetOption`  
 property), 30

`list()` (`zos_files.zowe.zos_files_for_zowe_sdk.datasets.Datasets`  
 method), 34

`list()` (`zos_files.zowe.zos_files_for_zowe_sdk.FileSystems`  
 method), 24

`list()` (`zos_files.zowe.zos_files_for_zowe_sdk.USSFiles`  
 method), 28

`list_dsn()` (`zos_files.zowe.zos_files_for_zowe_sdk.Files`  
 method), 26

`list_dsn_members()` (`zos_files.zowe.zos_files_for_zowe_sdk.Files`  
 method), 26

`list_files()` (`zos_files.zowe.zos_files_for_zowe_sdk.Files`  
 method), 26

`list_jobs()` (`zos_jobs.zowe.zos_jobs_for_zowe_sdk.Jobs`  
 method), 39

`list_members()` (`zos_files.zowe.zos_files_for_zowe_sdk.datasets.Datasets`  
 method), 34

`list_systems()` (`zosmf.zowe.zosmf_for_zowe_sdk.Zosmf`  
 method), 44

`list_unix_file_systems()`  
 (`zos_files.zowe.zos_files_for_zowe_sdk.Files`  
 method), 26

`load()` (`core.zowe.core_for_zowe_sdk.ProfileManager`  
 method), 11

`load()` (`core.zowe.core_for_zowe_sdk.session.Session`  
 method), 21

`load()` (`core.zowe.core_for_zowe_sdk.ZosmfProfile`  
 method), 14

`load_profile_properties()`  
 (`core.zowe.core_for_zowe_sdk.config_file.ConfigFile`  
 method), 16

load\_secure\_props() (core.zowe.core\_for\_zowe\_sdk.CredentialManager static method), 8  
 primary (zos\_files.zowe.zos\_files\_for\_zowe\_sdk.datasets.DatasetOption property), 30  
 Profile (class in core.zowe.core\_for\_zowe\_sdk.config\_file), 17  
 Log (class in core.zowe.core\_for\_zowe\_sdk), 8  
 loggers (core.zowe.core\_for\_zowe\_sdk.Log attribute), 8  
 ProfileManager (class in core.zowe.core\_for\_zowe\_sdk), 10  
 lrecl (zos\_files.zowe.zos\_files\_for\_zowe\_sdk.datasets.DatasetOption property), 30  
 ProfileNotFound (class in core.zowe.core\_for\_zowe\_sdk.exceptions), 19  
**M**  
 MaxAllocationQuantityExceeded (class in zos\_files.zowe.zos\_files\_for\_zowe\_sdk.exceptions), 36  
 ProfileNotFoundWarning (class in core.zowe.core\_for\_zowe\_sdk.custom\_warnings), 18  
 mgmtclass (zos\_files.zowe.zos\_files\_for\_zowe\_sdk.datasets.DatasetOption property), 30  
 ProfileParsingWarning (class in core.zowe.core\_for\_zowe\_sdk.custom\_warnings), 18  
 migrate() (zos\_files.zowe.zos\_files\_for\_zowe\_sdk.datasets.Datasets method), 34  
 profiles\_dir (core.zowe.core\_for\_zowe\_sdk.ZosmfProfile property), 14  
 migrate\_data\_set() (zos\_files.zowe.zos\_files\_for\_zowe\_sdk.Files method), 26  
 missing\_secure\_props (core.zowe.core\_for\_zowe\_sdk.config\_file.Profile attribute), 17  
**R**  
 MissingConnectionArgs (class in core.zowe.core\_for\_zowe\_sdk.exceptions), 19  
 recall\_migrated() (zos\_files.zowe.zos\_files\_for\_zowe\_sdk.datasets.Datasets method), 35  
 recall\_migrated\_data\_set() (zos\_files.zowe.zos\_files\_for\_zowe\_sdk.Files method), 26  
 mount() (zos\_files.zowe.zos\_files\_for\_zowe\_sdk.FileSystem method), 24  
 recfm (zos\_files.zowe.zos\_files\_for\_zowe\_sdk.datasets.DatasetOption property), 30  
 mount\_file\_system() (zos\_files.zowe.zos\_files\_for\_zowe\_sdk.Files method), 26  
 register\_logger() (core.zowe.core\_for\_zowe\_sdk.Log static method), 9  
 release\_job() (zos\_jobs.zowe.zos\_jobs\_for\_zowe\_sdk.Jobs method), 40  
**N**  
 name (core.zowe.core\_for\_zowe\_sdk.config\_file.Profile attribute), 17  
 rename() (zos\_files.zowe.zos\_files\_for\_zowe\_sdk.datasets.Datasets method), 35  
 rename\_data\_set() (zos\_files.zowe.zos\_files\_for\_zowe\_sdk.Files method), 26  
 rename\_data\_set\_member() (zos\_files.zowe.zos\_files\_for\_zowe\_sdk.Files method), 26  
**O**  
 open() (core.zowe.core\_for\_zowe\_sdk.Log static method), 9  
 rename\_member() (zos\_files.zowe.zos\_files\_for\_zowe\_sdk.datasets.Datasets method), 35  
 open\_all() (core.zowe.core\_for\_zowe\_sdk.Log static method), 9  
 RequestFailed (class in core.zowe.core\_for\_zowe\_sdk.exceptions), 20  
 open\_console\_output() (core.zowe.core\_for\_zowe\_sdk.Log static method), 9  
 RequestHandler (class in core.zowe.core\_for\_zowe\_sdk), 13  
 open\_file\_output() (core.zowe.core\_for\_zowe\_sdk.Log static method), 9  
 retrieve\_tso\_messages() (zos\_tso.zowe.zos\_tso\_for\_zowe\_sdk.Tso method), 42  
**P**  
 parse\_message\_ids() (zos\_tso.zowe.zos\_tso\_for\_zowe\_sdk.Tso method), 42  
**S**  
 perform\_request() (core.zowe.core\_for\_zowe\_sdk.RequestHandler method), 13  
 save() (core.zowe.core\_for\_zowe\_sdk.config\_file.ConfigFile method), 17  
 ping\_tso\_session() (zos\_tso.zowe.zos\_tso\_for\_zowe\_sdk.Tso method), 42  
 save() (core.zowe.core\_for\_zowe\_sdk.ProfileManager method), 12



save\_secure\_props() (core.zowe.core\_for\_zowe\_sdk.CredentialManager static method), 8

schema\_list() (core.zowe.core\_for\_zowe\_sdk.config\_file.ConfigFile method), 17

SdkApi (class in core.zowe.core\_for\_zowe\_sdk), 13

secondary (zos\_files.zowe.zos\_files\_for\_zowe\_sdk.datasets.DatasetOption property), 30

SecureProfileLoadFailed (class in core.zowe.core\_for\_zowe\_sdk.exceptions), 20

SecurePropsNotFoundWarning (class in core.zowe.core\_for\_zowe\_sdk.custom\_warnings), 19

SecureValuesNotFound (class in core.zowe.core\_for\_zowe\_sdk.exceptions), 20

send() (zos\_tso.zowe.zos\_tso\_for\_zowe\_sdk.Tso method), 42

send\_tso\_message() (zos\_tso.zowe.zos\_tso\_for\_zowe\_sdk.Tso method), 43

Session (class in core.zowe.core\_for\_zowe\_sdk.session), 21

set\_all\_logger\_level() (core.zowe.core\_for\_zowe\_sdk.Log static method), 9

set\_console\_output\_level() (core.zowe.core\_for\_zowe\_sdk.Log static method), 9

set\_file\_output\_level() (core.zowe.core\_for\_zowe\_sdk.Log static method), 9

set\_profile() (core.zowe.core\_for\_zowe\_sdk.config\_file.ConfigFile method), 17

set\_profile() (core.zowe.core\_for\_zowe\_sdk.ProfileManager method), 12

set\_property() (core.zowe.core\_for\_zowe\_sdk.config\_file.ConfigFile method), 17

set\_property() (core.zowe.core\_for\_zowe\_sdk.ProfileManager method), 12

start() (zos\_tso.zowe.zos\_tso\_for\_zowe\_sdk.Tso method), 43

start\_tso\_session() (zos\_tso.zowe.zos\_tso\_for\_zowe\_sdk.Tso method), 43

storclass (zos\_files.zowe.zos\_files\_for\_zowe\_sdk.datasets.DatasetOption property), 30

submit\_from\_local\_file() (zos\_jobs.zowe.zos\_jobs\_for\_zowe\_sdk.Jobs method), 40

submit\_from\_mainframe() (zos\_jobs.zowe.zos\_jobs\_for\_zowe\_sdk.Jobs method), 40

submit\_plaintext() (zos\_jobs.zowe.zos\_jobs\_for\_zowe\_sdk.Jobs method), 40

submit() (zos\_files.zowe.zos\_files\_for\_zowe\_sdk.datasets.DatasetOption method), 30

Tso (class in zos\_tso.zowe.zos\_tso\_for\_zowe\_sdk), 41

UnexpectedStatus (class in core.zowe.core\_for\_zowe\_sdk.exceptions), 20

unit (zos\_files.zowe.zos\_files\_for\_zowe\_sdk.datasets.DatasetOption property), 31

unmount() (zos\_files.zowe.zos\_files\_for\_zowe\_sdk.FileSystems method), 24

unmount\_file\_system() (zos\_files.zowe.zos\_files\_for\_zowe\_sdk.Files method), 26

UnsupportedAuthType (class in core.zowe.core\_for\_zowe\_sdk.exceptions), 21

upload() (zos\_files.zowe.zos\_files\_for\_zowe\_sdk.USSFiles method), 28

upload\_file() (zos\_files.zowe.zos\_files\_for\_zowe\_sdk.datasets.Datasets method), 35

upload\_file\_to\_dsn() (zos\_files.zowe.zos\_files\_for\_zowe\_sdk.Files method), 27

upload\_file\_to\_uss() (zos\_files.zowe.zos\_files\_for\_zowe\_sdk.Files method), 27

user\_config\_dir (core.zowe.core\_for\_zowe\_sdk.ProfileManager property), 12

uss (zos\_files.zowe.zos\_files\_for\_zowe\_sdk.Files attribute), 25

USSFiles (class in zos\_files.zowe.zos\_files\_for\_zowe\_sdk), 27

validate\_schema() (core.zowe.core\_for\_zowe\_sdk.config\_file.ConfigFile method), 17

volser (zos\_files.zowe.zos\_files\_for\_zowe\_sdk.datasets.DatasetOption property), 31

write() (zos\_files.zowe.zos\_files\_for\_zowe\_sdk.datasets.Datasets method), 36

write() (zos\_files.zowe.zos\_files\_for\_zowe\_sdk.USSFiles method), 28

write\_to\_dsn() (zos\_files.zowe.zos\_files\_for\_zowe\_sdk.Files method), 27

write\_to\_uss() (zos\_files.zowe.zos\_files\_for\_zowe\_sdk.Files method), 27

## Z

Zosmf (class in *zosmf.zowe.zosmf\_for\_zowe\_sdk*), [44](#)

ZosmfProfile (class in *core.zowe.core\_for\_zowe\_sdk*),

[14](#)