



# *OSDU Data Platform Application Developer Training*

## *Storage Service*



# Storage Service Overview

- » Provides APIs to ingest, modify, delete, and version metadata
- » The ingested metadata is called *record*
- » Records are split in 2 parts stored in
  - Document database
  - File storage
- » Created record is indexed by the Indexer service and then available in the Search

# Record Structure

```
{  
  "id": "partition:type:uuid",  
  "kind": "authority:source:entity-type:major-  
version:minor-version.patch-version",  
  "acl": {  
    "viewers": ["viewers@osdu.opengroup.org"],  
    "owners": ["owners@osdu.opengroup.org"]  
  },  
  "legal": {  
    "legaltags": ["sample-legaltag"],  
    "otherRelevantDataCountries": ["US", "FR"]  
  },  
  "data": {  
    "msg": "Some Data"  
  }  
}
```

- » **id**: Unique identifier of the record. Must follow the naming convention.
- » **kind**: Kind of the record. Must follow the naming convention.
- » **acl**: Users who have access to the record.
- » **legal**: Legal constraints on the record.
- » **data**: Record payload represented as a sequence of key-value pairs.

# Record Splitting

## » Database

```
{
  "_id": "opendes:dataset--File.Generic:1234567",
  "kind": "opendes:wks:dataset--File.Generic:1.0.0",
  "acl": {...},
  "legal": {...},
  "tags": {},
  "status": "active",
  "user": "osdu-ingestion@osdu.opengroup.com",
  "createTime": 1623240157732,
  "modifyTime": 0
}
```

## » Storage

```
{
  "data": {
    "DatasetProperties": {
      "FileSourceInfo": {
        "FileSource": "abc123",
        "PreLoadFilePath": "https://route/to/file",
        ...
      }
    },
    "ExtensionProperties": {...}
  },
  "meta": [{}]
```

# Kinds and Schemas

- » Schemas and records are connected through the kind attribute.
- » A schema defines the structure of a record.
- » Only fields present in the schema can be indexed.

# Patch API

- » Bulk Update API allows updating records in batch
- » Provide:
  - Record IDs (max 500)
  - Operation type: *replace*, *add*, or *remove*
  - Path to the property
  - New value

```
{  
  "query": {  
    "ids": ["tenant1:type:unique-  
identifier:version"]  
  },  
  "ops": [{  
    "op": "replace",  
    "path": "/legal/legaltags",  
    "value": [  
      "sample-legaltag1",  
      "sample-legaltag2"  
    ]  
  }  
}]  
}
```

# Prerequisites to Creating Records

- » Ingest **schemas** via the Schema service
- » Create **legal tags** via the Legal Service
- » Choose **data groups** to assign. Create new ones via the Entitlements Service and add members or use defaults:

```
data.default.viewers@{data-partition-id}.{domain}  
data.default.owners@{data-partition-id}.{domain}
```

# Thank you!