

```
In [ ]: import ldlite as ldlite
        ld = ldlite.LDLite()
```

```
In [ ]: db = ld.connect_db(filename='test6_ldlite.db')
```

```
In [ ]: ld.connect_okapi(url='https://okapi-wellesley.folio.ebsco.com',
                        tenant='',
                        user='',
                        password='',
                        )
```

```
In [ ]: allrec = 'cql.allRecords=1 sortby id'
```

```
In [ ]: queries = [
        ('folio_inventory.item_damaged_status', '/item-damaged-statuses', allrec),
        ('folio_inventory.item_note_type', '/item-note-types', allrec),
        ('folio_inventory.loan_type', '/loan-types', allrec),
        ('folio_inventory.location', '/locations', allrec),
        ('folio_inventory.loccampus', '/location-units/campuses', allrec),
        ('folio_inventory.locinstitution', '/location-units/institutions', allrec),
        ('folio_inventory.loclibrary', '/location-units/libraries', allrec),
        ('folio_inventory.material_type', '/material-types', allrec),
        ('folio_inventory.mode_of_issuance', '/modes-of-issuance', allrec),
        ('folio_inventory.nature_of_content_term', '/nature-of-content-terms', allrec),
        ('folio_inventory.service_point', '/service-points', allrec),
        ('folio_inventory.statistical_code', '/statistical-codes', allrec),
        ('folio_inventory.statistical_code_type', '/statistical-code-types', allrec),
        ('folio_notes.note_data', '/notes', allrec),
        ('folio_organizations.addresses', '/organizations-storage/addresses', allrec),
        ('folio_organizations.categories', '/organizations-storage/categories', allrec),
        ('folio_organizations.contacts', '/organizations-storage/contacts', allrec),
        ('folio_organizations.emails', '/organizations-storage/emails', allrec),
        ('folio_organizations.interfaces', '/organizations-storage/interfaces', allrec),
        ('folio_organizations.phone_numbers', '/organizations-storage/phone-numbers', allrec),
        ('folio_organizations.urls', '/organizations-storage/urls', allrec),
        ('folio_users.users', '/users', allrec),
        ('folio_agreements.entitlement', '/erm/entitlements', allrec),
        ('folio_agreements.erm_resource', '/erm/resource', allrec),
        ('folio_agreements.org', '/erm/org', allrec),
        ('folio_agreements.refdata_value', '/erm/refdata', allrec),
        ('folio_configuration.config_data', '/configurations/entries', allrec),
        ('folio_courses.coursereserves_copyrightstates', '/coursereserves/copyrightstates', allrec),
        ('folio_courses.coursereserves_courselistings', '/coursereserves/courselistings', allrec),
        ('folio_courses.coursereserves_courses', '/coursereserves/courses', allrec),
        ('folio_courses.coursereserves_coursetypes', '/coursereserves/coursetypes', allrec),
        ('folio_courses.coursereserves_departments', '/coursereserves/departments', allrec),
        ('folio_courses.coursereserves_processingstates', '/coursereserves/processingstates', allrec),
        ('folio_courses.coursereserves_reserves', '/coursereserves/reserves', allrec),
        ('folio_courses.coursereserves_roles', '/coursereserves/roles', allrec),
        ('folio_courses.coursereserves_terms', '/coursereserves/terms', allrec),
        ('folio_inventory.alternative_title_type', '/alternative-title-types', allrec),
        ('folio_inventory.call_number_type', '/call-number-types', allrec),
        ('folio_inventory.classification_type', '/classification-types', allrec),
        ('folio_inventory.contributor_name_type', '/contributor-name-types', allrec),
        ('folio_inventory.contributor_type', '/contributor-types', allrec),
        ('folio_inventory.electronic_access_relationship', '/electronic-access-relationships', allrec),
    ]
```

```

('folio_inventory.holdings_note_type', '/holdings-note-types', allrec),
('folio_inventory.holdings_records_source', '/holdings-sources', allrec),
('folio_inventory.holdings_type', '/holdings-types', allrec),
('folio_inventory.identifier_type', '/identifier-types', allrec),
('folio_inventory.ill_policy', '/ill-policies', allrec),
('folio_inventory.instance_format', '/instance-formats', allrec),
('folio_inventory.instance_note_type', '/instance-note-types', allrec),
('folio_inventory.instance_status', '/instance-statuses', allrec),
('folio_inventory.instance_type', '/instance-types', allrec),
('folio_inventory.holdings_record', '/holdings-storage/holdings', allrec),
]

```

```

In [ ]: tables = []
for q in queries:
    try:
        if len(q) == 4:
            t = ld.query(table=q[0], path=q[1], query=q[2], json_depth=q[3])
        else:
            t = ld.query(table=q[0], path=q[1], query=q[2])
        tables += t
    except (ValueError, RuntimeError):
        print('folio_demo.py: error processing "' + q[1] + '"', file=sys.stderr)
print()
print('Tables:')
for t in tables:
    print(t)
print('(' + str(len(tables)) + ' tables)')

```

```

In [ ]: queries = [
        ('folio_inventory.item', '/item-storage/items', allrec),
        ]

```

```

In [ ]: tables = []
for q in queries:
    try:
        if len(q) == 4:
            t = ld.query(table=q[0], path=q[1], query=q[2], json_depth=q[3])
        else:
            t = ld.query(table=q[0], path=q[1], query=q[2])
        tables += t
    except (ValueError, RuntimeError):
        print('folio_demo.py: error processing "' + q[1] + '"', file=sys.stderr)
print()
print('Tables:')
for t in tables:
    print(t)
print('(' + str(len(tables)) + ' tables)')

```

```

In [ ]: queries = [
        ('folio_inventory.instance_record', 'instance-storage/instances', allrec)
        ]

```

```

In [ ]: tables = []
for q in queries:
    try:
        if len(q) == 4:
            t = ld.query(table=q[0], path=q[1], query=q[2], json_depth=q[3])
        else:

```

```

        t = ld.query(table=q[0], path=q[1], query=q[2])
        tables += t
    except (ValueError, RuntimeError):
        print('folio_demo.py: error processing "' + q[1] + '"', file=sys.stderr)
print()
print('Tables:')
for t in tables:
    print(t)
print('(' + str(len(tables)) + ' tables')

```

```

In [ ]: queries = [
        ('folio_inventory.instance_relationship', '/instance-storage/instance-relationship
        ]

```

```

In [ ]: tables = []
        for q in queries:
            try:
                if len(q) == 4:
                    t = ld.query(table=q[0], path=q[1], query=q[2], json_depth=q[3])
                else:
                    t = ld.query(table=q[0], path=q[1], query=q[2])
                tables += t
            except (ValueError, RuntimeError):
                print('folio_demo.py: error processing "' + q[1] + '"', file=sys.stderr)
print()
print('Tables:')
for t in tables:
    print(t)
print('(' + str(len(tables)) + ' tables')

```

```

In [ ]: queries = [
        ('folio_source_record.records', '/source-storage/records', {}, 2),
        ]

```

```

In [ ]: tables = []
        for q in queries:
            try:
                if len(q) == 4:
                    t = ld.query(table=q[0], path=q[1], query=q[2], json_depth=q[3])
                else:
                    t = ld.query(table=q[0], path=q[1], query=q[2])
                tables += t
            except (ValueError, RuntimeError):
                print('folio_demo.py: error processing "' + q[1] + '"', file=sys.stderr)
print()
print('Tables:')
for t in tables:
    print(t)
print('(' + str(len(tables)) + ' tables')

```