Wsruler Interface

API

This defines the REST based API offered to the user.

1. CRUD for Data Entities
2. CRUD for Link Entities
3. CRUD for Directory Service

1.CRUD for Data Entities

GET - read	POST - create	PUT - update	DELETE
/v1/ws/{id}	/v1/ws	/v1/ws/{id}	/v1/ws/{id}
/v1/env/{id}	/v1/env	/v1/env/{id}	/v1/env/{id}
/v1/repo/{id}	/v1/repo	/v1/repo/{id}	/v1/repo/{id}
/v1/db/{id}	/v1/db	/v1/db/{id}	/v1/db/{id}
/v1/ws/children/{id}			
/v1//env/children/{id}			

Note that Workspaces and Environments may have children.
Here are the **API** for *Querying Sub-components*:

- **/v1/ws/sub/{id}**
- **/v1/env/sub/{id}**

Note also that DELETE of a Data Entity will cause a **cascade** of deletion of Links that are associated with the deleted entity.

2. CRUD for Link Entities

Note that the **{id}** used in the API always refers to the Data Entity **{id}**.

GET	POST - link	DELETE - unlink
/v1/link/ws/{id}	/v1/link/ws	/v1/link/ws/{id}
/v1/link/env/{id}	/v1/link/env	/v1/link/env/{id}
/v1/link/repo/{id}	/v1/link/repo	/v1/link/repo/{id}
/v1/link/db/{id}	/v1/link/db	/v1/link/db/{id}

Examples of the semantics of the Link API are like the following:

- GET a Link entity with identified by **{id}**
- POST or Create a Link entity
- DELETE or Unlink an entity identified by **{id}**

3. CRUD for Directory Service

The Directory Service is not part of the WsRuler API. It is used by WsRuler only for retrieval of Owner-Group data.
For this demo, the CRUD for the Directory Service will simply model itself after the Wsruler Service data and link API. The GET API /dirsvc//olist is used by the WsRuler service. It will be pre-populated with Owner and Group via initialization script. The WsRuler service will not create Owners and Groups, but will access them through a special API to get the List of Owners entities for a given Group id.

GET dirsvc/olist/{groupname}

Data Entity API

GET	POST	PUT	DELETE
/v1/dirsvc_groups/{id}	/v1/dirsvc_groups	/v1/dirsvc_groups/{id}	/v1/dirsvc_groups/{id}
/v1/dirsvc_owners/{id}	/v1/dirsvc_owners	/v1/dirsvc_owners/{id}	/v1/dirsvc_ownesr/{id}

Link Entity API

GET	POST	PUT	DELETE
/v1/dirsvc_link/{id}	/v1/dirsvc_link	/v1/dirsvc_link/{id}	/v1/dirsvc_link/{id}

API Entities

1. Owner - Directory Service
2. Group - Directory Service
3. Workspace - Wsruler Service
4. Repository - Wsruler Service
5. Environment - Wsruler Service
6. Database - Wsruler Service
7. Link - Wsruler Service and Directory Service

8. Children - Wsruler Service

Note: POST requests do not need specify the **id** field. The **id** field will be calculated by the service as a unique string. Successful POST will return the new Object with the new **id**.

1.Directory Service Owner API Entity

id	name	email_address
Unique string	string	string

2. Directory Service Group API Entity

This record is what is stored for the Owners Group via POST or PUT:

id	name	groups
Unique string	string	List of strings

This record is returned for getting List of Owners for a Group:

id	name	owners
Unique string	string	List of Owner objects

3. Workspace API Entity

A. This is sent for POST

name	groups
string	List of strings

B. This is sent for PUT and returned by POST

id	name	groups

Unique string	string	List of strings

C. This is returned by GET

id	name	owners
Unique string	string	List of Owner objects

The Owner object in owners List is defined as:

id	name	email_address
Unique string	string	string

4. Repository API Entity

A. This is sent by POST

name
string

B. This sent by PUT

id	name
Unique string	string

B. This is is returned by GET and POST

id	name
Unique string	string

5. Environment API Entity

Same as for 4. Repository API Entity

6. Database API Entity

Same as for 4. Repository API Entity

7. Link API Entity

id	data_link	parent
Unique string	Id of an Data Entity	Id of the parent to the Data Entity

A Link Entity will have its own unique id. Its **data_link** field points to a Data Entity it represents. The **parent** field points to the Data Entity that represents the parent of the data_link entity.

The relationship rules will dictate that for any of the Data Entities for Environment, Repository, and Database there may be only 1 Link to link that Data Entity to its parent.

Env Link → 1 to 1 → Workspace Entity
Repo Link → 1 to 1 → Workspace Entity
DB Link → 1 to 1 → Environment Entity

The rules will allow multiple Link Entities for the Workspace to Group Data Entities since a Workspace may be "**owned**" by multiple Groups.

Workspace Link → many to 1 → Directory Service Group Entity

There can also be multiple links to map a Directory Service Owner Entity to more than 1 Group.

Owner Link → many to 1 → Group Entity

Having Link Entities allow for easy lookup by Data Entity **id** to accommodate Unlink or DELETE of the Data Entity.

8. Children API Entity

A. Sent with GET

id
Unique string

B. Returned by GET

id	names
Unique string	List of Child objects

Child Object is:

id	name	type
Unique string	string	String: *env* or *db* or *repo*

DB

DB Entities

1. General Data
2. Link

Only minimal number of "**tables**" or "**records**" needed. There are some CouchDB specific fields that are placed into all **records** but are not mentioned here.

1. General DB Entity

As CouchDB supports JSON directly, the API Entities described above will be represented in the DB accordingly. Differences include the **id** vs **_id** fields as CouchDB used **_id** as the unique identifier.

Many the API Entities for Wsruler Service defined above will fit into this "**table**".

_id	name
Unique string	string

This table shows what an Owner is defined as.

_id	name	email_address
Unique string	string	string

2. Link DB Entity

All links for both Wsruler Service and Directory Service will use this “**table**”.

_id	parent	data_link
Unique string	Id of parent entity	Id of General data entity