
Article

[Robert Cemper](#) · Feb 25, 2022 7m read

GlobalToJson-embeddedPython-pure#3

In my article I described the work using iris.gref .

As the [official documentation](#) is rather slim on the subject it was necessary to dig into it.

Using the power of Python I was able to detect what I needed but was hidden.

I decided to share this with you. pydoc did the magic.

And this is it:

```
>>> pydoc.help(iris)
Help on built-in module iris:

NAME
    iris
CLASSES
    builtins.object
        isc.StdoutType
        isc.gref      Global = class gref(builtins.object)
    | InterSystems IRIS global reference object.
    | Use the iris.gref() method to obtain a reference to a global

    Methods defined here:

        __delitem__(self, key, /)
            Delete self[key].

        __getitem__(self, key, /)
            Return self[key].

        __len__(self, /)
            Return len(self).

        __setitem__(self, key, value, /)
            Set self[key] to value.

        get(...)
            Given the keys of a global, returns the value stored at that node of the
global.
            Example: x = g.get([i,j]) sets x to the value stored at key i,j of global
g.

        getAsBytes(...)
            Given the keys of a global, returns a string stored at that node of the g
lobal, as bytes.
            Example: x = g.getAsBytes([i,j]) sets x to the value stored at key i,j of
global g, as bytes.

        keys(...)
            Traverses a global starting at the specified key, returning each key in t
he global.
            Example: for key in g.keys([i, j]) traverses g from key i,j, returning ea
```

```
ch key in turn.

    |
    | kill(...)
    |     Given the keys of a global, kills that node of the global and its subtree
    |
    |     Example: g.kill([i,j]) kills the node stored at key i,j of global g and a
    |     ny descendants.
    |
    | order(...)
    |     Given the keys of a global, returns the next key of the global.
    |     Example: j = g.order([i,j]) sets j to the next second-
    level key of global g.
    |
    | orderiter(...)
    |     Traverses a global starting at the specified key, returning the next key
    and value as a tuple.
    |     Example: for (key, value) in g.orderiter([i,j]) traverses g from key i,j,
    returning the next key and value.
    |
    | query(...)
    |     Traverses a global starting at the specified key, returning each key and
    value as a tuple.
    |     Example: for (key, value) in g.query([i,j]) traverses g from key i,j, ret-
    urning each key and value in turn
    |
    | set(...)
    |     Given the keys of a global, sets the value stored at that key of the glob-
    al.
    |     Example: g.set([i,j], 10) sets the value of the node at key i,j of global
    g to 10
    |
    | -----
    | Static methods defined here:

    __new__(*args, **kwargs) from builtins.type
        Create and return a new object.
        See help(type) for accurate signature. Stdout = class StdoutType(builtins
.object)
    |     isc.Stdout objects

    Methods defined here:

    flush(...)
        no-op

    isatty(...)
        false

    read(...)
        IRIS internal

    readl(...)
        IRIS internal

    readline(...)
        IRIS internal

    write(...)
        IRIS internal
```

```
-----  
| Static methods defined here:  
|  
|     __new__(*args, **kwargs) from builtins.type  
|         Create and return a new object. See help(type) for accurate signature.  
|  
-----  
| Data descriptors defined here:  
|  
|     encoding  
|         encoding  
FUNCTIONS  
check_status(...)  
    Raises an exception on an error status, or returns None if no error condition occurs.  
    Example: iris.check_status(st) checks the status code st to see if it contains an error.  
    cls(...)  
        Returns a reference to an InterSystems IRIS class.  
    Example: iris.cls("%SYSTEM.INetInfo").LocalHostName() calls a method in the class %SYSTEM.INetInfo.  
    gref(...)  
        Returns a reference to an InterSystems IRIS global.  
    Example: g = iris.gref("^foo") sets g to a reference to global ^foo  
    lock(...)  
        Sets locks, given a list of lock names, an optional timeout value (in seconds), and an optional locktype.  
    Example: iris.lock(["^foo", "^bar"], 30, "S") sets locks "^foo" and "^bar", waiting up to 30 seconds, and using shared locks.  
    ref(...)  
        Creates an iris.ref object with a specified value.  
    Example: iris.ref("hello") creates an iris.ref object with the value "hello"  
routine(...)  
    Invokes an InterSystems IRIS routine, optionally at a given tag.  
    Example: iris.routine("Stop^SystemPerformance", "20211221_160620_test") calls tag Stop in routine ^SystemPerformance.  
    tcommit(...)  
        Marks a successful end of an InterSystems IRIS transaction.  
    Example: iris.commit() marks the successful end of a transaction and decrements the nesting level by 1  
    tlevel(...)  
        Detects whether a transaction is currently in progress and returns the nesting level. Zero means not in a transaction.  
    Example: iris.tlevel() returns the current transaction nesting level, or zero if not in a transaction  
    trollback(...)  
        Terminates the current transaction and restores all journaled database values to their values at the start of the transaction.  
    Example: iris.rollback() rolls back all current transactions in progress and resets the transaction nesting level to 0  
    trollbackone(...)  
        Rolls back the current level of nested transactions, that is, the one initiated by the most recent tstart().  
    Example: iris.rollbackone() rolls back the current level of nested transactions and decrements the nesting level by 1  
    tstart(...)  
        Starts an InterSystems IRIS transaction.  
    Example: iris.tstart() marks the beginning of a transaction.  
    unlock(...)  
        Removes locks, given a list of lock names, an optional timeout value (in seconds), and an optional locktype.  
    Example: iris.unlock(["^foo", "^bar"], 30, "S") removes locks "^foo" and "^bar", waiting up to 30 seconds, and using shared locks.  
DATA  
sql = <iris.%SYS.Python.SQL object>  
utils = <iris.%SYS.Python_Utils object>  
FILE
```

```
(built-in) >>>
```

There are missing pieces and some strange behavior.
But definitely more important stuff than in the official documentation.
I didn't try everything. So there's a lot of room for your discoveries

[#Embedded Python](#) [#Globals](#) [#InterSystems IRIS](#)

Source URL:<https://community.intersystems.com/post/globaltojson-embeddedpython-pure3>