

C3 AI COVID-19 Data Lake Public API Tips and Tricks

Tips

Expression Engine Functions

Since the Public API is using the C3 AI Suite as its back-end it can be useful to know certain aspects of the C3 AI Suite, especially if you have access to the official documentation. The best example of this is Expression Engine Functions, which are used in various places throughout C3 AI Suite such as the 'filter' field of the fetch spec. These functions can allow you to create much better filters to ensure only the data you need is collected through the public API.

The full documentation for Expression Engine Functions can be found [here](#) if you have credentials to access C3 AI developer documentation.

A few highlights:

Simple boolean expressions like 'X < 3' and compound expressions like '(X < 3) && (Y > 2)' are supported.

The 'contains(str1, str2)' function returns true if str1 contains str2 as a substring within it.

Troubleshooting

Here are some issues you may encounter when working with the Public Data Lake API and their solutions.

Remember to quote strings in expressions within fetch commands.

This can result in: ERROR: syntax error at or near ")"

Example command data:

```
data = json.dumps({
  'spec': {
    'filter': 'id == Champaign_Illinois_UnitedStates', //error occurred
  }
})
```

Resulting Error:

```
{ 'cause': { 'message': 'ERROR: syntax error at or near ")"\n Position: 1207'},
  'codes': ['NotClassified'],
  'id': '4126.702231',
  'message': 'Unable to execute sql SELECT T.PUBLICHEALTHCARECENTERBE_SEN_I, '
    'T.HOSPITALICUBEDS_I, T.HOSPITALSTAFFEDBEDS_I, '
    'T.HOSPITALLICENSEDBEDS_I, T.LATESTTOTALPOPULATION_D, '
    'T.POPULATIONOFALLCHILDREN_D, T.ELEMENTARYSCHOOLCOUNT_I, '
    'T.KINDERGARTENCOUNT_I, T.UNIVERSITYCOUNT_I, '
    'T.NURSINGHOMECOUNT_I, T.ELDERLYPOPULATIONRATIO_D, '
    'T.ELDERLYALONERATIO_D, T.HOSPITALPREDICTION_TIMES_NQW_T, '
    'T.HOSPITALPREDICTION_TIMES_NQW_I, '
    'T.LATESTKINDERGARTENPOPULA_Y9R_D, T.LOCATIONTYPE_S, '
    'T.POPULATIONCDS_I, T.COUNTRYAREA_D, T.COUNTRYCODE_S, '
    'T.LOCATION_VALUE_R, T.LOCATION_TIMESTAMP_T, '
    'T.LOCATION_TIMESTAMP_I, T.DESCRPTION_L, '
    'T.RELATIVEPOSITION_REFEREN_KZD_R, '
    'T.RELATIVEPOSITION_XOFFSET_PN9_D, '
    'T.RELATIVEPOSITION_XOFFSET_WPP_R, '
    'T.RELATIVEPOSITION_YOFFSET_O0_D, '
    'T.RELATIVEPOSITION_YOFFSET_SRH_R, '
    'T.RELATIVEPOSITION_ZOFFSET_XBS_D, '
    'T.RELATIVEPOSITION_ZOFFSET_Y6F_R, '
    'T.RELATIVEPOSITION_TIMESTA_87W_T, '
    'T.RELATIVEPOSITION_TIMESTA_87W_I, T.FIPS_R, T.POPULATION2018_I, '
    'T.ID, T.NAME, T.TENANT_TAG_ID, T.CREATED, T.CREATED_I, '
    'T.CREATED_BY, T.UPDATED, T.UPDATED_I, T.UPDATED_BY, '
    'T.META_TIMESTAMP_T, T.META_TIMESTAMP_I, T.META_COMMENT, '
    'T.PROVENANCE, T.SOURCE, T.VERSION, T.TYPE_IDENT\n'
    'FROM C3_2_ASSET T\n'
    'WHERE T.TENANT_TAG_ID=:v1 AND ((T.TYPE_IDENT LIKE 'EP\\_LOC%') '
    'AND (T.ID=))\n'
    'ORDER BY T.ID\n'
    'LIMIT 2001\n'
    'ERROR: syntax error at or near ")"\n'
    ' Position: 1207'
```

The correct command has the string quoted:

```
data = json.dumps({
    'spec': {
        'filter': 'id == "Champaign_Illinois_UnitedStates"', // solving the error
    }
})
```