Pixiedust

GitHub - pixiedust/pixiedust: Python Helper library for Jupyter Notebooks

PixieDust's current capabilities include:

packageManager lets you install Spark packages inside a Python notebook. This is something that you can't do today on hosted Jupyter notebooks, which prevents developers from using a large number of spark package add-ons.

Visualizations. One single API called display() lets you visualize your Spark object in different ways: table, charts, maps, etc.... This module is designed to be extensible, providing an API that lets anyone easily contribute a new visualization plugin.

Spark progress monitor. Track the status of your Spark job. No more waiting in the dark. Notebook users can now see how a cell's code is running behind the scenes.

Pixiedust implements a custom `display` function and registers it in the ipython context. This is how they are able to offer custom rendering of spark dataframes with a rich interactive UI like databricks.

Display Data — PixieDust Documentation

Pixiedust also registers event listeners on the spark context so that they can monitor job execution and render the status in a notebook.

Spark Progress Monitor — PixieDust Documentation

Install

```
pip3 install pixiedust
# run python in verbose to see where the package got installed
python3 -v
>>> import pixiedust

# Then edit this file to remove the check for spark version
# (pixiedust only supports version 2 of spark, not version 3)
vi /opt/conda/lib/python3.8/site-packages/pixiedust/display/datahandler/init.py

#if Environment.hasSpark:
from .pysparkDataFrameHandler import PySparkDataFrameDataHandler

```

Caveats

- Only works in classic notebook
- Issues count query first then the query for fetching data
- Runs a sampling (but the ui suggest it can be turned off)
- Not all UI functionally works correctly

This project might be interesting as a base to build custom visualization of dataframes.