NODE's webapi paths

- http://<node-url>/api/v1/**tables**
  - **GET**: http://<node-url>/api/v1/**tables**/{tableName} ? numOfRows=<nOfRows> & attributes=<attrs> & filters=<filters>
    - executes a SELECT on a TABLE
  - POST: http://<node-url>/api/v1/**tables**
    - executes a CREATE TABLE, the node decides the name of the table
      data → {"schema":<schema>}
      returns the name of the table {"tableName":<tableName>}
  - PUT: …
  - **DELETE**: data→ {"tableName":<tableName>}

- http://<node-url>/api/v1/**views**
  - **GET**: http://<node-url>/api/v1/**views**/{viewName} ? numOfRows=<nOfRows> & attributes=<attrs> & filters=<filters>
    executes a SELECT on a VIEW
  - **POST**: http://<node-url>/api/v1/**views**
    executes a CREATE VIEW, the node decides the name of the view
    data → {"attributes":<attributes>, "tableName":<tableName>}
    retruns the name of the view {"viewName":<viewName>}
  - **DELETE**: data→ {"viewName":<viewName>}

- http://<node-url>/api/v1/**udfs**
  - **GET**: http://<node-url>/api/v1/**udfs**/?*type*=<python or pythonUDF>
    - *type* defines if the python or the pythonUDF syntax is requested
    - returns all the udf definitions available on this node
      example:

for ***type=python***

```
[{"udfName":<udfName1>,
  "definintion":"def <udfName1>(var1,var2,var3) … return result"},
 {"udfName":<udfName2>,
  "definintion":"def <udfName2>(var1,var2,var3) … return result"},
  ...]
```

for ***type=pythonUDF***

```
[{"udfName":<udfName1>,
  "definintions":[
         "CREATE FUNCTION <udfName1>(var1 FLOAT,var2 INT) RETURNS TABLE(r1 FLOAT,r2 FLOAT) LANGUAGE
         PYTHON{...return result},
         "CREATE FUNCTION <udfName1>(var1 FLOAT,var2 INT,var3 FLOAT) RETURNS TABLE(r1 FLOAT,r2 FLOAT) LANGUAGE
          PYTHON{...return result},
         "..."]},
 {"udfName":<udfName2>,
  "definintions":[
         "CREATE FUNCTION <udfName2>(var1 FLOAT,var2 INT) RETURNS TABLE(r1 FLOAT,r2 FLOAT) LANGUAGE
PYTHON{...return result},
         "CREATE FUNCTION <udfName2>(var1 FLOAT,var2 INT,var3 FLOAT) RETURNS TABLE(r1 FLOAT,r2 FLOAT) LANGUAGE
PYTHON{...return result},
         "..."]},
...]
```

    - this is mainly for debugging purposes

- **GET**: http://<node-url>/api/v1/**udfs**/{udfName}?*type*=<python or pythonUDF>
  - *type* defines if the python or the pythonUDF syntax is requested
  - returns all the function definitions, for this function name, available on this node

    for ***type=python***

    ```
    {“udfName”:<udfName1>,
     “definintion”:”def <udfName1>(var1,var2,var3) … return result”}
    ```

    for ***type=pythonUDF***

    ```
    {“udfName”:<udfName1>,
     “definintions”:[
              ”CREATE FUNCTION <udfName1>(var1 FLOAT,var2 INT) RETURNS TABLE(r1 FLOAT,r2 FLOAT) LANGUAGE
              PYTHON{...return result},
              ”CREATE FUNCTION <udfName1>(var1 FLOAT,var2 INT,var3 FLOAT) RETURNS TABLE(r1 FLOAT,r2 FLOAT) LANGUAGE
               PYTHON{...return result},
              ”...”]}
    ```

  - this is mainly for debugging purposes
- **POST**: http://<node-url>/api/v1/**udfs**
  - executes a UDF
  - data-> {“udfName”:<udfName>, “input”:[<table1Name,table2Name,...>]}
  - returns {“tableName”:<tableName>}, the table containing the udf result
- DELETE: ...