SalesforcexytoolsCore

Introduce

> SalesforcexytoolsCore is a python library for salesforce. It is base on simple-salesforce

- Sobject Record Management : sobject create, get, get_by_custom_id, update, delete
- Sobject Queries : query, query_more, search, query_allsearch
- Sobject Bulk action
- Run Apex Script
- Metadata Control : describeMetadata, listAllMetadata, getAllMetadataMap, listmeta, listFolder, retrieve, etc.
- Package.xml builder.
- Retrieve Metadata to memory, Retrieve Metadata zip file.
- ApexClass, Trigger, ApexComponent, ApexPage: create, update, delete, get
- Get Apex Log
- Run test class

Requirement

- Python3
- requests

Install

```
pip install SalesforcexytoolsCore
```

Document

SalesforceXyToolsCore IP Document

Download PDF

SalesforcexytoolsCore-python-library-for-salesforce(日本語).pdf

About Author : Exia.Huang

- SalesforceXyTools HP
- Github
- Twitter
- Facebook
- Qiita
- Hatenaはてなブログ

Sample Source

Sample Source

```
apexclass-create.py
apexclass-delete.py
apexclass-update.py
apexcomponent-create.py
build-package-xml.py
config.py
describe-metadata.py
download-attachments.py
get-all-metadata-map.py
get-apexlog.py
list-all-metadata.py
list-folders.py
list-metadata.py
package-type-list.py
retrieve-metadata-to-memory.py
retrieve-metadata-to-zip.py
run-apex-script.py
run-rest-api.py
run-soql-queries.py
run-testclass.py
sobject-CURD.py
trigger-create.py
visualforce-create.py
```

SalesforceXyToolsCore/Python上でApexClassを作成・更新・削除

Topic

- SalesforceXyToolsCoreを使ってApexClassを作成・更新・削除

ApexClassの作成

- Salesforce組織のユーザ名、パスワード、Apiバージョン、Product/Sandboxを設定してください。

```
from SalesforceXytoolsCore import *
import pprint

config = {
    "api_version": 42.0,
    "username": "sfdc username",
    "password": "sfdc password",
    "security_token": "",
    "is_sandbox": True
}

tooling_api = ToolingApi(username=config["username"],
                password=config["password"],
                security_token=config["security_token"],
                sandbox=config["is_sandbox"],
                version=config["api_version"]
                )

"""createApexClass"""
name = "HelloWorld"
body = """public class HelloWorld {
        private String mystr;
}"""


status_code, result = tooling_api.createApexClass(name, body)
print(status_code)
pprint.pprint(result)
```

結果確認:

```
{'errors': [], 'id': '01pxxxxxxxxxxxxxxx', 'success': True}
```

ApexClassの更新

```
from SalesforceXytoolsCore import *
import pprint

config = {
    "api_version": 42.0,
    "username": "sfdc username",
    "password": "sfdc password",
    "security_token": "",
    "is_sandbox": True
}

tooling_api = ToolingApi(username=config["username"],
                password=config["password"],
                security_token=config["security_token"],
                sandbox=config["is_sandbox"],
                version=config["api_version"]
                )

apexclass_id = "set your apex class id"
body = """public class HelloWorld {
        private String mystr1;
}"""
tooling_api.updateApexClass(apexclass_id, body)
```

ApexClassの削除

```
from SalesforceXytoolsCore import *
import pprint

config = {
    "api_version": 42.0,
    "username": "sfdc username",
    "password": "sfdc password",
    "security_token": "",
    "is_sandbox": True
}

tooling_api = ToolingApi(username=config["username"],
                password=config["password"],
                security_token=config["security_token"],
                sandbox=config["is_sandbox"],
                version=config["api_version"]
                )

apexclass_id = "set your apex class id"
body = """public class HelloWorld {
        private String mystr1;
```

```
}"""
tooling_api.updateApexClass(apexclass_id, body)
```

その他

Triggerの作成

```
"""createTrigger"""
tooling_api.createTrigger(tableEnumOrId, name, body)
```

VisualForceの作成

```
"""createApexPage"""
tooling_api.createApexPage(MasterLabel, name, markup)
```

ApexComponentの作成

```
"""createApexComponent"""
tooling_api.createApexComponent(MasterLabel, name, markup)
```

SalesforceXyToolsCore/Python上でSobjectを作成・更新・削除

Topic

- SalesforceXyToolsCoreを使ってSobjectを作成・更新・削除

IDでSobjectの取得

Salesforce組織のユーザ名、パスワード、Apiバージョン、Product/Sandboxを設定してください。

> Accountを取得する

```
from SalesforceXytoolsCore import *
import pprint

config = {
    "api_version": 42.0,
    "username": "sfdc username",
    "password": "sfdc password",
    "security_token": "",
    "is_sandbox": True
}

soap_api = Soap(username=config["username"],
                password=config["password"],
                security_token=config["security_token"],
                sandbox=config["is_sandbox"],
                version=config["api_version"]
                )

print('hello SalesforceXytoolsCore Test start')
Account = soap_api.get_sobject("Account")

"""
get a Sobject in Salesforce
"""
acc_id="set your sobject id"
account1 = Account.get(acc_id)
pprint.pprint(account1)
```

外部キーでSobjectの取得

```
"""
get a Sobject by External ID
"""
account1 = Account.get_by_custom_id('My_Custom_ID__c', 'custom_id')
pprint.pprint(account1)
```

IDでSobjectの更新

```
"""
update a Sobject in Salesforce
"""
acc_id="set your sobject id"
account1 = Account.update(acc_id,{'LastName': 'sfdc'})
pprint.pprint(account1)
```

IDでSobjectの削除

```
"""
delete a Sobject in Salesforce
"""
acc_id="set your sobject id"
Account.delete(acc_id)
```

SalesforceXyToolsCore/Python上でApexログを取得

Topic

- SalesforceXyToolsCoreを使ってApexログを取得

Apexログを取得

- Salesforce組織のユーザ名、パスワード、Apiバージョン、Product/Sandboxを設定してください。

```
from SalesforceXytoolsCore import *
import pprint

config = {
    "api_version": 42.0,
    "username": "sfdc username",
    "password": "sfdc password",
    "security_token": "",
    "is_sandbox": True
}

tooling_api = ToolingApi(username=config["username"],
                password=config["password"],
                security_token=config["security_token"],
                sandbox=config["is_sandbox"],
                version=config["api_version"]
                )

"""get apex log"""
log_id = '07LXXXXXXXXXXXXXXXXXX'
result = tooling_api.getLog(log_id)
pprint.pprint(result)
```

SalesforceXyToolsCore/Python上でApexScriptを実行する

Topic

- SalesforceXyToolsCoreを使ってSalesforceのApexScriptを実行する

ApexScriptを実行する

Salesforce組織のユーザ名、パスワード、Apiバージョン、Product/Sandboxを設定してください。

```
from SalesforceXytoolsCore import *
import pprint

config = {
    "api_version": 42.0,
    "username": "sfdc username",
    "password": "sfdc password",
    "security_token": "",
    "is_sandbox": True
}

soap_api = Soap(username=config["username"],
                password=config["password"],
                security_token=config["security_token"],
                sandbox=config["is_sandbox"],
                version=config["api_version"]
                )

"""
Run Apex Script
"""
apex_string = "System.debug('hello world');"
debug_levels = {
        "DB": "Info",
        "System": "DEBUG",
        "Workflow": "INFO",
        "Callout": "INFO",
        "Validation": "INFO",
        "Apex_Code": "DEBUG",
        "Apex_Profiling": "INFO"
    }
result = soap_api.execute_anonymous(apex_string, debug_levels)
pprint.pprint(result)
```

結果ログを確認:

```
{'debugLog': '42.0 '

'APEX_CODE,DEBUG;APEX_PROFILING,INFO;CALLOUT,INFO;DB,INFO;SYSTEM,DEBUG;VALIDATION,INFO;
WORKFLOW,INFO\n'
             "Execute Anonymous: System.debug('hello world');\n"
             '22:22:14.19 '
             '()|USER_INFO|[EXTERNAL]|xxxxxxxxxxxxxxxxxxxxxxxx|日本標準時|GMT+09:00\n'
             '22:22:14.19 ()|EXECUTION_STARTED\n'
             '22:22:14.19 '
             '()|CODE_UNIT_STARTED|[EXTERNAL]|execute_anonymous_apex\n'
             '22:22:14.19 ()|USER_DEBUG|[1]|DEBUG|hello world\n'
             '22:22:14.19 ()|CUMULATIVE_LIMIT_USAGE\n'
             '22:22:14.19 ()|LIMIT_USAGE_FOR_NS|(default)|\n'
             '  Number of SOQL queries: 0 out of 100\n'
             '  Number of query rows: 0 out of 50000\n'
             '  Number of SOSL queries: 0 out of 20\n'
             '  Number of DML statements: 0 out of 150\n'
             '  Number of DML rows: 0 out of 10000\n'
             '  Maximum CPU time: 0 out of 10000\n'
             '  Maximum heap size: 0 out of 6000000\n'
             '  Number of callouts: 0 out of 100\n'
             '  Number of Email Invocations: 0 out of 10\n'
             '  Number of future calls: 0 out of 50\n'
             '  Number of queueable jobs added to the queue: 0 out of 50\n'
             '  Number of Mobile Apex push calls: 0 out of 10\n'
             '\n'
             '22:22:14.19 ()|CUMULATIVE_LIMIT_USAGE_END\n'
             '\n'
             '22:22:14.19 '
             '()|CODE_UNIT_FINISHED|execute_anonymous_apex\n'
             '22:22:14.19 ()|EXECUTION_FINISHED\n',
 'success': True}
```

SalesforceXyToolsCore/Python上でSalesforce組織のSoqlを実行する

Topic

- SalesforceXyToolsCoreを使ってSalesforce組織のSoqlを実行する

メールテンプレートのフォルダーを取得する

- Salesforce組織のユーザ名、パスワード、Apiバージョン、Product/Sandboxを設定してください。

```
from SalesforceXytoolsCore import *
import pprint

config = {
    "api_version": 42.0,
    "username": "sfdc username",
    "password": "sfdc password",
    "security_token": "",
    "is_sandbox": True
}

soap_api = Soap(username=config["username"],
                password=config["password"],
                security_token=config["security_token"],
                sandbox=config["is_sandbox"],
                version=config["api_version"]
                )

"""
Run Soql Queries
"""
soql_string = "SELECT Id, Name FROM User LIMIT 3"
result = soap_api.query(soql_string)
pprint.pprint(result)
```

結果確認

```
OrderedDict([('totalSize', 3),
             ('done', True),
             ('records',
              [OrderedDict([('attributes',
                             OrderedDict([('type', 'User'),
```

```
                                        ('url',

'/services/data/v42.0/sobjects/User/005XXXXXXXXXXXXX')])),
                            ('Id', '005XXXXXXXXXXXXX'),
                            ('Name', 'Process Automated')]),
               OrderedDict([('attributes',
                             OrderedDict([('type', 'User'),
                                          ('url',

'/services/data/v42.0/sobjects/User/005XXXXXXXXXXXXX')])),
                            ('Id', '005XXXXXXXXXXXXX'),
                            ('Name', 'サイトゲストユーザ')]),
               OrderedDict([('attributes',
                             OrderedDict([('type', 'User'),
                                          ('url',

'/services/data/v42.0/sobjects/User/005XXXXXXXXXXXXX')])),
                            ('Id', '005XXXXXXXXXXXXX'),
                            ('Name', 'SFDC Exia')])])])
```

その他方法

query_more

```
soap_api.query_more(sobject_id)
soap_api.query_more("/services/data/v43.0/query/sobject_id", True)
```

query_allですべてのデータを取得する

```
soap_api.query_all("SELECT Id, Email FROM Contact WHERE LastName = 'Jones'")
```

search

```
soap_api.search("FIND {exia}")
```

quick_search

```
soap_api.quick_search("exia")
```

SalesforceXyToolsCore/Python上でテストクラスを実行する

Topic

- SalesforceXyToolsCoreを使ってSalesforceのテストクラスを実行する

テストクラスを実行する

Salesforce組織のユーザ名、パスワード、Apiバージョン、Product/Sandboxを設定してください。

```
from SalesforceXytoolsCore import *
import pprint

config = {
    "api_version": 42.0,
    "username": "sfdc username",
    "password": "sfdc password",
    "security_token": "",
    "is_sandbox": True
}

tooling_api = ToolingApi(username=config["username"],
                password=config["password"],
                security_token=config["security_token"],
                sandbox=config["is_sandbox"],
                version=config["api_version"]
                )

"""run test class"""
id_list = ['xxx', 'xxxx']
tooling_api.runTestSynchronous(id_list)
```

SalesforceXyToolsCore/Python上でSalesforceのREST APIへアクセス

Topic

- SalesforceXyToolsCoreを使ってSalesforceのREST APIへアクセス

Apexclass内容を取得する

- Salesforce組織のユーザ名、パスワード、Apiバージョン、Product/Sandboxを設定してください。

```
from SalesforceXytoolsCore import *
import pprint

config = {
    "api_version": 42.0,
    "username": "sfdc username",
    "password": "sfdc password",
    "security_token": "",
    "is_sandbox": True
}

rest_api = RestApi(username=config["username"],
                password=config["password"],
                security_token=config["security_token"],
                sandbox=config["is_sandbox"],
                version=config["api_version"]
                )

"""
Run Rest API : query apexclass
"""
sel_string = "Select Id, Name From ApexClass Limit 3"
params = {'q': sel_string}
result = rest_api.restful('tooling/query', params)
pprint.pprint(result)
```

結果を確認

```
OrderedDict([('size', 3),
             ('totalSize', 3),
             ('done', True),
             ('queryLocator', None),
             ('entityTypeName', 'ApexClass'),
             ('records',
              [OrderedDict([('attributes',
                             OrderedDict([('type', 'ApexClass'),
                                          ('url',
```

```
'/services/data/v42.0/tooling/sobjects/ApexClass/01pXXXXXXXXXXXX')])),
                            ('Id', '01pXXXXXXXXXXXX'),
                            ('Name', 'OpportXXXXXX')]),
               OrderedDict([('attributes',
                             OrderedDict([('type', 'ApexClass'),
                                          ('url',

'/services/data/v42.0/tooling/sobjects/ApexClass/01pXXXXXXXXXXXX')])),
                            ('Id', '01pXXXXXXXXXXXX'),
                            ('Name', 'DaoXXXXXXXXXXXX')]),
               OrderedDict([('attributes',
                             OrderedDict([('type', 'ApexClass'),
                                          ('url',

'/services/data/v42.0/tooling/sobjects/ApexClass/01pXXXXXXXXXXXX')])),
                            ('Id', '01pXXXXXXXXXXXX'),
                            ('Name', 'ExiaXXXXXXXXXXXX')])])])
```

ApexCodeCoverageカバー率を取得する

```
"""
Run Rest API : query ApexCodeCoverage
"""
sel_string = "SELECT Id, ApexTestClassId, TestMethodName, ApexClassorTriggerId,
NumLinesCovered, NumLinesUncovered, Coverage FROM ApexCodeCoverage"
params = {'q': sel_string}
result = rest_api.restful(
    path='tooling/query',
    params=params,
    method='GET'
)
pprint.pprint(result)
```

Sobjects一覧を取得する

```
"""
Run Rest API : search sobjects
"""
result = rest_api.call_rest(
    method='GET',
    path='/services/data/v37.0/sobjects',
    params={},
)
pprint.pprint(result)
```

結果を確認

```
{'encoding': 'UTF-8',
 'maxBatchSize': 200,
 'sobjects': [
              {'activateable': False,
               'createable': True,
               'custom': False,
               'customSetting': False,
               'deletable': True,
               'deprecatedAndHidden': False,
               'feedEnabled': True,
               'keyPrefix': '006',
               'label': 'Opportunity',
               'labelPlural': 'Opportunities',
               'layoutable': True,
               'mergeable': False,
               'mruEnabled': True,
               'name': 'Opportunity',
               'queryable': True,
               'replicateable': True,
               'retrieveable': True,
               'searchable': True,
               'triggerable': True,
               'undeletable': True,
               'updateable': True,
               'urls': {'approvalLayouts':
'/services/data/v37.0/sobjects/Opportunity/describe/approvalLayouts',
                        'compactLayouts':
'/services/data/v37.0/sobjects/Opportunity/describe/compactLayouts',
                        'defaultValues':
'/services/data/v37.0/sobjects/Opportunity/defaultValues?recordTypeId&fields',
                        'describe':
'/services/data/v37.0/sobjects/Opportunity/describe',
                        'layouts':
'/services/data/v37.0/sobjects/Opportunity/describe/layouts',
                        'listviews':
'/services/data/v37.0/sobjects/Opportunity/listviews',
                        'quickActions':
'/services/data/v37.0/sobjects/Opportunity/quickActions',
                        'rowTemplate':
'/services/data/v37.0/sobjects/Opportunity/{ID}',
                        'sobject': '/services/data/v37.0/sobjects/Opportunity'}},
    ..............
    ..............
    .....省略.......
    ..............
    ..............
 ]
}
```

SalesforceXyToolsCore/Python上でSalesforce組織を説明するメタデータを取得する

Topic

- SalesforceXyToolsCoreを使ってDescribeMetadataを実行して、DescribeMetadataを実行してSalesforce組織を説明するメタデータを取得します。この情報には Apex クラスおよびトリガ、カスタムオブジェクト、標準オブジェクトのカスタム項目、アプリケーションを定義するタブセット、および他の多くのメタデータ型が含まれています。

DescribeMetadataの実行

- Salesforce組織のユーザ名、パスワード、Apiバージョン、Product/Sandboxを設定してください。

```
from SalesforceXytoolsCore import *
import pprint

config = {
    "api_version": 42.0,
    "username": "sfdc username",
    "password": "sfdc password",
    "security_token": "",
    "is_sandbox": True
}

meta_api = MetadataApi(username=config["username"],
                password=config["password"],
                security_token=config["security_token"],
                sandbox=config["is_sandbox"],
                version=config["api_version"]
                )

"""
describeMetadata :
    This call retrieves the metadata that describes your organization.
    This information includes Apex classes and triggers, custom objects, custom fields
on standard objects, tab sets that define an app, and many other metadata types.
"""
result = meta_api.describeMetadata()
pprint.pprint(result)
## print(json.dumps(result, indent=4))
```

結果確認

```
OrderedDict([('metadataObjects',
```

```
[OrderedDict([('directoryName', 'installedPackages'),
              ('inFolder', 'false'),
              ('metaFile', 'false'),
              ('suffix', 'installedPackage'),
              ('xmlName', 'InstalledPackage')]),
 OrderedDict([('childXmlNames', 'CustomLabel'),
              ('directoryName', 'labels'),
              ('inFolder', 'false'),
              ('metaFile', 'false'),
              ('suffix', 'labels'),
              ('xmlName', 'CustomLabels')]),
 OrderedDict([('directoryName', 'staticresources'),
              ('inFolder', 'false'),
              ('metaFile', 'true'),
              ('suffix', 'resource'),
              ('xmlName', 'StaticResource')]),
 OrderedDict([('directoryName', 'scontrols'),
              ('inFolder', 'false'),
              ('metaFile', 'true'),
              ('suffix', 'scf'),
              ('xmlName', 'Scontrol')]),
 OrderedDict([('directoryName', 'certs'),
              ('inFolder', 'false'),
              ('metaFile', 'true'),
              ('suffix', 'crt'),
              ('xmlName', 'Certificate')]),
 OrderedDict([('directoryName', 'aura'),
              ('inFolder', 'false'),
              ('metaFile', 'false'),
              ('xmlName', 'AuraDefinitionBundle')]),
 OrderedDict([('directoryName', 'lightningcomponents'),
              ('inFolder', 'false'),
              ('metaFile', 'false'),
              ('xmlName', 'LightningComponentBundle')]),
 OrderedDict([('directoryName', 'components'),
              ('inFolder', 'false'),
              ('metaFile', 'true'),
              ('suffix', 'component'),
              ('xmlName', 'ApexComponent')]),
 OrderedDict([('directoryName', 'pages'),
              ('inFolder', 'false'),
              ('metaFile', 'true'),
              ('suffix', 'page'),
              ('xmlName', 'ApexPage')]),
 OrderedDict([('directoryName', 'queues'),
              ('inFolder', 'false'),
              ('metaFile', 'false'),
              ('suffix', 'queue'),
              ('xmlName', 'Queue')]),
 OrderedDict([('directoryName', 'CaseSubjectParticles'),
              ('inFolder', 'false'),
              ('metaFile', 'false'),
              ('suffix', 'CaseSubjectParticle'),
```

```
                ('xmlName', 'CaseSubjectParticle')]),
           OrderedDict([('directoryName', 'dataSources'),
                ('inFolder', 'false'),
                ('metaFile', 'false'),
                ('suffix', 'dataSource'),
                ('xmlName', 'ExternalDataSource')]),
           OrderedDict([('directoryName', 'namedCredentials'),
                ('inFolder', 'false'),
                ('metaFile', 'false'),
                ('suffix', 'namedCredential'),
                ('xmlName', 'NamedCredential')]),
           OrderedDict([('directoryName', 'externalServiceRegistrations'),
                ('inFolder', 'false'),
                ('metaFile', 'false'),
                ('suffix', 'externalServiceRegistration'),
                ('xmlName', 'ExternalServiceRegistration')]),
           OrderedDict([('directoryName', 'roles'),
                ('inFolder', 'false'),
                ('metaFile', 'false'),
                ('suffix', 'role'),
                ('xmlName', 'Role')]),
           OrderedDict([('directoryName', 'groups'),
                ('inFolder', 'false'),
                ('metaFile', 'false'),
                ('suffix', 'group'),
                ('xmlName', 'Group')]),
           OrderedDict([('directoryName', 'globalValueSets'),
                ('inFolder', 'false'),
                ('metaFile', 'false'),
                ('suffix', 'globalValueSet'),
                ('xmlName', 'GlobalValueSet')]),
           OrderedDict([('directoryName', 'standardValueSets'),
                ('inFolder', 'false'),
                ('metaFile', 'false'),
                ('suffix', 'standardValueSet'),
                ('xmlName', 'StandardValueSet')]),
           OrderedDict([('directoryName', 'customPermissions'),
                ('inFolder', 'false'),
                ('metaFile', 'false'),
                ('suffix', 'customPermission'),
                ('xmlName', 'CustomPermission')]),
           OrderedDict([('childXmlNames',
                 ['CustomField',
                  'Index',
                  'BusinessProcess',
                  'CompactLayout',
                  'RecordType',
                  'WebLink',
                  'ValidationRule',
                  'SharingReason',
                  'ListView',
                  'FieldSet']),
                ('directoryName', 'objects'),
```

```
                ('inFolder', 'false'),
                ('metaFile', 'false'),
                ('suffix', 'object'),
                ('xmlName', 'CustomObject')]),
        OrderedDict([('directoryName', 'reportTypes'),
                ('inFolder', 'false'),
                ('metaFile', 'false'),
                ('suffix', 'reportType'),
                ('xmlName', 'ReportType')]),
        OrderedDict([('directoryName', 'reports'),
                ('inFolder', 'true'),
                ('metaFile', 'false'),
                ('suffix', 'report'),
                ('xmlName', 'Report')]),
        OrderedDict([('directoryName', 'dashboards'),
                ('inFolder', 'true'),
                ('metaFile', 'false'),
                ('suffix', 'dashboard'),
                ('xmlName', 'Dashboard')]),
        OrderedDict([('directoryName', 'analyticSnapshots'),
                ('inFolder', 'false'),
                ('metaFile', 'false'),
                ('suffix', 'snapshot'),
                ('xmlName', 'AnalyticSnapshot')]),
        OrderedDict([('directoryName', 'feedFilters'),
                ('inFolder', 'false'),
                ('metaFile', 'false'),
                ('suffix', 'feedFilter'),
                ('xmlName', 'CustomFeedFilter')]),
        OrderedDict([('directoryName', 'layouts'),
                ('inFolder', 'false'),
                ('metaFile', 'false'),
                ('suffix', 'layout'),
                ('xmlName', 'Layout')]),
        OrderedDict([('directoryName', 'documents'),
                ('inFolder', 'true'),
                ('metaFile', 'true'),
                ('xmlName', 'Document')]),
        OrderedDict([('directoryName', 'weblinks'),
                ('inFolder', 'false'),
                ('metaFile', 'false'),
                ('suffix', 'weblink'),
                ('xmlName', 'CustomPageWebLink')]),
        OrderedDict([('directoryName', 'letterhead'),
                ('inFolder', 'false'),
                ('metaFile', 'false'),
                ('suffix', 'letter'),
                ('xmlName', 'Letterhead')]),
        OrderedDict([('directoryName', 'email'),
                ('inFolder', 'true'),
                ('metaFile', 'true'),
                ('suffix', 'email'),
                ('xmlName', 'EmailTemplate')]),
```

```
                OrderedDict([('directoryName', 'quickActions'),
                             ('inFolder', 'false'),
                             ('metaFile', 'false'),
                             ('suffix', 'quickAction'),
                             ('xmlName', 'QuickAction')]),
                OrderedDict([('directoryName', 'flexipages'),
                             ('inFolder', 'false'),
                             ('metaFile', 'false'),
                             ('suffix', 'flexipage'),
                             ('xmlName', 'FlexiPage')]),
                OrderedDict([('directoryName', 'tabs'),
                             ('inFolder', 'false'),
                             ('metaFile', 'false'),
                             ('suffix', 'tab'),
                             ('xmlName', 'CustomTab')]),
                OrderedDict([('directoryName', 'customApplicationComponents'),
                             ('inFolder', 'false'),
                             ('metaFile', 'false'),
                             ('suffix', 'customApplicationComponent'),
                             ('xmlName', 'CustomApplicationComponent')]),
                OrderedDict([('directoryName', 'applications'),
                             ('inFolder', 'false'),
                             ('metaFile', 'false'),
                             ('suffix', 'app'),
                             ('xmlName', 'CustomApplication')]),
                OrderedDict([('directoryName', 'EmbeddedServiceConfig'),
                             ('inFolder', 'false'),
                             ('metaFile', 'false'),
                             ('suffix', 'EmbeddedServiceConfig'),
                             ('xmlName', 'EmbeddedServiceConfig')]),
                OrderedDict([('directoryName', 'EmbeddedServiceBranding'),
                             ('inFolder', 'false'),
                             ('metaFile', 'false'),
                             ('suffix', 'EmbeddedServiceBranding'),
                             ('xmlName', 'EmbeddedServiceBranding')]),
                OrderedDict([('directoryName', 'flows'),
                             ('inFolder', 'false'),
                             ('metaFile', 'false'),
                             ('suffix', 'flow'),
                             ('xmlName', 'Flow')]),
                OrderedDict([('directoryName', 'flowDefinitions'),
                             ('inFolder', 'false'),
                             ('metaFile', 'false'),
                             ('suffix', 'flowDefinition'),
                             ('xmlName', 'FlowDefinition')]),
                OrderedDict([('directoryName', 'eventSubscriptions'),
                             ('inFolder', 'false'),
                             ('metaFile', 'false'),
                             ('suffix', 'subscription'),
                             ('xmlName', 'EventSubscription')]),
                OrderedDict([('directoryName', 'eventDeliveries'),
                             ('inFolder', 'false'),
                             ('metaFile', 'false'),
```

```
                    ('suffix', 'delivery'),
                    ('xmlName', 'EventDelivery')]),
          OrderedDict([('childXmlNames',
                        ['WorkflowFieldUpdate',
                         'WorkflowKnowledgePublish',
                         'WorkflowTask',
                         'WorkflowAlert',
                         'WorkflowSend',
                         'WorkflowOutboundMessage',
                         'WorkflowRule']),
                       ('directoryName', 'workflows'),
                       ('inFolder', 'false'),
                       ('metaFile', 'false'),
                       ('suffix', 'workflow'),
                       ('xmlName', 'Workflow')]),
          OrderedDict([('childXmlNames', 'AssignmentRule'),
                       ('directoryName', 'assignmentRules'),
                       ('inFolder', 'false'),
                       ('metaFile', 'false'),
                       ('suffix', 'assignmentRules'),
                       ('xmlName', 'AssignmentRules')]),
          OrderedDict([('childXmlNames', 'AutoResponseRule'),
                       ('directoryName', 'autoResponseRules'),
                       ('inFolder', 'false'),
                       ('metaFile', 'false'),
                       ('suffix', 'autoResponseRules'),
                       ('xmlName', 'AutoResponseRules')]),
          OrderedDict([('childXmlNames', 'EscalationRule'),
                       ('directoryName', 'escalationRules'),
                       ('inFolder', 'false'),
                       ('metaFile', 'false'),
                       ('suffix', 'escalationRules'),
                       ('xmlName', 'EscalationRules')]),
          OrderedDict([('directoryName', 'postTemplates'),
                       ('inFolder', 'false'),
                       ('metaFile', 'false'),
                       ('suffix', 'postTemplate'),
                       ('xmlName', 'PostTemplate')]),
          OrderedDict([('directoryName', 'approvalProcesses'),
                       ('inFolder', 'false'),
                       ('metaFile', 'false'),
                       ('suffix', 'approvalProcess'),
                       ('xmlName', 'ApprovalProcess')]),
          OrderedDict([('directoryName', 'homePageComponents'),
                       ('inFolder', 'false'),
                       ('metaFile', 'false'),
                       ('suffix', 'homePageComponent'),
                       ('xmlName', 'HomePageComponent')]),
          OrderedDict([('directoryName', 'homePageLayouts'),
                       ('inFolder', 'false'),
                       ('metaFile', 'false'),
                       ('suffix', 'homePageLayout'),
                       ('xmlName', 'HomePageLayout')]),
```

```
                OrderedDict([('directoryName', 'objectTranslations'),
                             ('inFolder', 'false'),
                             ('metaFile', 'false'),
                             ('suffix', 'objectTranslation'),
                             ('xmlName', 'CustomObjectTranslation')]),
                OrderedDict([('directoryName', 'globalValueSetTranslations'),
                             ('inFolder', 'false'),
                             ('metaFile', 'false'),
                             ('suffix', 'globalValueSetTranslation'),
                             ('xmlName', 'GlobalValueSetTranslation')]),
                OrderedDict([('directoryName', 'standardValueSetTranslations'),
                             ('inFolder', 'false'),
                             ('metaFile', 'false'),
                             ('suffix', 'standardValueSetTranslation'),
                             ('xmlName', 'StandardValueSetTranslation')]),
                OrderedDict([('directoryName', 'classes'),
                             ('inFolder', 'false'),
                             ('metaFile', 'true'),
                             ('suffix', 'cls'),
                             ('xmlName', 'ApexClass')]),
                OrderedDict([('directoryName', 'triggers'),
                             ('inFolder', 'false'),
                             ('metaFile', 'true'),
                             ('suffix', 'trigger'),
                             ('xmlName', 'ApexTrigger')]),
                OrderedDict([('directoryName', 'testSuites'),
                             ('inFolder', 'false'),
                             ('metaFile', 'false'),
                             ('suffix', 'testSuite'),
                             ('xmlName', 'ApexTestSuite')]),
                OrderedDict([('directoryName', 'profiles'),
                             ('inFolder', 'false'),
                             ('metaFile', 'false'),
                             ('suffix', 'profile'),
                             ('xmlName', 'Profile')]),
                OrderedDict([('directoryName', 'permissionsets'),
                             ('inFolder', 'false'),
                             ('metaFile', 'false'),
                             ('suffix', 'permissionset'),
                             ('xmlName', 'PermissionSet')]),
                OrderedDict([('directoryName', 'customMetadata'),
                             ('inFolder', 'false'),
                             ('metaFile', 'false'),
                             ('suffix', 'md'),
                             ('xmlName', 'CustomMetadata')]),
                OrderedDict([('directoryName', 'profilePasswordPolicies'),
                             ('inFolder', 'false'),
                             ('metaFile', 'false'),
                             ('suffix', 'profilePasswordPolicy'),
                             ('xmlName', 'ProfilePasswordPolicy')]),
                OrderedDict([('directoryName', 'profileSessionSettings'),
                             ('inFolder', 'false'),
                             ('metaFile', 'false'),
```

```
                ('suffix', 'profileSessionSetting'),
                ('xmlName', 'ProfileSessionSetting')]),
            OrderedDict([('directoryName', 'datacategorygroups'),
                ('inFolder', 'false'),
                ('metaFile', 'false'),
                ('suffix', 'datacategorygroup'),
                ('xmlName', 'DataCategoryGroup')]),
            OrderedDict([('directoryName', 'remoteSiteSettings'),
                ('inFolder', 'false'),
                ('metaFile', 'false'),
                ('suffix', 'remoteSite'),
                ('xmlName', 'RemoteSiteSetting')]),
            OrderedDict([('directoryName', 'cspTrustedSites'),
                ('inFolder', 'false'),
                ('metaFile', 'false'),
                ('suffix', 'cspTrustedSite'),
                ('xmlName', 'CspTrustedSite')]),
            OrderedDict([('childXmlNames', 'MatchingRule'),
                ('directoryName', 'matchingRules'),
                ('inFolder', 'false'),
                ('metaFile', 'false'),
                ('suffix', 'matchingRule'),
                ('xmlName', 'MatchingRules')]),
            OrderedDict([('directoryName', 'duplicateRules'),
                ('inFolder', 'false'),
                ('metaFile', 'false'),
                ('suffix', 'duplicateRule'),
                ('xmlName', 'DuplicateRule')]),
            OrderedDict([('directoryName', 'cleanDataServices'),
                ('inFolder', 'false'),
                ('metaFile', 'false'),
                ('suffix', 'cleanDataService'),
                ('xmlName', 'CleanDataService')]),
            OrderedDict([('directoryName', 'authproviders'),
                ('inFolder', 'false'),
                ('metaFile', 'false'),
                ('suffix', 'authprovider'),
                ('xmlName', 'AuthProvider')]),
            OrderedDict([('directoryName', 'eclair'),
                ('inFolder', 'false'),
                ('metaFile', 'true'),
                ('suffix', 'geodata'),
                ('xmlName', 'EclairGeoData')]),
            OrderedDict([('directoryName', 'sites'),
                ('inFolder', 'false'),
                ('metaFile', 'false'),
                ('suffix', 'site'),
                ('xmlName', 'CustomSite')]),
            OrderedDict([('directoryName', 'channelLayouts'),
                ('inFolder', 'false'),
                ('metaFile', 'false'),
                ('suffix', 'channelLayout'),
                ('xmlName', 'ChannelLayout')]),
```

```
            OrderedDict([('directoryName', 'contentassets'),
                         ('inFolder', 'false'),
                         ('metaFile', 'true'),
                         ('suffix', 'asset'),
                         ('xmlName', 'ContentAsset')]),
            OrderedDict([('childXmlNames',
                          ['SharingOwnerRule', 'SharingCriteriaRule']),
                         ('directoryName', 'sharingRules'),
                         ('inFolder', 'false'),
                         ('metaFile', 'false'),
                         ('suffix', 'sharingRules'),
                         ('xmlName', 'SharingRules')]),
            OrderedDict([('directoryName', 'sharingSets'),
                         ('inFolder', 'false'),
                         ('metaFile', 'false'),
                         ('suffix', 'sharingSet'),
                         ('xmlName', 'SharingSet')]),
            OrderedDict([('directoryName', 'communities'),
                         ('inFolder', 'false'),
                         ('metaFile', 'false'),
                         ('suffix', 'community'),
                         ('xmlName', 'Community')]),
            OrderedDict([('directoryName', 'ChatterExtensions'),
                         ('inFolder', 'false'),
                         ('metaFile', 'false'),
                         ('suffix', 'ChatterExtension'),
                         ('xmlName', 'ChatterExtension')]),
            OrderedDict([('directoryName', 'callCenters'),
                         ('inFolder', 'false'),
                         ('metaFile', 'false'),
                         ('suffix', 'callCenter'),
                         ('xmlName', 'CallCenter')]),
            OrderedDict([('directoryName', 'connectedApps'),
                         ('inFolder', 'false'),
                         ('metaFile', 'false'),
                         ('suffix', 'connectedApp'),
                         ('xmlName', 'ConnectedApp')]),
            OrderedDict([('directoryName', 'appMenus'),
                         ('inFolder', 'false'),
                         ('metaFile', 'false'),
                         ('suffix', 'appMenu'),
                         ('xmlName', 'AppMenu')]),
            OrderedDict([('directoryName', 'delegateGroups'),
                         ('inFolder', 'false'),
                         ('metaFile', 'false'),
                         ('suffix', 'delegateGroup'),
                         ('xmlName', 'DelegateGroup')]),
            OrderedDict([('directoryName', 'siteDotComSites'),
                         ('inFolder', 'false'),
                         ('metaFile', 'true'),
                         ('suffix', 'site'),
                         ('xmlName', 'SiteDotCom')]),
            OrderedDict([('directoryName', 'networkBranding'),
```

```
                ('inFolder', 'false'),
                ('metaFile', 'true'),
                ('suffix', 'networkBranding'),
                ('xmlName', 'NetworkBranding')]),
            OrderedDict([('directoryName', 'brandingSets'),
                ('inFolder', 'false'),
                ('metaFile', 'false'),
                ('suffix', 'brandingSet'),
                ('xmlName', 'BrandingSet')]),
            OrderedDict([('directoryName', 'flowCategories'),
                ('inFolder', 'false'),
                ('metaFile', 'false'),
                ('suffix', 'flowCategory'),
                ('xmlName', 'FlowCategory')]),
            OrderedDict([('directoryName', 'lightningBolts'),
                ('inFolder', 'false'),
                ('metaFile', 'false'),
                ('suffix', 'lightningBolt'),
                ('xmlName', 'LightningBolt')]),
            OrderedDict([('directoryName', 'lightningExperienceThemes'),
                ('inFolder', 'false'),
                ('metaFile', 'false'),
                ('suffix', 'lightningExperienceTheme'),
                ('xmlName', 'LightningExperienceTheme')]),
            OrderedDict([('directoryName', 'samlssoconfigs'),
                ('inFolder', 'false'),
                ('metaFile', 'false'),
                ('suffix', 'samlssoconfig'),
                ('xmlName', 'SamlSsoConfig')]),
            OrderedDict([('directoryName', 'corsWhitelistOrigins'),
                ('inFolder', 'false'),
                ('metaFile', 'false'),
                ('suffix', 'corsWhitelistOrigin'),
                ('xmlName', 'CorsWhitelistOrigin')]),
            OrderedDict([('directoryName', 'actionLinkGroupTemplates'),
                ('inFolder', 'false'),
                ('metaFile', 'false'),
                ('suffix', 'actionLinkGroupTemplate'),
                ('xmlName', 'ActionLinkGroupTemplate')]),
            OrderedDict([('directoryName', 'transactionSecurityPolicies'),
                ('inFolder', 'false'),
                ('metaFile', 'false'),
                ('suffix', 'transactionSecurityPolicy'),
                ('xmlName', 'TransactionSecurityPolicy')]),
            OrderedDict([('directoryName', 'synonymDictionaries'),
                ('inFolder', 'false'),
                ('metaFile', 'false'),
                ('suffix', 'synonymDictionary'),
                ('xmlName', 'SynonymDictionary')]),
            OrderedDict([('directoryName', 'pathAssistants'),
                ('inFolder', 'false'),
                ('metaFile', 'false'),
                ('suffix', 'pathAssistant'),
```

```
                    ('xmlName', 'PathAssistant')]),
          OrderedDict([('directoryName', 'LeadConvertSettings'),
                    ('inFolder', 'false'),
                    ('metaFile', 'false'),
                    ('suffix', 'LeadConvertSetting'),
                    ('xmlName', 'LeadConvertSettings')]),
          OrderedDict([('directoryName', 'cachePartitions'),
                    ('inFolder', 'false'),
                    ('metaFile', 'false'),
                    ('suffix', 'cachePartition'),
                    ('xmlName', 'PlatformCachePartition')]),
          OrderedDict([('directoryName', 'topicsForObjects'),
                    ('inFolder', 'false'),
                    ('metaFile', 'false'),
                    ('suffix', 'topicsForObjects'),
                    ('xmlName', 'TopicsForObjects')]),
          OrderedDict([('directoryName', 'emailservices'),
                    ('inFolder', 'false'),
                    ('metaFile', 'false'),
                    ('suffix', 'xml'),
                    ('xmlName', 'EmailServicesFunction')]),
          OrderedDict([('directoryName', 'settings'),
                    ('inFolder', 'false'),
                    ('metaFile', 'false'),
                    ('suffix', 'settings'),
                    ('xmlName', 'Settings')])]),
        ('organizationNamespace', None),
        ('partialSaveAllowed', 'true'),
        ('testRequired', 'false')])
```

SalesforceXyToolsCore/Python上でSalesforce組織からのPackage.xmlを作成

Topic

- SalesforceXyToolsCoreを使ってSalesforce組織からのPackage.xmlを作成

Package.xmlの作成

- Salesforce組織のユーザ名、パスワード、Apiバージョン、Product/Sandboxを設定してください。

```
from SalesforceXytoolsCore import *
import pprint

config = {
    "api_version": 42.0,
    "username": "sfdc username",
    "password": "sfdc password",
    "security_token": "",
    "is_sandbox": True
}

meta_api = MetadataApi(username=config["username"],
                password=config["password"],
                security_token=config["security_token"],
                sandbox=config["is_sandbox"],
                version=config["api_version"]
                )


"""buildPackageXml"""
packagexml = meta_api.buildPackageXml()
```

結果確認:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">

    <types>
        <members>*</members>
        <name>InstalledPackage</name>
    </types>
    <types>
        <members>*</members>
        <name>CustomLabels</name>
    </types>
```

```
        .....................
        .........省略.........
        .....................
    <types>
        <members>Account</members>
        <members>AccountChangeEvent</members>
        <members>AccountCleanInfo</members>
        <members>AccountContactRole</members>
        <members>AccountFeed</members>
        <members>AccountHistory</members>
        <members>AccountPartner</members>
        <members>AccountShare</members>
        <members>ActionLinkGroupTemplate</members>
        <members>ActionLinkTemplate</members>
        <members>ActivityHistory</members>
        <members>AdditionalNumber</members>
        <members>AggregateResult</members>
        .....................
        .........省略.........
        .....................
        <name>CustomObject</name>
    </types>
    <types>
        <members>unfiled$public</members>
        .....................
        .........省略.........
        .....................
        <members>unfiled$public/SupportCaseAssignmentNotification</members>
        <members>unfiled$public/SalesNewCustomerEmail</members>
        <members>unfiled$public/SupportCaseCreatedWebInquiries</members>
        <members>unfiled$public/SupportEscalatedCaseNotification</members>
        <members>unfiled$public/SupportSelfServiceResetPassword</members>
        <members>unfiled$public/MarketingProductInquiryResponse</members>
        <members>unfiled$public/SupportCaseCreatedPhoneInquiries</members>
        <members>unfiled$public/SUPPORTSelfServiceResetPasswordSAMPLE</members>
        <members>unfiled$public/SupportEscalatedCaseReassignment</members>

<members>unfiled$public/SUPPORTSelfServiceNewCommentNotificationSAMPLE</members>
        <name>EmailTemplate</name>
    </types>

        .....................
        .........省略.........
        .....................
    <types>
        <members>*</members>
        <name>TopicsForObjects</name>
    </types>
    <types>
        <members>*</members>
        <name>EmailServicesFunction</name>
    </types>
```

```
    <types>
        <members>*</members>
        <name>Settings</name>
    </types>
    <version>42.0</version>
</Package>
```

SFDC組織からPackage関するリストを取得

```
"""packageTypeList"""
print('>>>packageTypeList')
pprint.pprint(meta_api.packageTypeList())
```

結果確認

```
[{'members': ['*'], 'name': 'InstalledPackage'},
 {'members': ['*'], 'name': 'CustomLabels'},
 {'members': ['*'], 'name': 'StaticResource'},
         ..................
        ........省略.........
         ..................
 {'members': ['*'], 'name': 'TopicsForObjects'},
 {'members': ['*'], 'name': 'EmailServicesFunction'},
 {'members': ['*'], 'name': 'Settings'}]
```

SalesforceXyToolsCore/Python上ですべてのMetadataを情報を取得してみよう

Topic

- SalesforceXyToolsCoreを使ってすべてのMetadataを情報を取得する。
- メタデータのID,　Typeを取得可能です。

すべてのMetadataを情報を取得する

- Salesforce組織のユーザ名、パスワード、Apiバージョン、Product/Sandboxを設定してください。

```
from SalesforceXytoolsCore import *
import pprint

config = {
    "api_version": 42.0,
    "username": "sfdc username",
    "password": "sfdc password",
    "security_token": "",
    "is_sandbox": True
}

meta_api = MetadataApi(username=config["username"],
                password=config["password"],
                security_token=config["security_token"],
                sandbox=config["is_sandbox"],
                version=config["api_version"]
                )


"""getAllMetadataMap"""
all_metadata_map = meta_api.getAllMetadataMap()
pprint.pprint(all_metadata_map)
```

結果確認

```
{'ApexClass': {'classes/A_Batch.cls': OrderedDict([('createdById',
                                                    '005xxxxxxxxxxxx'),
                                                   ('createdByName',
                                                    'SFDC Exia'),
                                                   ('createdDate',
                                                    '2010-08-09T00:07:13.000Z'),
                                                   ('fileName',
```

```
                                                 'classes/A_Batch.cls'),
                                                ('fullName', 'A_Batch'),
                                                ('id', '01pxxxxxxxxxxxx'),
                                                ('lastModifiedById',
                                                 '005xxxxxxxxxxxx'),
                                                ('lastModifiedByName',
                                                 'SFDC Exia'),
                                                ('lastModifiedDate',
                                                 '2010-08-09T00:24:29.000Z'),
                                                ('manageableState',
                                                 'unmanaged'),
                                                ('type', 'ApexClass')]),
                .................省略
                 'Report': {
              'reports/Xy/MyID.report': OrderedDict([('createdById',
                                                 '005xxxxxxxxxxxx'),
                                                ('createdByName',
                                                 'SFDC Exia'),
                                                ('createdDate',
                                                 '2010-09-15T00:29:36.000Z'),
                                                ('fileName',
                                                 'reports/Xy/MyID.report'),
                                                ('fullName', 'Xy/MyID'),
                                                ('id', '00Oxxxxxxxxxxxxxx'),
                                                ('lastModifiedById',
                                                 '005XXXXXXXXXXXXX'),
                                                ('lastModifiedByName',
                                                 'SFDC Exia'),
                                                ('lastModifiedDate',
                                                 '2010-09-25T00:03:33.000Z'),
                                                ('manageableState',
                                                 'unmanaged'),
                                                ('type', 'Report')]),
                .................省略
}
```

Metadataのリストを取得したい場合

```
"""listAllMetadata """
for meta in meta_api.listAllMetadata():
    pprint.pprint(meta)
```

指定したメタデータを取得する

```
"""listmeta"""
query_option_list = [
```

```
    {
        "metadata_type" : "EmailFolder",
        "folder" : ""
    },
    {
        "metadata_type" : "ApexClass",
        "folder" : ""
    }
]
listmeta_result = meta_api.listMetadata(query_option_list)
print(len(listmeta_result))
for meta in listmeta_result:
    pprint.pprint(meta)
```

SalesforceXyToolsCore/Python上でSalesforce組織のフォルダーを取得する

Topic

- SalesforceXyToolsCoreを使ってすべてのSalesforce組織のフォルダーを取得する

メールテンプレートのフォルダーを取得する

- Salesforce組織のユーザ名、パスワード、Apiバージョン、Product/Sandboxを設定してください。

```
from SalesforceXytoolsCore import *
import pprint

config = {
    "api_version": 42.0,
    "username": "sfdc username",
    "password": "sfdc password",
    "security_token": "",
    "is_sandbox": True
}

meta_api = MetadataApi(username=config["username"],
                password=config["password"],
                security_token=config["security_token"],
                sandbox=config["is_sandbox"],
                version=config["api_version"]
                )


"""listFolder"""
folders = meta_api.listFolder("EmailTemplate")
pprint.pprint(folders)
```

結果確認

```
{'ApexClass': {'classes/A_Batch.cls': OrderedDict([('createdById',
                                                    '005xxxxxxxxxxxx'),
                                                   ('createdByName',
                                                    'SFDC Exia'),
                                                   ('createdDate',
                                                    '2010-08-09T00:07:13.000Z'),
                                                   ('fileName',
                                                    'classes/A_Batch.cls'),
```

```
                                                ('fullName', 'A_Batch'),
                                                ('id', '01pxxxxxxxxxxxx'),
                                                ('lastModifiedById',
                                                 '005xxxxxxxxxxxx'),
                                                ('lastModifiedByName',
                                                 'SFDC Exia'),
                                                ('lastModifiedDate',
                                                 '2010-08-09T00:24:29.000Z'),
                                                ('manageableState',
                                                 'unmanaged'),
                                                ('type', 'ApexClass')]),
               .................省略
                'Report': {
           'reports/Xy/MyID.report': OrderedDict([('createdById',
                                                 '005xxxxxxxxxxxx'),
                                                ('createdByName',
                                                 'SFDC Exia'),
                                                ('createdDate',
                                                 '2010-09-15T00:29:36.000Z'),
                                                ('fileName',
                                                 'reports/Xy/MyID.report'),
                                                ('fullName', 'Xy/MyID'),
                                                ('id', '00Oxxxxxxxxxxxxx'),
                                                ('lastModifiedById',
                                                 '005xxxxxxxxxxxx'),
                                                ('lastModifiedByName',
                                                 'SFDC Exia'),
                                                ('lastModifiedDate',
                                                 '2010-09-25T00:03:33.000Z'),
                                                ('manageableState',
                                                 'unmanaged'),
                                                ('type', 'Report')]),
               .................省略
}
```

その以外のフォルダー

Salesforce には、現在次の 4 つのフォルダの種類があります。

- ドキュメントフォルダ
- メールフォルダ
- レポートフォルダ
- ダッシュボードフォルダ

Four folder types currently exist in Salesforce:

- Document folder
- Email folder

- Report folder
- Dashboard folder

取得方法:

```
"""Email folderの取得"""
folders = meta_api.listFolder("EmailTemplate")

"""Document folderの取得"""
folders = meta_api.listFolder("Document")

"""Report folderの取得"""
folders = meta_api.listFolder("Report")

"""Dashboard folderの取得"""
folders = meta_api.listFolder("Dashboard")
```

SalesforceXyToolsCore/Python上でたった3行でSalesforce組織からのメタデータの取得

Topic

- SalesforceXyToolsCoreを使ってSalesforce組織からのメタデータの取得

メリット

- package.xml配置不要、動的に生成
- すべてのSalesforce組織からのメタデータ一括で取得

Salesforce組織からのメタデータの取得

Salesforce組織のユーザ名、パスワード、Apiバージョン、Product/Sandboxを設定してください。

```
from SalesforceXytoolsCore import *
import pprint

config = {
    "api_version": 42.0,
    "username": "sfdc username",
    "password": "sfdc password",
    "security_token": "",
    "is_sandbox": True
}

meta_api = MetadataApi(username=config["username"],
                password=config["password"],
                security_token=config["security_token"],
                sandbox=config["is_sandbox"],
                version=config["api_version"]
                )


"""retrieve zip file"""
meta_api.retrieveZip("./save_dir","metadata-retrieve.zip")
```

結果確認:

```
ls ./save_dir
```

分析

メタデータ取得ロジック

1. retrieve() コールを発行し、非同期的な取得を開始すると、AsyncResult オブジェクトが返されます。id 項目の値をメモし、次のステップで使用します。
2. checkRetrieveStatus() コールを発行して、最初のステップの AsyncResult オブジェクトから id 値を渡します。返された RetrieveResult の done 項目の値をチェックします。true の場合、コールが完了して、次のステップに進むことを意味します。それ以外の場合は、done 項目が true になるまで、このステップを繰り返して checkRetrieveStatus() を再度コールします。
3. 前のステップの checkRetrieveStatus() への最後のコールで返された RetrieveResult から zip ファイル (zipFile 項目) および他の必要な項目を取得します。

retrieveタスクを開始

タスクを発行する

```
"""retrieve"""
result = meta_api.startRetrieve()
pprint.pprint(result)
## print(result["done"])
## print(result["id"])
## print(result["state"])
```

retrieveの状況を確認

宣言的なメタデータコール retrieve() の状況を確認し、zip ファイルの内容を返します。

```
"""checkRetrieveStatus"""
retrieve_id = result["id"]
check_result = meta_api.checkRetrieveStatus(retrieve_id)
pprint.pprint(check_result)
```

SalesforceXyToolsCore/Python上でSalesforce組織の添付ファイルを一括ダウンロードする

Topic

- SalesforceXyToolsCoreを使ってSalesforce組織の添付ファイルを一括ダウンロードする

Attachment

User が親オブジェクトにアップロードおよび添付したファイルをダウンロードする。 `Body` をファイルに保存すれば、完了です。

Salesforce組織の添付ファイルを一括ダウンロードする

- Salesforce組織のユーザ名、パスワード、Apiバージョン、Product/Sandboxを設定してください。

```
from SalesforceXytoolsCore import *
import pprint

config = {
    "api_version": 42.0,
    "username": "sfdc username",
    "password": "sfdc password",
    "security_token": "",
    "is_sandbox": True
}

SAVE_DIR = './attachments_download'
if not os.path.exists(SAVE_DIR):
    os.mkdir(SAVE_DIR)

rest_api = RestApi(username=config["username"],
                password=config["password"],
                security_token=config["security_token"],
                sandbox=config["is_sandbox"],
                version=config["api_version"]
                )

attachments = rest_api.query("SELECT Id, Name, Body FROM Attachment LIMIT 2000")
print("添付ファイル件数 : " + str(len(attachments)))

for attachment in attachments["records"]:
    print("start to download : " + attachment["Name"])
    """
    Run Rest API : download attachment
    """
    rest_path =
"/services/data/v42.0/sobjects/Attachment/{id}/Body".format(id=attachment["Id"])
    result = rest_api.call_rest(
```

```
        method='GET',
        path=rest_path,
        params={},
    )
    with open(os.path.join(SAVE_DIR, attachment["Id"] + "_" + attachment["Name"]),
mode='wb') as f:
        f.write(result.content)
```

実行すれば、添付ファイル件数ファイルをダウンロードする可能です。