# Sqoop 2 (1.99.4) Entity Nomenclature and Relationships

## Sqoop 2 Actors

| Actor | Actions Performed |
|---|---|
| Sqoop Admins | Creating LINK objects and LINK Config Inputs |
| Sqoop Users | Creating and Executing Sqoop JOBs |
| Sqoop Connector developers | Developing Connectors and Connector related config semantics |

## Sqoop 2 Entities

Since 1.99.4 release we renamed a sqoop entities and added one new entity "**CONFIGURABLE"** that acts as a one of the core entities representing sqoop object exposing configs.  Here is the list of configs

| Sqoop Entity | Before 1.99.4 It was called? | Sqoop Java Classes | Relationships to to other entities | Description | CRUD Operations supported via command shell or REST |
|---|---|---|---|---|---|
| **CONFIGURABLE** | N/A | `Configurable.java ( abstract class)` | Top Level Entity | Represents a core entity that exposes config objects and used in sqoop job lifecycle.<br><br>Configurable have a associated version that acts as a identifier for connector config upgrades.<br><br>**MConfigurableType**<br>`/**`<br>` * Represents the sqoop entities that can own configs`<br>` */`<br>`public enum MConfigurableType {`<br>`  /** Connector as a owner of config keys */`<br>`  CONNECTOR,`<br>`  /** Driver as a owner of config keys */`<br>`  DRIVER;`<br>`}` | **READ ONLY.**<br><br>• /v1/configurable/driver<br>• /v1/configurable/connector/[cid] |

| | | | | | |
|---|---|---|---|---|---|
| **CONNECTOR** | Same | `MConnector.java` | <ul><li>HAS 1-n **CONFIG** objects</li><li>HAS 1-n **LINK** objects</li></ul> | ***is a type of configurable***<br><br>*There can be many connectors registered to the sqoop server* | **READ ONLY**<br><br>Connectors and their exposed config objects are registered to the sqoop server at run-time when server starts. They are actual code artifacts packaged as jars. But they are also stored in the sqoop persistent store referred to as the repository to uniquely identify them and their config objects they expose.<br><br>Connector upgrades are also supported across sqoop releases.<br><ul><li>/v1/connectors - [GET] Get all Connectors</li><li>/v1/connector/[cname] or /v1/connector/[cid] - [GET] - Get Connector</li></ul> |
| **DRIVER** | FRAMEWORK | `MDriver.java` | <ul><li>HAS 1-n **CONFIG** objects</li></ul> | ***is a type of configurable***<br><br>*There is only one Driver object representing sqoop in the system* | **READ ONLY.**<br><br>Driver is also registered to the sqoop server during server start time along with its associated config objects.<br><br>It also has a upgrade path similar to connectors.<br><ul><li>/v1/driver - [GET]- Get Sqoop Driver</li></ul> |
| **CONFIG** | FORM | `MConfig.java` and @Config annotation | Top Level Entity | MConfigType with supported config types are<br><br>**MConfigType**<br><br>`public enum MConfigType {`<br>`  /** Unknown config type */`<br>`  OTHER,`<br>`  @Deprecated`<br>`  // NOTE: only exists to support the connector data upgrade path`<br>`  CONNECTION,`<br>`  /** link config type */`<br>`  LINK,`<br>`  /** Job config type */`<br>`  JOB;`<br>`}` | **READ ONLY, created once during the server start up, we do not allow update /delete via shell or REST**<br><br>*Note: We do not yet allow creating/deleting/editing configs at runtime via shell/ REST, and we will not probably do that ever since we want the config objects be declared in code via the @Config annotation. But Config and Inputs objects can be deleted as part of the configurable upgrade code path. Thus connector developers can delete/update it but not the sqoop users* |

| | | | | | |
|---|---|---|---|---|---|
| **INPUT ( Keys and Values )** | Same | `MInput.java` an `abstract class` and @Input annotation<br><br>Concrete classes for each supported types<br><br>`MIntegerInput.java`<br><br>`MStringInput.java` | • Associated with a **CONFIG** object | Represents the key-value pairs for a given config.<br><br>MInputTypes supported are<br><br>**MInputType**<br><br>`public enum MInputType {`<br>`  /** Unknown input type */`<br>`  OTHER,`<br>`  /** String input type */`<br>`  STRING,`<br>`  /** Map input type */`<br>`  MAP,`<br>`  /** Integer input type */`<br>`  INTEGER,`<br>`  /** Boolean input type */`<br>`  BOOLEAN,`<br>`  /** String based input that can contain only predefined values **/`<br>`  ENUM,`<br>`  ;`<br>`}` | **READ ONLY for Input Keys**<br><br>Input keys are created as configs are registered. We do not allow deletes/updates via the shell/REST.<br><br>**RU for Input values**<br><br>Input values can be edited per config object<br><br>*See SQOOP-1516 for rest apis related to config input Read/Updates per job/configId* |
| **LINK** | CONNECTION | `MLink.java`<br><br>`MLinkConfig.java` | • Associated with a **CONNECTOR**<br>• HAS a **CONFIG-INPUT** object | Represents the config inputs required to physically connect to the data-source a connector represents. Hence it is associated with a connector.<br><br>It has mainly one config object represented by MLinkConfig | **CRUD**<br><br>• /v1/links/ - [GET] Get all links<br>• /v1/links?cname=[cname] - [GET] Get all links by Connector<br>• /v1/link/[lname] or /v1/link/[lid] - [GET] - Get Link<br>• /v1/link - [POST] - Create Link<br>• /v1/link/[lname] or /v1/link/[lid] - [PUT] - Update Link<br>• /v1/link/[lname] or /v1/link/[lid] - [DELETE] - Delete Link<br>• /v1/link/[lid]/enable or /v1/link/[lname]/enable - [PUT] - Enable Link<br>• /v1/link/[lid]/disable - [PUT] - Disable Link |
| **JOB** | Same | `MJob.java`<br><br>`MFromConfig.java`<br><br>`MToConfig.java`<br><br>`MDriverConfig.java` | • HAS 3 **CONFIG-INPUT** objects<br>• HAS 1-n **SUBMISSIONS** | Represents the sqoop job. It encapsulates all the required configs to run the sqoop job.<br><br>Primarily the sqoop job has the 3 main components, the FROM, TO and the DRIVER.<br><br>FROM and its related MFromConfig represent the config-inputs-values required to Extract data from the source<br><br>TO and its related MToConfig represent the config-inputs-values required to load data to the destination<br><br>DRIVER and its related MDriverConfig the config-inputs-values required by the execution engine that runs the sqoop job optimally. | **CRUD**<br><br>• /v1/job - [POST] - Create Job<br>• /v1/job/[jid] - [PUT] - Update Job<br>• /v1/job/[jid] - [DELETE] - Delete Job<br>• /v1/job/[jid]/enable - [PUT] - Enable Job<br>• /v1/job/[jid]/disable - [PUT] - Disable Job<br>• /v1/job/[jid]/start or /v1/job/[jname]/start - [PUT]- Start Job<br>• /v1/job/[jid]/stop or /v1/job/[jname]/stop - [PUT]- Stop Job<br>• /v1/job/[jid]/status or /v1/job/[jname]/status - [GET]- Get Job Status |
| **SUBMISSION** | Same | `MSubmission.java` | | Represents the job run details. Includes the job status, job counters and metrics from the job execution engine | **READ ONLY**<br><br>• /v1/submissions? - [GET] - Get all job Submissions<br>• /v1/submissions?jname=[jname] - [GET] - Get Submissions by Job |

## Related tickets

Entity Renames: SQOOP-1497 && SQOOP-1498

### Rest API changes :

- SQOOP-1509

SQOOP-1516 ( scheduled for 1.99.5 though )

## Related Docs

https://issues.apache.org/jira/secure/attachment/12667274/SimplifySqoopEntityNomenclature.pdf

https://issues.apache.org/jira/secure/attachment/12667576/Sqoop2.pdf

https://issues.apache.org/jira/secure/attachment/12668107/SimplifySQOOPRESTAPIs.pdf