


TIGGE retrieval efficiency




- How is the TIGGE data organised in MARS?
- Best practise to iterate over all dates and all times for for a particular level type (eg Pressure levels)
- Web-API examples:
 - Example 1: Two TIGGE requests (sfc and pl), for some dates, for the same 'origin'
 - Example 2: A TIGGE sfc request for some dates, for several 'origins'
- Useful links

How is the TIGGE data organised in MARS?


 In general it is organised, as a **huge tree**, with the indentation below, showing different levels down that tree:

- type of level
 - date
 - time
 - steps, origins (ie Centres), members, parameters (same tape file 

Best practise to iterate over all dates and all times for for a particular level type (eg [Pressure levels](#))

  The idea is to request as much data as possible from the same tape file .


The best approach is to iterate over the dates and for each date iterate over the times. At this point you may wish to [check TIGGE availability](#).




```
for date in dates
  for time in times
    TIGGE-request(date, time, origin)
    (Here you must add everything you need)
```

Web-API examples:

Example 1: Two TIGGE requests (sfc and pl), for some dates, for the same 'origin'

- 
- The objective of this example is to **demonstrate** how to iterate **efficiently** over some dates (eg '2014-10-01' and '2014-11-01') two requests for 'perturbed forecast', 'pressure' and 'sfc' level, for TIGGE, ECMWF Center.
 - It can be used as a **starting point**, please keep in mind that you have to adapt it to your needs eg to **set** the **keyword, values** according to your requirements ('param', 'levtype', 'step' etc).
 - In this way you can **extend** these requests to download more data or a longer period of TIGGE.

 The most **efficient way** to **organise** your requests is **per dates and times**

ie select a set of 'dates' and 'times' and add the 'steps', 'origins', 'levels' and 'parameters' that you **need**.

 It is **not efficient** to organise your requests per **parameter**, per **origin** or per **step**. (eg to submit a different request per parameter or per origin for a list of dates)

```

#!/usr/bin/env python
from ecmwfapi import ECMWFDataServer
server = ECMWFDataServer()

def retrieve_tigge_data():
    dates = ['2014-10-01', '2014-11-01']
    times = ['00', '12']
    for date in dates:
        for time in times:
            target = 'ecmwf_sfc_%s_%s.grb' % (date, time)
            tigge_pf_sfc_request(date, time, target)
            target = 'ecmwf_pl_%s_%s.grb' % (date, time)
            tigge_pf_pl_request(date, time, target)

def tigge_pf_pl_request(date, time, target):
    """
    A TIGGE request for perturbed forecast, pressure level, ECMWF Center.
    Please note that a subset of the available data is requested below.
    Change the keywords below to adapt it to your needs. (ie to add more parameters, or numbers etc)
    """
    server.retrieve({
        'class': 'ti',
        'dataset': 'tigge',
        'date': date,
        'expver': 'prod',
        'grid': '0.5/0.5',
        'levelist': '200/250/300/500/',
        'levtype': 'pl',
        'number': '1/TO/10',
        'origin': 'ecmf',
        'param': '130/131/132/133',
        'step': '0/TO/96/BY/12',
        'target': target,
        'time': time,
        'type': 'pf',
    })

def tigge_pf_sfc_request(date, time, target):
    """
    A TIGGE request for perturbed forecast, sfc, ECMWF Center.
    Please note that a subset of the available data is requested below.
    Change the keywords below to adapt it to your needs. (ie to add more parameters, or numbers etc)
    """
    server.retrieve({
        'class': 'ti',
        'dataset': 'tigge',
        'date': date,
        'expver': 'prod',
        'grid': '0.5/0.5',
        'levtype': 'sfc',
        'number': '1/TO/10',
        'origin': 'ecmf',
        'param': '146/147/151/165/166/167/168/176/177',
        'step': '0/TO/96/BY/12',
        'target': target,
        'time': time,
        'type': 'pf',
    })

if __name__ == '__main__':
    retrieve_tigge_data()

```

Example 2: A TIGGE sfc request for some dates, for several 'origins'



💡 If you wish to download the **same data** for **more than one origins** it is more **efficient** to request **all of them** in one go.

- The objective of this example is to **demonstrate** how you can get the same data for several origins in one request.
- Before you start writing your request, you need to check that the requested data are available on all the selected 'origins'.

```
#!/usr/bin/env python
from ecmwfapi import ECMWFDataServer
server = ECMWFDataServer()

def retrieve_tigge_data():
    dates = ['2014-12-01', '2014-12-02']
    times = ['00', '12']
    for date in dates:
        for time in times:
            target = 'tigge_cf_sfc_%s_%s.grb' % (date, time)
            tigge_cf_sfc_request(date, time, target)

def tigge_cf_sfc_request(date, time, target):
    '''
    A TIGGE request for control forecast, sfc, for 3 origins : ECMWF, JMA and KMA.
    Keep in mind that if you wish to download the same data, for more than one origins,
    it is more efficient to request all of them in one go.
    You can change the keywords below to adapt it to your needs,
    (ie to add more parameters, or steps, or even more origins etc),
    Presumably you need to check the availability of the requested origins.
    '''
    server.retrieve({
        'class': 'ti',
        'dataset': 'tigge',
        'date': date,
        'expver': 'prod',
        'grid': '0.5/0.5',
        'levtype': 'sfc',
        'origin': 'ecmf/rjtd/rksl',
        'param': '167/168',
        'step': '0/96/168',
        'target': target,
        'time': time,
        'type': 'cf',
    })

if __name__ == '__main__':
    retrieve_tigge_data()
```

Useful links



- [ECMWF Web API Home](#)
- [TIGGE availability](#)
- [TIGGE Home](#)