

```
In [1]: import dataiku
import pandas as pd, numpy as np
from dataiku import pandasutils as pdu
import json
import requests
```

```
In [2]: x= requests.get('https://api.data.abs.gov.au/data/ABS,ERP_Q,1.0.0/1.3+2+1.A10+A59+A04+A15+A20+A25+A30+A35+A40+A45+A50+A55+A60+A65+A70+A75+A80+8599+TOT..Q?startPeriod=2021-Q2', verify=False)
```

```

-----
KeyboardInterrupt                                Traceback (most recent call last)
<ipython-input-2-9fcba5fb6cc3> in <module>()
----> 1 x= requests.get('https://api.data.abs.gov.au/data/ABS,ERP_Q,1.0.0/1.3
+2+1.A10+A59+A04+A15+A20+A25+A30+A35+A40+A45+A50+A55+A60+A65+A70+A75+A80+8599
+TOT..Q?startPeriod=2021-Q2')

/opt/dataiku-dss-9.0.3/python36.packages/requests/api.py in get(url, params,
**kwargs)
    73
    74     kwargs.setdefault('allow_redirects', True)
--> 75     return request('get', url, params=params, **kwargs)
    76
    77

/opt/dataiku-dss-9.0.3/python36.packages/requests/api.py in request(method, u
rl, **kwargs)
    58     # cases, and look like a memory leak in others.
    59     with sessions.Session() as session:
--> 60         return session.request(method=method, url=url, **kwargs)
    61
    62

/opt/dataiku-dss-9.0.3/python36.packages/requests/sessions.py in request(sel
f, method, url, params, data, headers, cookies, files, auth, timeout, allow_r
edirects, proxies, hooks, stream, verify, cert, json)
    531     }
    532     send_kwargs.update(settings)
--> 533     resp = self.send(prepare_request(method, url, **kwargs), **send_kwargs)
    534
    535     return resp

/opt/dataiku-dss-9.0.3/python36.packages/requests/sessions.py in send(self, r
equest, **kwargs)
    644
    645     # Send the request
--> 646     r = adapter.send(request, **kwargs)
    647
    648     # Total elapsed time of the request (approximately)

/opt/dataiku-dss-9.0.3/python36.packages/requests/adapters.py in send(self, r
equest, stream, timeout, verify, cert, proxies)
    447         decode_content=False,
    448         retries=self.max_retries,
--> 449         timeout=timeout
    450     )
    451

/opt/dataiku-dss-9.0.3/python36.packages/urllib3/connectionpool.py in urlopen
(self, method, url, body, headers, retries, redirect, assert_same_host, timeo
ut, pool_timeout, release_conn, chunked, body_pos, **response_kw)
    675         body=body,
    676         headers=headers,
--> 677         chunked=chunked,
    678     )

```

679

```

/opt/dataiku-dss-9.0.3/python36.packages/urllib3/connectionpool.py in _make_request(self, conn, method, url, timeout, chunked, **httplib_request_kw)
424         # Python 3 (including for exceptions like SystemExit).
425         # Otherwise it looks like a bug in the code.
--> 426         six.raise_from(e, None)
427     except (SocketTimeout, BaseSSLError, SocketError) as e:
428         self._raise_timeout(err=e, url=url, timeout_value=read_timeout)

```

```

/opt/dataiku-dss-9.0.3/python36.packages/urllib3/packages/six.py in raise_from(value, from_value)

```

```

/opt/dataiku-dss-9.0.3/python36.packages/urllib3/connectionpool.py in _make_request(self, conn, method, url, timeout, chunked, **httplib_request_kw)
419         # Python 3
420         try:
--> 421             httplib_response = conn.getresponse()
422         except BaseException as e:
423             # Remove the TypeError from the exception chain in
n

```

```

/usr/lib64/python3.6/http/client.py in getresponse(self)
1344         try:
1345             try:
-> 1346                 response.begin()
1347             except ConnectionError:
1348                 self.close()

```

```

/usr/lib64/python3.6/http/client.py in begin(self)
305         # read until we get a non-100 response
306         while True:
--> 307             version, status, reason = self._read_status()
308             if status != CONTINUE:
309                 break

```

```

/usr/lib64/python3.6/http/client.py in _read_status(self)
266
267     def _read_status(self):
--> 268         line = str(self.fp.readline(_MAXLINE + 1), "iso-8859-1")
269         if len(line) > _MAXLINE:
270             raise LineTooLong("status line")

```

```

/usr/lib64/python3.6/socket.py in readinto(self, b)
584         while True:
585             try:
--> 586                 return self._sock.recv_into(b)
587             except timeout:
588                 self._timeout_occurred = True

```

```

/usr/lib64/python3.6/ssl.py in recv_into(self, buffer, nbytes, flags)
969         "non-zero flags not allowed in calls to recv_into()
on %s" %
970         self.__class__)

```

```

--> 971         return self.read(nbytes, buffer)
      972     else:
      973         return socket.recv_into(self, buffer, nbytes, flags)

/usr/lib64/python3.6/ssl.py in read(self, len, buffer)
      831         raise ValueError("Read on closed or unwrapped SSL socket.")
      832     try:
--> 833         return self._sslobj.read(len, buffer)
      834     except SSLError as x:
      835         if x.args[0] == SSL_ERROR_EOF and self.suppress_ragged_eofs:

/usr/lib64/python3.6/ssl.py in read(self, len, buffer)
      588         """
      589         if buffer is not None:
--> 590             v = self._sslobj.read(len, buffer)
      591         else:
      592             v = self._sslobj.read(len)

```

KeyboardInterrupt:

```
In [0]: print(x.status_code)
```

```
In [0]: out= json.loads(x)
```

```
In [0]: df = pd.DataFrame(x["results"]["records"])
```

```
In [0]: poc_ABS = dataiku.Dataset("POC_ABS")
```

```
In [0]: poc_ABS.write_with_schema(df)
```

```
In [4]: import os
hostname = "google.com" #example
response = os.system("ping -c 1 " + hostname)

#and then check the response...
if response == 0:
    print(hostname, 'is up!')
else:
    print(hostname, 'is down!')
```

google.com is down!