LiveDemo

January 15, 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
        import karabo_data as kd
        run_folder = '/gpfs/exfel/exp/XMPL/201750/p700000/raw/r0273'
        exmpl_file = '/gpfs/exfel/exp/XMPL/201750/p700000/proc/r0273/CORR-R0273-AGIPD03-S00000.h
        hdf5_file = kd.H5File(exmpl_file)
        run_dir = kd.RunDirectory(run_folder)
```

```
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 [ ]: train_id, train_data = hdf5_file.train_from_index(5)
```

```
In [ ]: train_data.keys()
```

```
In [ ]: detector_data = train_data['SPB_DET_AGIPD1M-1/DET/3CH0:xtdf']
        sorted(detector_data.keys())
```

```
In [ ]: detector_image = detector_data['image.data']
```

```
In [ ]: detector_image
```

```
In [ ]: detector_image.shape
```

```
In [ ]: from matplotlib import pyplot as plt
        plt.imshow(detector_image[0])
```

```
In [ ]: fig = plt.figure(figsize=(10,5))
        plt.imshow(detector_image[0].T, vmin=-50, vmax=100)
```

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

Back to talk

```
In [ ]: sorted(run_dir.selection.keys())
```

```
In [ ]: #Live-demo (get series)
        ph_flux = run_dir.get_series('SA1_XTD2_XGM/XGM/DOOCS', 'pulseEnergy.photonFlux.value')
        ph_flux.head()
```

```
In [ ]: ph_flux.plot()
```

Back to talk

```
In [ ]: xgm_sa1 = run_dir.select('SA1_XTD2_XGM/XGM/DOOCS*', '*')
        xgm_spb = run_dir.select('SPB_XTD9_XGM/XGM/DOOCS*', '*')
```

```
In [ ]: xgm_union = xgm_sa1.union(xgm_spb)
        xgm_union.selection.keys()
```

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

```
In [ ]: fluxes_pos.plot.scatter(x='SA1_XTD2_XGM/XGM/DOOCS/beamPosition.iyPos', y='SPB_XTD9_XGM/X
```

Back to talk

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

```
In [ ]: plt.imshow(xgm_intensity[:,:120].T)
```

```
In [ ]: plt.imshow(xgm_intensity[:,:120].T, origin='lower', extent=(xgm_intensity.trainId[0], xg
                   cmap='RdYlBu_r')
```