LiveDemo

January 11, 2019

```
In [ ]: import warnings

        from IPython.display import HTML
        import matplotlib
        from matplotlib import pyplot as plt
        import numpy as np
        plt.rcParams['figure.figsize'] = (20.0, 10.0)
        matplotlib.rc('image', cmap='RdYlBu')

        warnings.filterwarnings("ignore")
        %matplotlib inline
```

```
In [ ]: #(Open dataset)
        import karabo_data as kd
        exmpl_file = '/gpfs/exfel/exp/XMPL/201750/p700000/proc/r0273/CORR-R0273-AGIPD03-S00000.h
        hdf5_file = kd.H5File(exmpl_file)
```

Back to talk

```
In [ ]: #Live-demo (sel from train-id)
        train_id, train_data = hdf5_file.train_from_id(198425246)
        print(train_id, train_data.keys())
```

```
In [ ]: #Live-demo (sel from index)
        train_id, train_data = hdf5_file.train_from_index(5)
        print(train_id, train_data.keys())
```

```
In [ ]: #Live-demo (iteration)
        for train_id, train_data in hdf5_file.trains():
            print(train_id, train_data.keys())
            break
```

Train data is of type dictionary. Hence the data can be accessed by giving keys:

```
In [ ]: #Live-demo show-dict
        train_data['SPB_DET_AGIPD1M-1/DET/3CH0:xtdf']
```

```
In [ ]: #Live-demo data array and plot
        print(train_data['SPB_DET_AGIPD1M-1/DET/3CH0:xtdf']['image.data'].shape)
        print(type(train_data['SPB_DET_AGIPD1M-1/DET/3CH0:xtdf']['image.data']))
        from matplotlib import pyplot as plt
        fig = plt.figure(figsize=(10,10))
        _ = plt.imshow(train_data['SPB_DET_AGIPD1M-1/DET/3CH0:xtdf']['image.data'][0].T,
                       vmin=-50, vmax=100)
```

Back to talk

```
In [ ]: #Live-demo run dir
        run_folder = '/gpfs/exfel/exp/XMPL/201750/p700000/raw/r0273'
        run_dir = kd.RunDirectory(run_folder)
        run_dir.info()
```

Back to talk

```
In [ ]: #Live-demo select with glob pattern
        sel = run_dir.select('*/DET/*', 'image.*')
        sel.all_sources
```

Data can be access just like *previously* mentioned:

```
In [ ]: #Live-demo (iterator example)
        for train_id, train_data in sel.trains():
            #print(train_id, train_data.keys())
            break
        train_id, train_data = sel.train_from_index(5)
        train_id, train_data.keys()
```

Back to talk

```
In [ ]: #Live-demo get series and plot
        ph_flux = run_dir.get_series('SA1_XTD2_XGM/XGM/DOOCS', 'pulseEnergy.photonFlux.value')
        ph_flux.head()
        fig = plt.figure(figsize=(10,5))
        _ = ph_flux.plot()
```

Back to talk

```
In [ ]: #Live-demo get dataframe
        fluxes_pos = run_dir.get_dataframe(fields=[("*_XGM/*", "*.photonFlux"),
                                                   ("*_XGM/*", "*.i[xy]Pos")])
        fluxes_pos.head(10)
```

Back to talk

```
In [ ]: #Live-demo get array
        xgm_intensity = run_dir.get_array('SA1_XTD2_XGM/XGM/DOOCS:output', 'data.intensityTD',
                                          extra_dims=['pulseId'])
        xgm_intensity
```

```
In [ ]: #Live-demo plot xgm data
        fig = plt.figure(figsize=(10,8))
        _= plt.imshow(xgm_intensity[:,:120].T,
                      extent=(xgm_intensity.trainId[0], xgm_intensity.trainId[-1], 1, 120),
                      origin='lower', cmap='RdYlBu_r')
```