
wzdat Documentation

Release 0.1

JeongJu Kim

August 18, 2015

1	User's Guide	3
1.1	Foreword	3
1.2	What's new in WzDat 0.1?	3
1.3	Requirements	4
1.4	Solution And Projects	4
1.5	Ignition	5
1.6	Tutorial	6
2	Additional Notes	7
2.1	WzDat Changelog	7
3	Indices and tables	9

WzDat stands for **Webzen Data Analysis Toolkit** (pronounce it like “What’s that?”), which started as an attempt to build **IPython** & **Pandas** based data analysis system. WzDat augments the power of IPython & Pandas.

Currently WzDat consists of three applications. WzDat Python Module, Dashboard and **WzDat Forwarder for Windows**. This document is for WzDat Python Module & Dashboard.

1.1 Foreword

If you are a Python enthusiast or data scientist, you might have been heard about the [IPython](#) and [Pandas](#). Mixing them together forms one of the most beloved tools in this data centric era. IPython gives us nice interactive Python programming environment, and Pandas provides easy-to-use professional data tools. They are convinient and powerful tools not only for casual data work, but for big companies and laboratories. WzDat was started to empower this greate tool even more.

1.1.1 Profits

Let's say you have thousands of log files, want to find some of them by their traits and specific words it contains, then feed the result to Pandas for more delicate analysis work. How would you do that? Some scripts or shell commands might be helpful, but the process is(in general) cumbersome and error prone to repeat everytime you get new files.

That's where WzDat comes in handy. Once WzDat Python module & its compliant project are imported under the IPython notebook environment, It will pick out your target files with impressive speed even from hundreds of thounds of files.

When you finished your analysis work in the IPython notebook, you can forwarding your outcome to WzDat Dash-board, from where your colleagues share your work instantly with concise web interface.

1.1.2 Docker Depedent

[Docker](#) is a great platform which solves complex deploy problems. Since WzDat is one of the building blocks to construct an IPython & Pandas based data analysis system, there are much more softwares, configs, admin works to build one. To make deployment simpler, I choose Docker as default platform. Though it may sound strange, WzDat is premised on Docker, and its code presumes certain environments(folders, files, packages) provided by its Docker container.

1.2 What's new in WzDat 0.1?

This is initial public release.

1.3 Requirements

1.3.1 Linux System

WzDat has been developed and tested on dockerized Linux(Ubuntu). You can deploy WzDat server under OSX or Windows host by using tools like Boot2Docker, but unknown defects could arise.

Warning: Mounting directory from OSX or Windows to docker container would not work.

1.3.2 WzDat Solution & Project

Before using WzDat to analyze your data, you need your **solution** and **project**. Refer *Solution And Projects* and make your own solution & project.

1.4 Solution And Projects

Your WzDat solution is a place where your projects exist. Your WzDat projects are place for your domain specific **file adapters** and utilities reside.

Note: There are many log/data formats. They have different file naming rule, date time and log level format. To mitigate these differences, **file adapter** plays role. Refer to the following section for further details.

1.4.1 Creating The Skeleton

Let's start by a barebone folders & files:

```
ws_mysql/  
  ws_mysql/  
    __init__.py  
  myprj/  
    __init__.py
```

In this case, your solution name is `mysql` (Preferably, solution nams comes from your personal name, or company name), then make topmost `ws_mysql` as solution folder, and inner `ws_mysql` folder, and as **solution python package**. In the solution package, there is a WzDat **project python package** `myprj`.

Tip: I advise you to start your solution with `ws_` prefix(`ws` stands for **WzDat Solution**).

As need arises, you can append more projects to your solution.

1.4.2 File Adapter

File adapter is a python module in which you implement functions to feed required information about the file format you are dealing. For example, if you want your WzDat project can handle two types of file(`log`, `dblog`), your project might look like this:


```
ws_mysql/
  ws_mysql/
    __init__.py
  myprj/
    __init__.py
    log.py      <--
    dbdump.py   <--
```

File Type	File Extension	Adapter
Log	.log	log.py
DB Dump	.csv	dbdump.py

Each adapter module is asked to implement following functions:

1.4.3 Configs

To feed setting information, config files are located at your solution and projects:

```
ws_mysql/
  ws_mysql/
    __init__.py      <--
    config.yaml
  myprj/
    __init__.py
    config.yaml      <--
    log.py
    dbdump.py
```

If some settings are common among your projects, you can place them at solution config file. If some others are specific for a certain project, create project config file.

1.4.4 Notes Folder

Finally, create `__notes__` folder to accomodate your IPython Notebooks, and create nested folder per project:

```
ws_mysql/
  __notes__      <--
  myprj/         <--
  ws_mysql/
    __init__.py
    config.yaml
  myprj/
    __init__.py
    config.yaml
    log.py
    dbdump.py
```

When you create new IPython notebook, it will be directly put into `ws_mysql/__notes__/myprj`.

1.5 Ignition

Warning: WzDat requires compliant solution & project to use it as data analysis system. If this is your first time to setup WzDat, visit [Solution And Projects](#) page to get acquaint with them.

After you are done with making solution & project, clone `wzdat-sys` and build local docker image.

```
$ git clone https://github.com/haje01/wzdat-sys
$ cd wzdat-sys
$ sys/build.sh
```

Replace (..) variables with your own, then run script.

```
$ WZDAT_HOST=(server-host-name) \
> WZDAT_DATA_DIR=(data-folder) \
> WZDAT_SOL_DIR=(solution-folder) \
> WZDAT_SOL_PKG=(solution-package-name) \
> WZDAT_PRJ=(project-id) \
> WZDAT_IPYTHON_PORT=(ipython-port) \
> WZDAT_DASHBOARD_PORT=(dashboard-port) \
> sys/run.sh
```

1.6 Tutorial

Let's have a quick glimpse of WzDat. In this tutorial, we're going to build a simple solution & project to analyze Linux Syslog.

1.6.1 Build Docker Container

1.6.2 Make Solution & Project

1.6.3 Data Directory

1.6.4 Deploy By Docker

After you are done with the solution & project, clone `wzdat-sys` and build local docker image.

```
$ git clone https://github.com/haje01/wzdat-sys
$ cd wzdat-sys
$ sys/build.sh
```

Replace (..) variables with your own, then run script.

```
$ WZDAT_HOST=my.host.com\
$ WZDAT_DATA_DIR=/home/myhome/data\
$ WZDAT_SOL_DIR=/home/myhome/ws_mysql\
$ WZDAT_SOL_PKG=ws_mysql\
$ WZDAT_PRJ=myprj\
$ WZDAT_IPYTHON_PORT=8085\
$ WZDAT_DASHBOARD_PORT=8080\
$ sys/run.sh
```

1.6.5 Select, Find And Analyze

1.6.6 Expose Result To Dashboard

Additional Notes

2.1 WzDat Changelog

2.1.1 Version 0.1.0

Release on September 1, 2014.

First public release.

Indices and tables

- `genindex`
- `modindex`
- `search`