Big Data Energy Services Analytics Portal

Documentation

Author: Jonathan Bennett, Kono Analytics

Version: 1.1.1

Date: 5/7/2018

This Document applies to version 1.1.1 of the BDES Application portal located at <https://solar.bigdataenergy.com>

Contents

[User Administration 3](#_Toc512325919)

[Adding a User 3](#_Toc512325920)

[Changing a User’s Password or Making the User an Administrator 6](#_Toc512325921)

[User Permission Level 7](#_Toc512325922)

[Graph Required Access Level 9](#_Toc512325923)

[Application Navigation 12](#_Toc512325924)

[Aggregated Hourly Generation 12](#_Toc512325925)

[Aggregated Hourly Cash Flow 14](#_Toc512325926)

[Monthly Cash Flow 15](#_Toc512325927)

[Cash Waterfall 17](#_Toc512325928)

[Revenue Ranking 19](#_Toc512325929)

[Solar Ranking 20](#_Toc512325930)

[Data Flow 21](#_Toc512325931)

[Notes 21](#_Toc512325932)

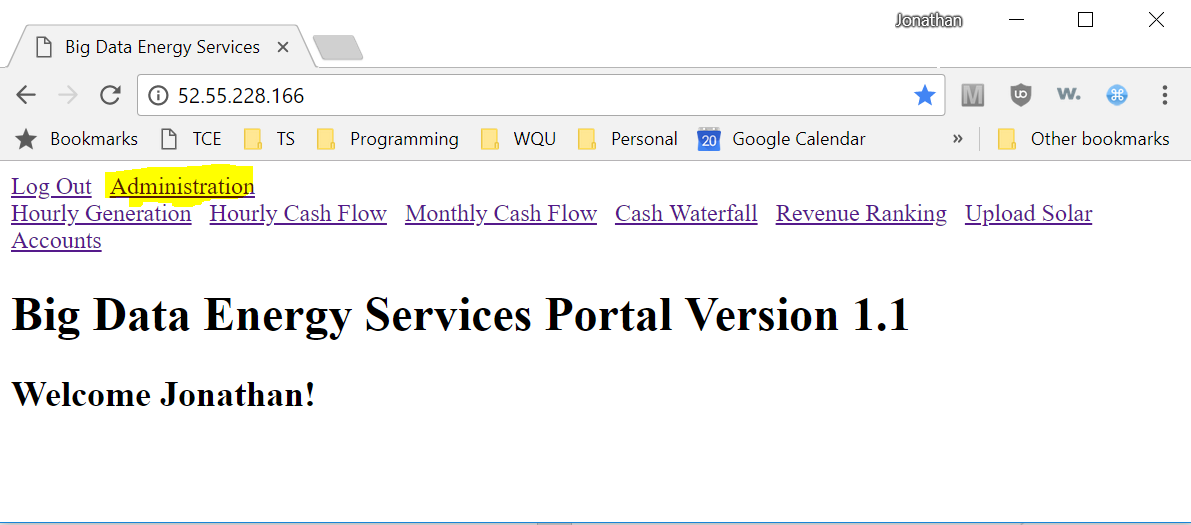
[To Do 21](#_Toc512325933)

[Possible Add-Ons 21](#_Toc512325934)

# User Administration

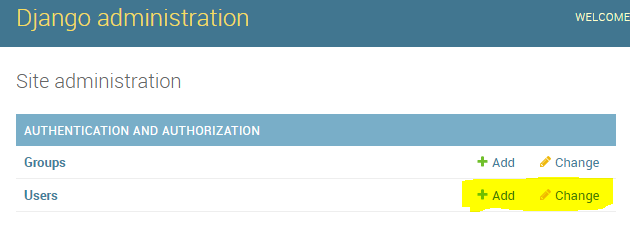
Only users with Administrator access can create and edit users. The administrative panel is used to make these changes. Click the “Administration” link

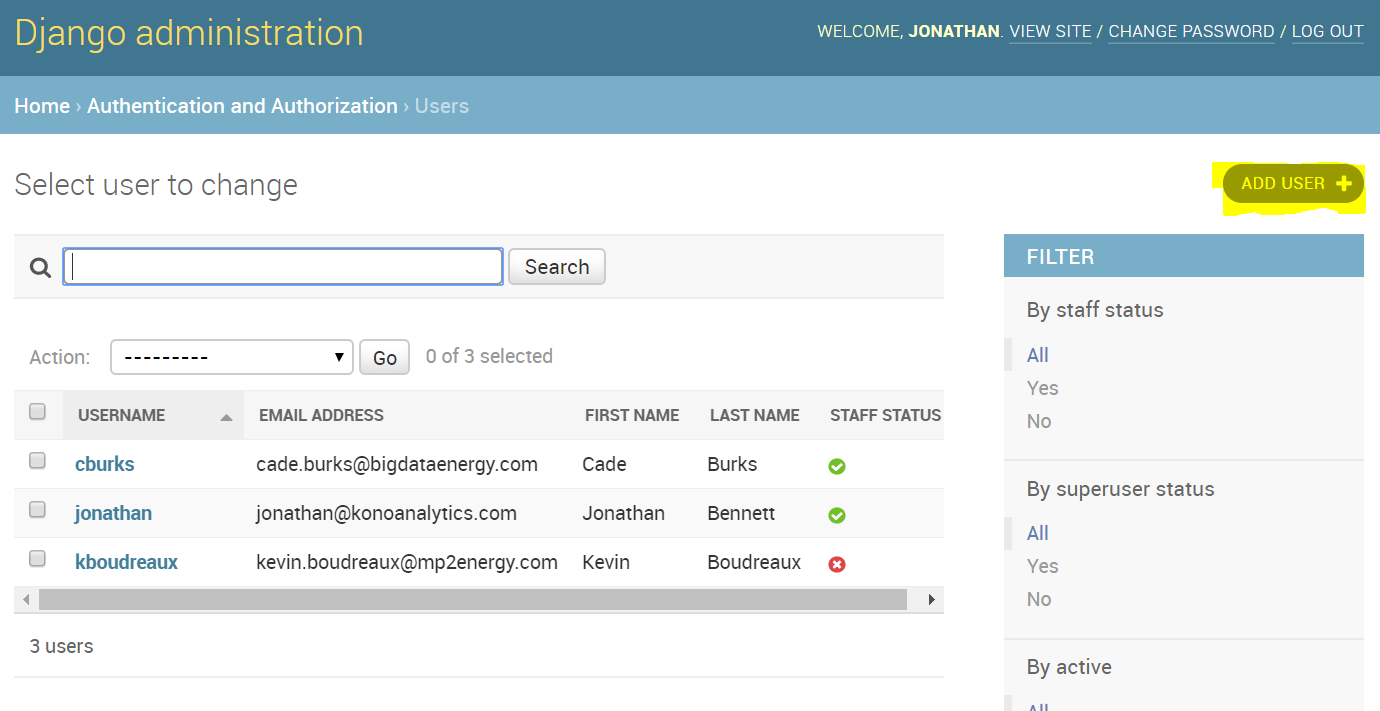
<https://solar.bigdataenergy.com/>

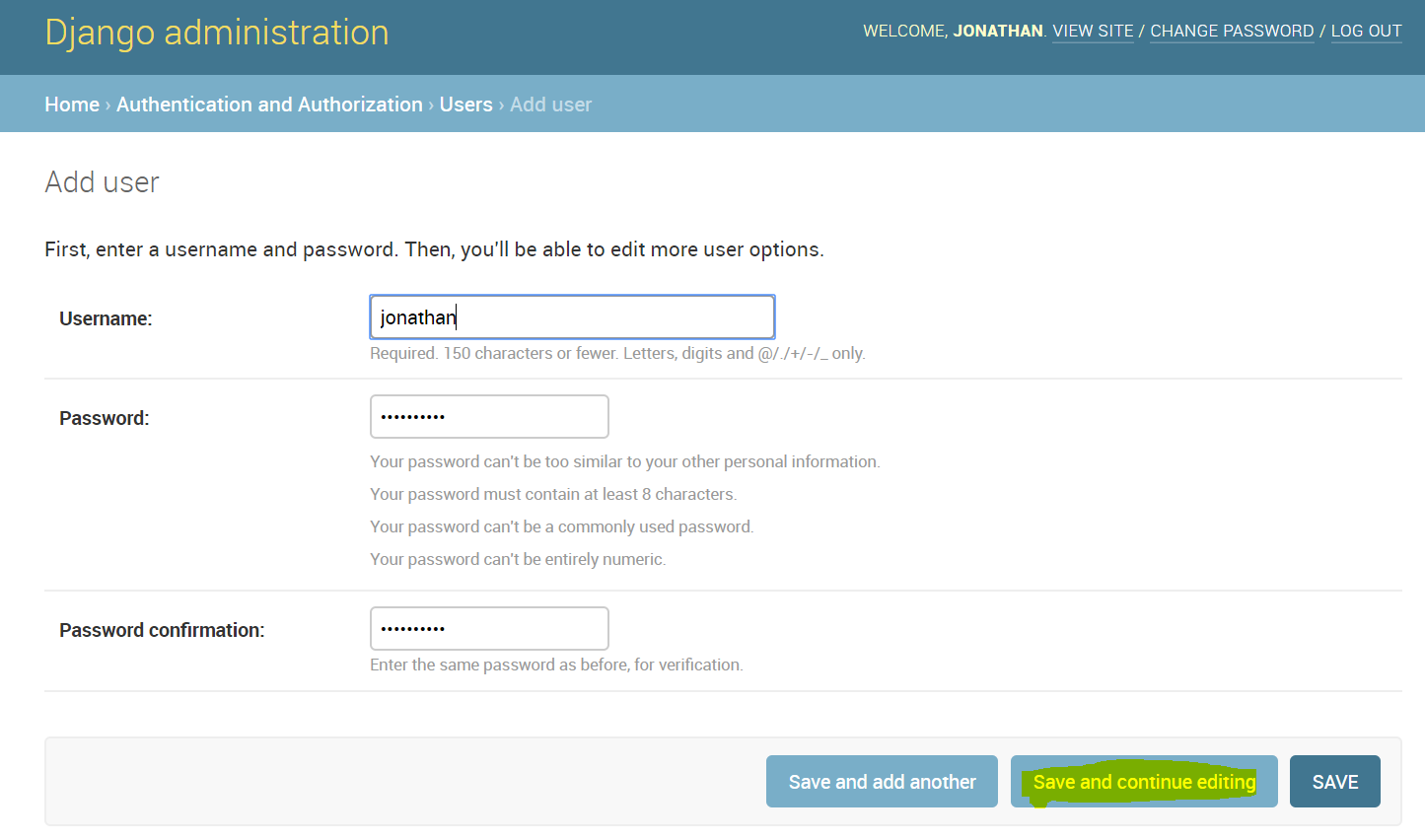


## Adding a User

After logging in you can add here:



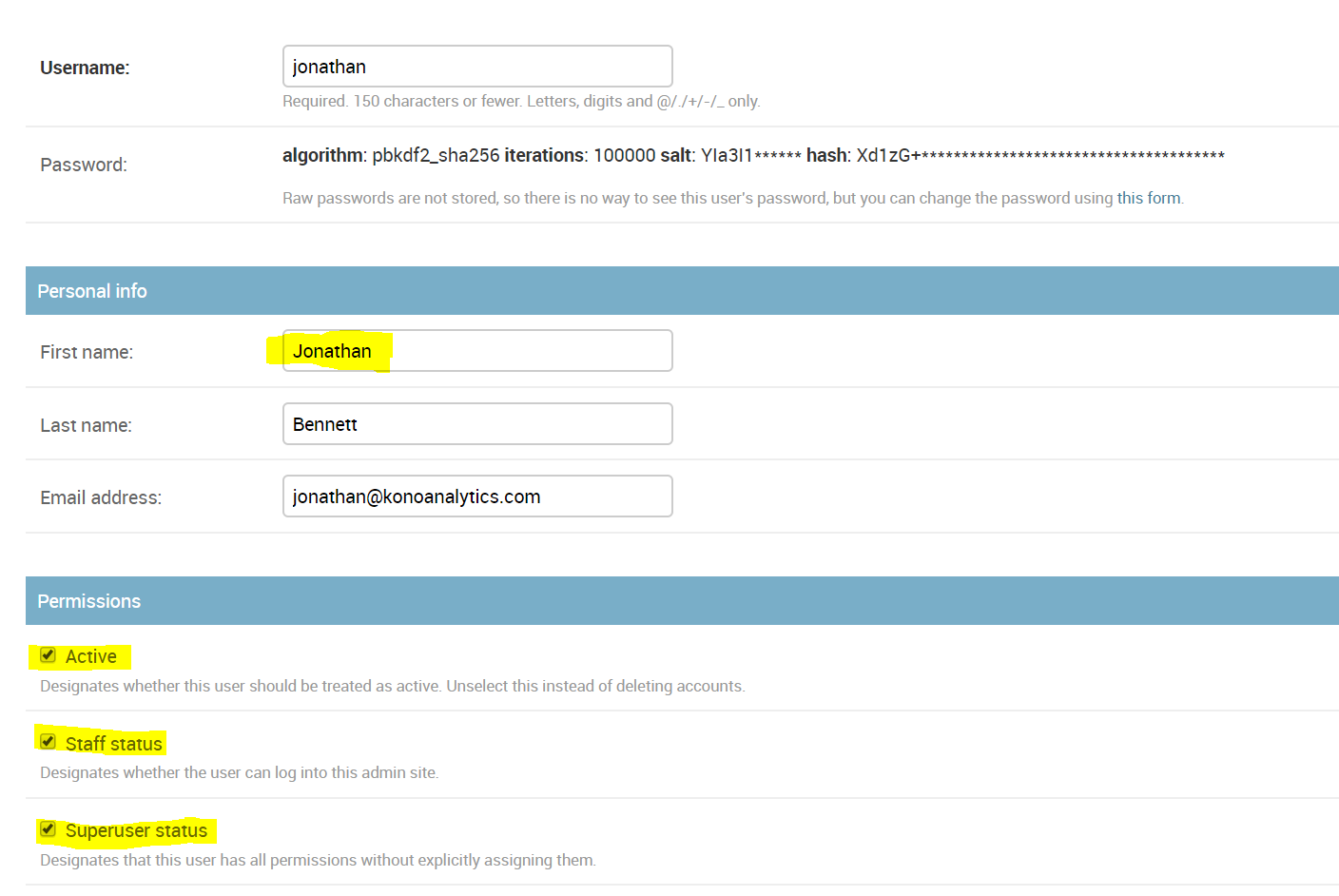




It’s important to fill out the first name because it’s displayed on the home screen.

Active can be checked and unchecked to quickly disable/enable access. You can delete the user by clicking the “Delete” button on the bottom of the screen, but it’s advisable to uncheck the “Active” bok and not actually delete the users’ records.

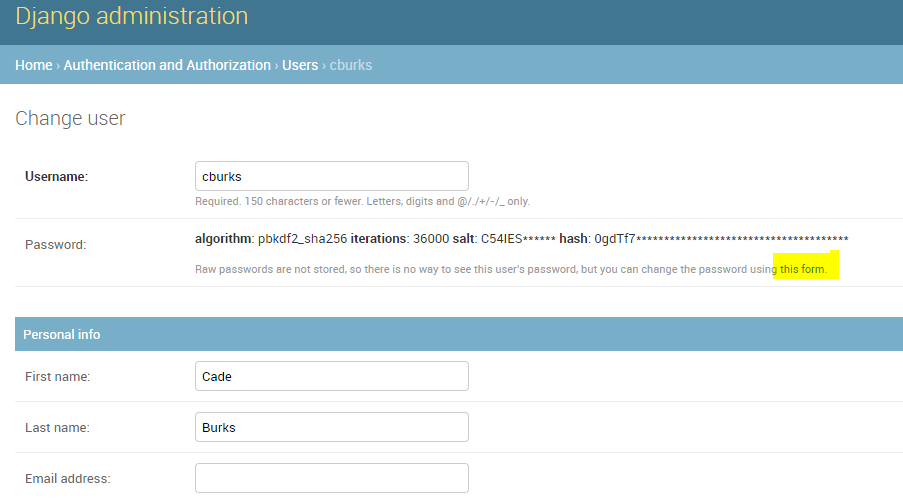
Only BDES Employees should have the “Staff status” and the “Superuser status” checked. This emables access to the administration panel and shows the administration hyperlink.



## Changing a User’s Password or Making the User an Administrator

From the main panel, click “Change” under “Users” and then click the account to modify.

To change the password, click this link:

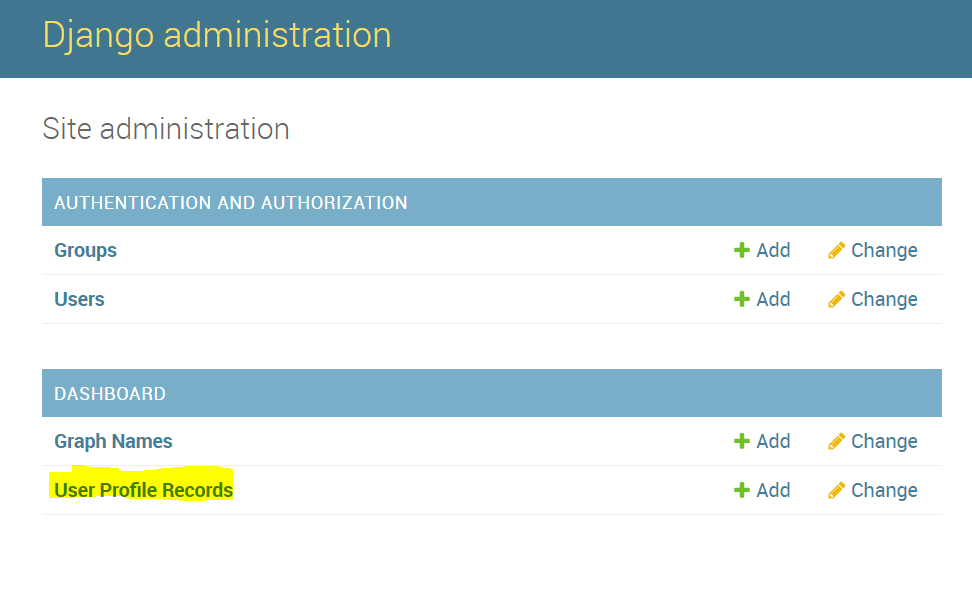


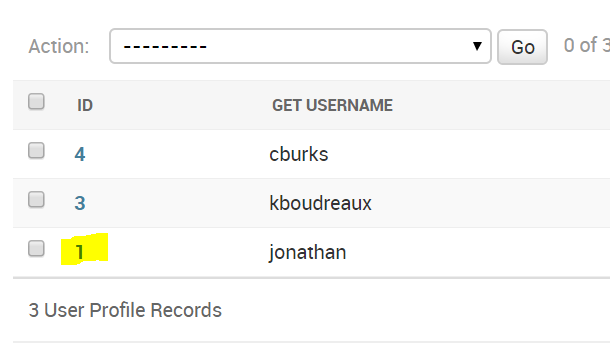
## 

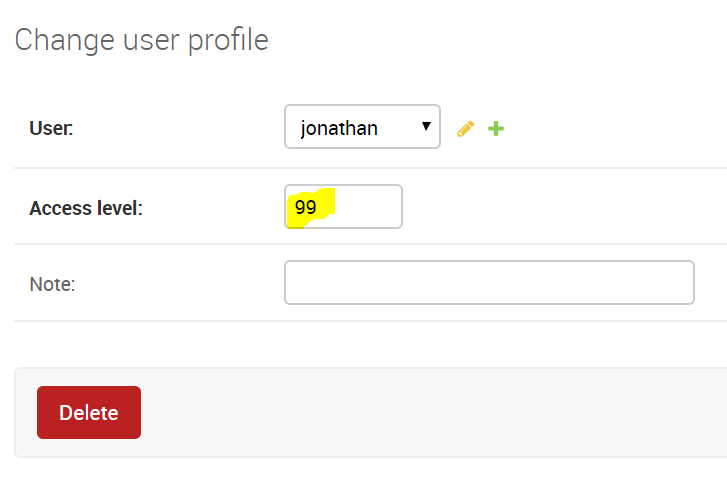
## User Permission Level

The ability of a user to see specific aspects of the application depends on two factors. Their own access level, and the required level of each individual function. If the user’s access level is at leas as high as the graph, then they will be able to see and use it.

To set the user access level:

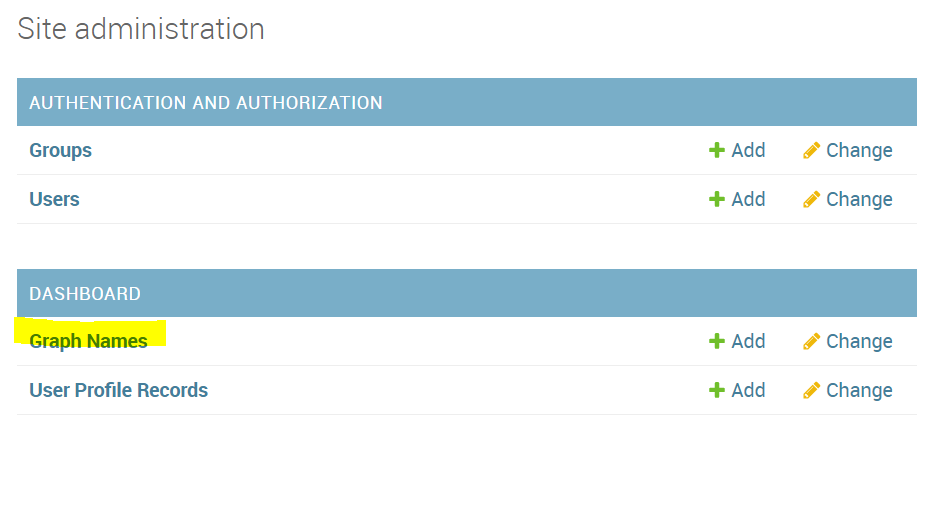


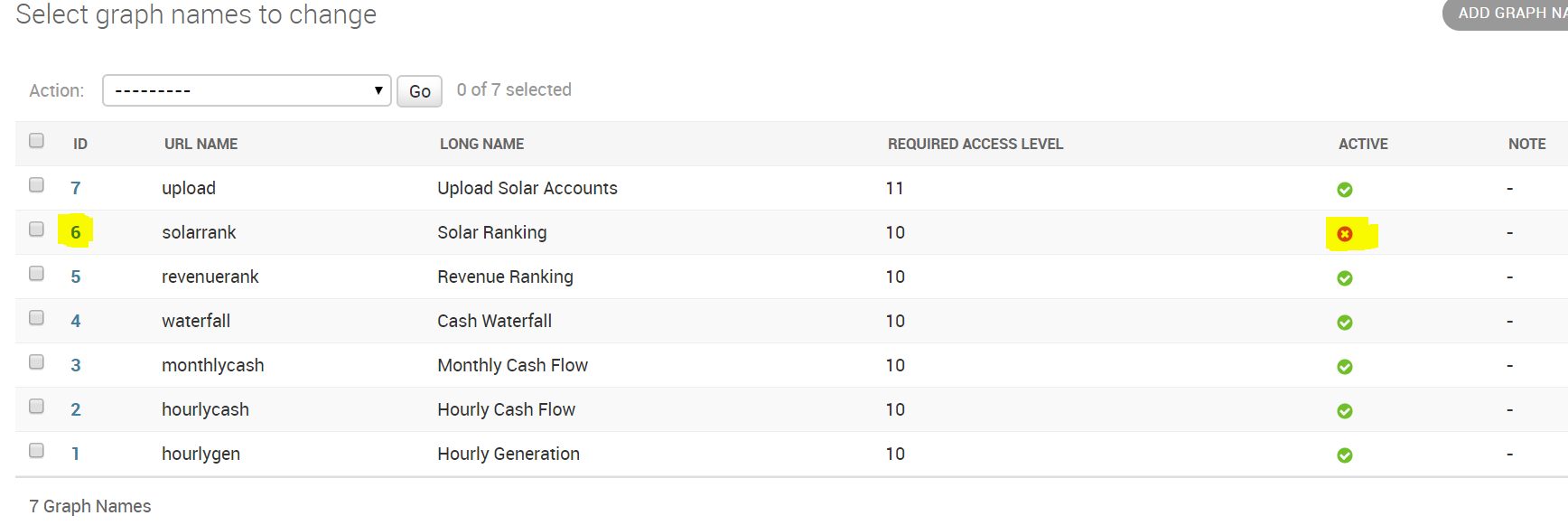




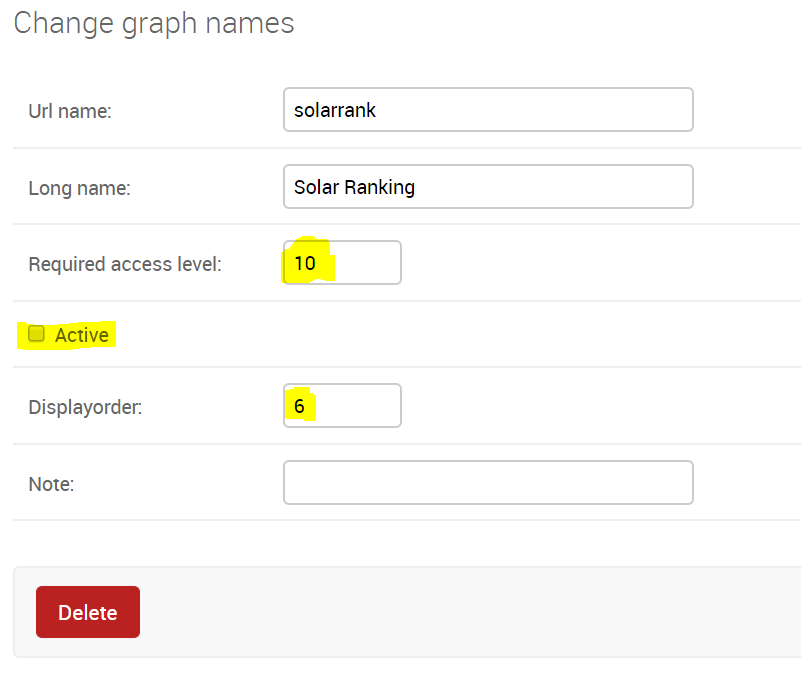
## Graph Required Access Level

Modify the graph access level to enable/disable it altogether, and to set the permission level

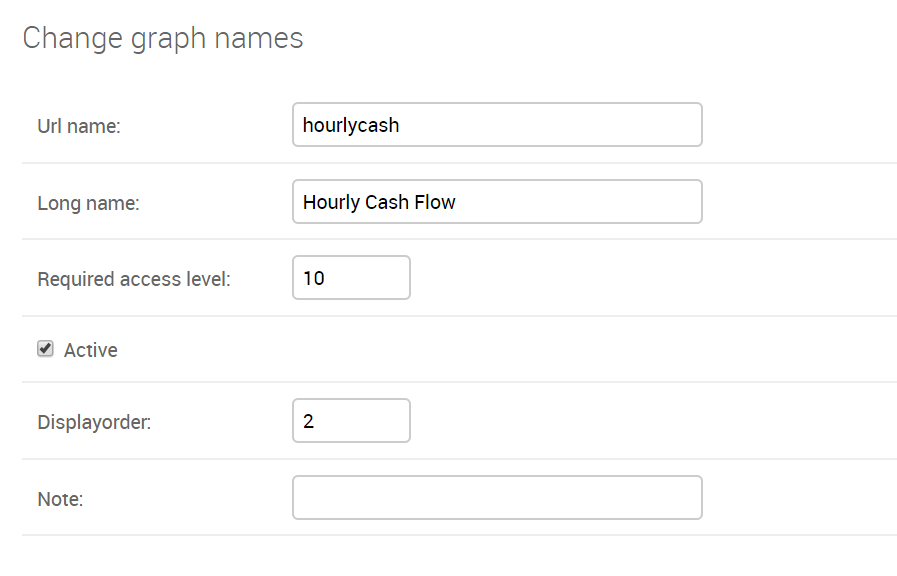




Notice the solarrank graph has an access level of 10, is NOT active (so the hyperlink won’t show up for users even if they have sufficient privileges) and the display order is 6. Please note, despite being inactive, the user can still see it by navigating directly to the hyperlink. The display order affects the order of the hyperlinks on the home screen. Don’t delete these records.

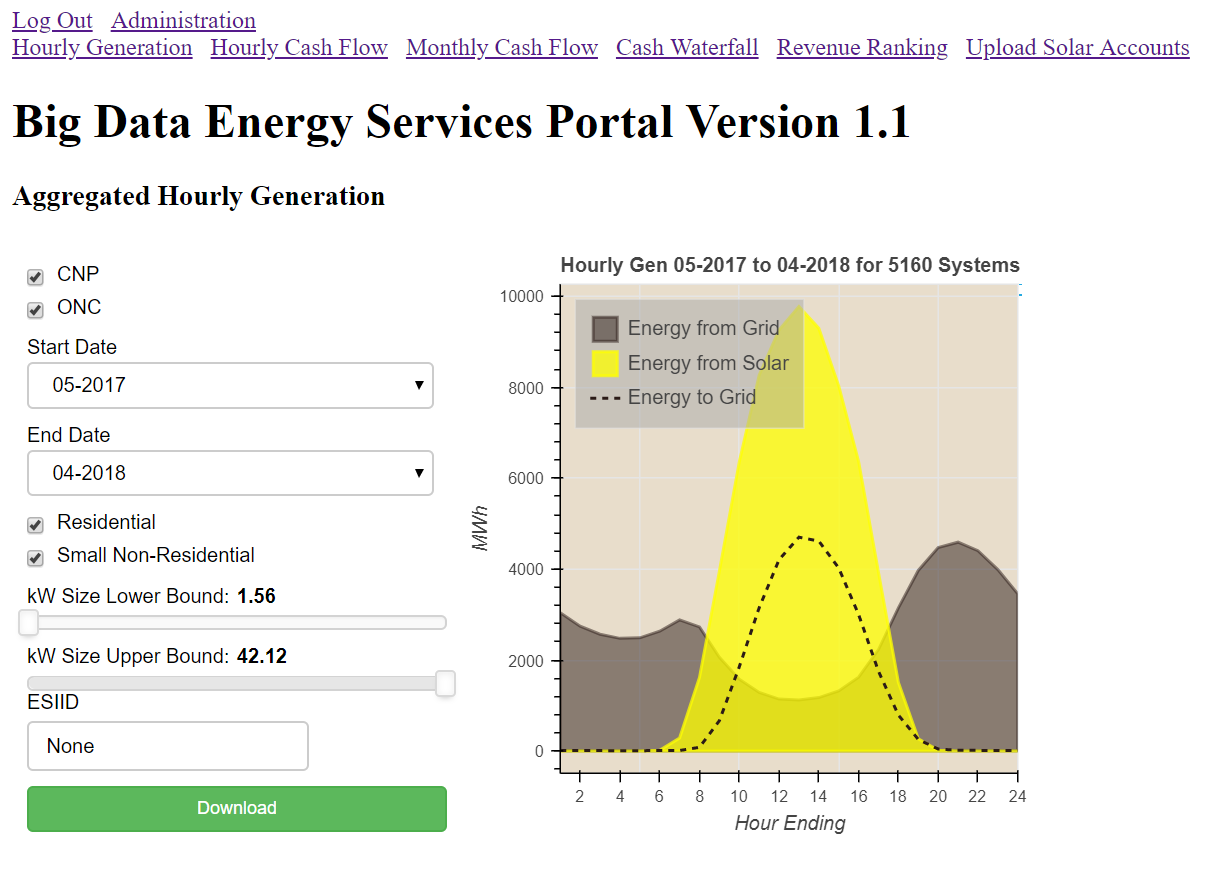


For the hourlycash graph, any user with an access level of at least 10 will be able to see it since it’s active.



# Application Navigation

## Aggregated Hourly Generation



This chart displays an entire month worth of solar generation and consumption along with a line showing the net consumption.

The user has the option to filter based on:

1. TDSP
2. Flow Date Range
3. Premise Type
4. System Size
5. ESIID

The MP2Energy database has a table called CustomerUsage15 that contains most of the data we need. You can see the actual query in the queries.py file.

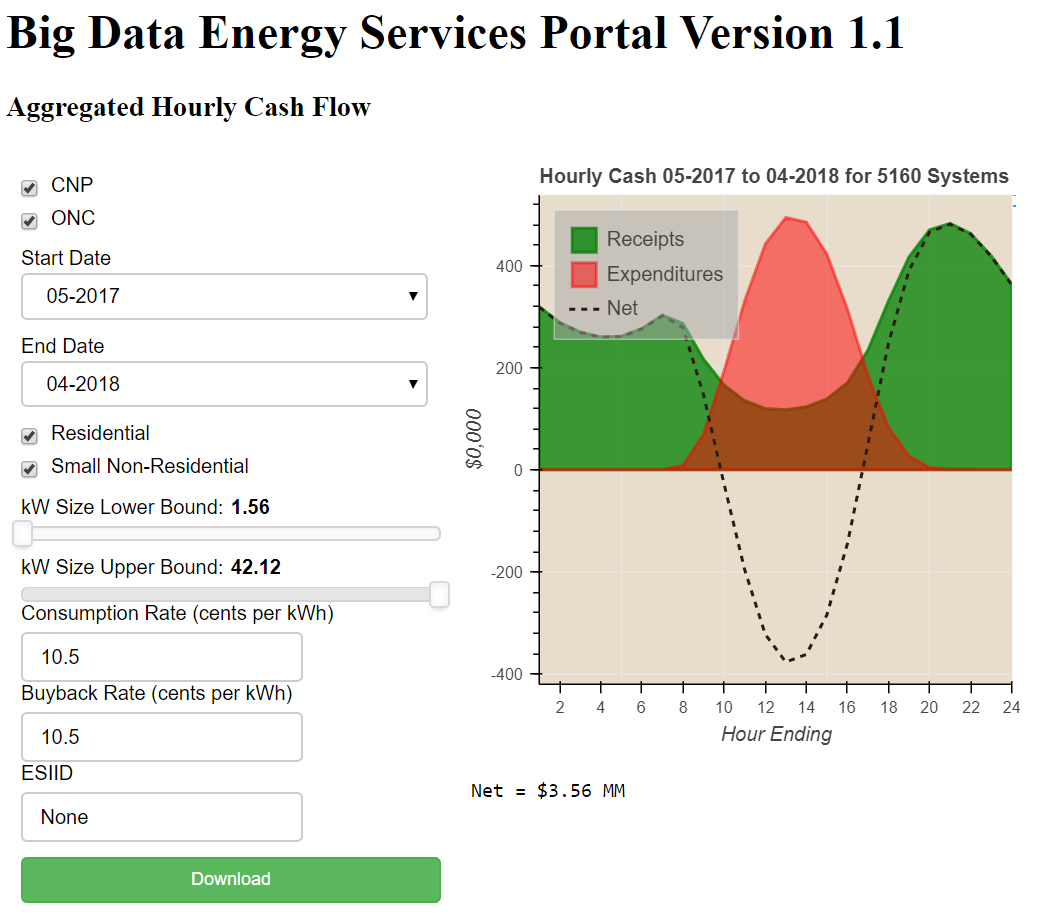
Columns in the download file:

1. Actual GHI – Measured by instrument. Actually landed at UT-Pan America site
2. Expected GHI – from NREL TMY3 dataset
3. Export kWh – Amount that flowed from the site back to the grid
4. Import kWh – Amount that flowed from the grid to the site
5. Solar Generation – kWh that the system likely generated, based on PVWatts model, assuming system located in Houston or Dallas (determined by TDSP) and rounded to the nearest 0.5kW for system size

The TDSP is calculated using this formula:

**def** get\_tdu(df):  
 *# https://www.startexpower.com/content/dam/startexpower/Misc%20Docs/Critical%20Care%20Form%20-%20ESP.pdf* df[**'TDSP'**] = **"unknown"** df.loc[df.Account\_No.str.startswith(**"1020404"**), **"TDSP"**] = **"AEPN"** df.loc[df.Account\_No.str.startswith(**"1003278"**), **"TDSP"**] = **"AEPC"** df.loc[df.Account\_No.str.startswith(**"1008901"**), **"TDSP"**] = **"CNP"** df.loc[df.Account\_No.str.startswith(**"1044372"**), **"TDSP"**] = **"ONC"** df.loc[df.Account\_No.str.startswith(**"1017699"**), **"TDSP"**] = **"ONCS"** df.loc[df.Account\_No.str.startswith(**"1017008"**), **"TDSP"**] = **"SHD"** df.loc[df.Account\_No.str.startswith(**"1013830"**), **"TDSP"**] = **"NUE"** df.loc[df.Account\_No.str.startswith(**"1040051"**), **"TDSP"**] = **"TNMP"  
 return** df

## Aggregated Hourly Cash Flow



This chart displays an entire month worth of cash flows related to receipts and expenditures, along with a line that nets the two figures. Receipts are calculated based on the consumption rate times the amount of kWh that flows to the site from the grid. Expenditures are calculated based on the buyback rate times the amount of kWh that flows from the site to the grid.

The user has the option to filter based on:

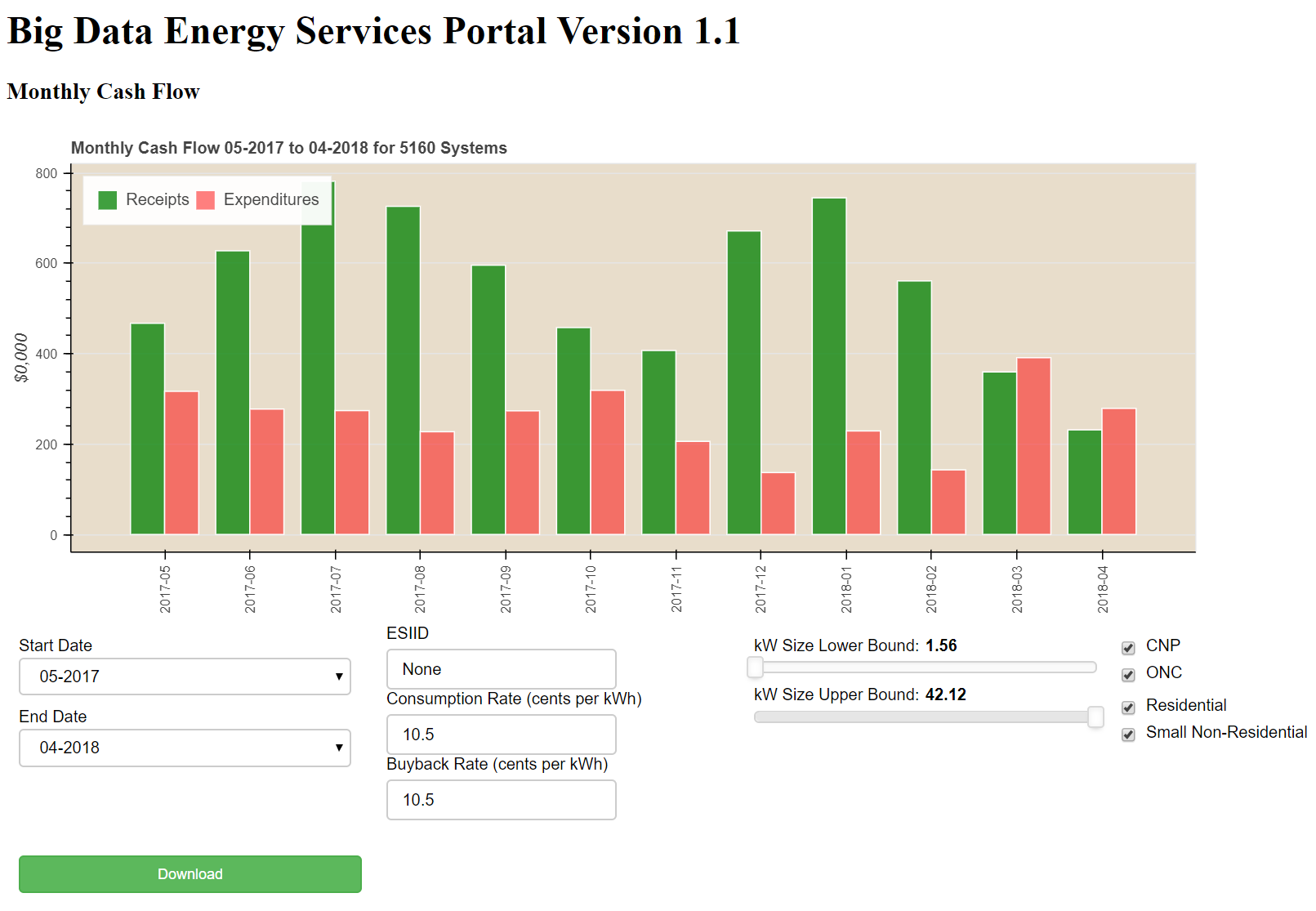
1. TDSP
2. Flow Date Range
3. Premise Type
4. System Size
5. ESIID

The user must define the consumption rate and buyback rate. If a non-numeric number is entered, the default (10.5 cents per kWh) will be used instead.

Columns in the download file in addition to the column in Hourly Generation:

1. Actual GHI – Measured by instrument. Actually landed at UT-Pan America site
2. Expected GHI – from NREL TMY3 dataset
3. Export kWh – Amount that flowed from the site back to the grid
4. Export Wholesale Dollars – Export kWh \* buyback rate in dollars
5. Import kWh – Amount that flowed from the grid to the site
6. Import Wholesale Dollars – Import kWh \* consumption rate in dollars
7. Solar Generation – kWh that the system likely generated, based on PVWatts model, assuming system located in Houston or Dallas (determined by TDSP) and rounded to the nearest 0.5kW for system size

## Monthly Cash Flow



The monthly cash flow is closely related to the hourly cash flow. However, rather than aggregating into a single cumulative day, the Monthly view looks at multiple months and groups payments and receipts by month.

The user has the option to filter based on:

1. Start Date
2. End Date
3. TDSP
4. Premise Type
5. System Size
6. ESIID

The user must define the consumption rate and buyback rate. If a non-numeric number is entered, the default (10.5 cents per kWh) will be used instead.

Columns in the download file:

1. Flow Date – First day of the flow month
2. Flow Month – Integer of flow month
3. Flow Year – Integer of flow year
4. Import kWh – Amount that flowed from the grid to the site
5. Export kWh – Amount that flowed from the site back to the grid
6. Expected GHI – from NREL TMY3 dataset
7. Actual GHI – Measured by instrument. Actually landed at UT-Pan America site
8. Solar Generation – kWh that the system likely generated, based on PVWatts model, assuming system located in Houston or Dallas (determined by TDSP) and rounded to the nearest 0.5kW for system size
9. Export Wholesale Dollars – real time power rate \* Export kWh in dollars
10. Import Wholesale Dollars – real time power rate \* Import kWh in dollars
11. Receipts – Export kWh \* consumption rate in dollars
12. Expenditures – Import kWh \* buyback rate in dollars
13. FlowDateString – YYYY-MM

## Cash Waterfall



The waterfall view starts from revenue and goes down to gross margin for a given month, factoring in wholesale energy costs, TDSP charges, solar payments (buyback payments), and wholesale sellback.

The user has the option to filter based on:

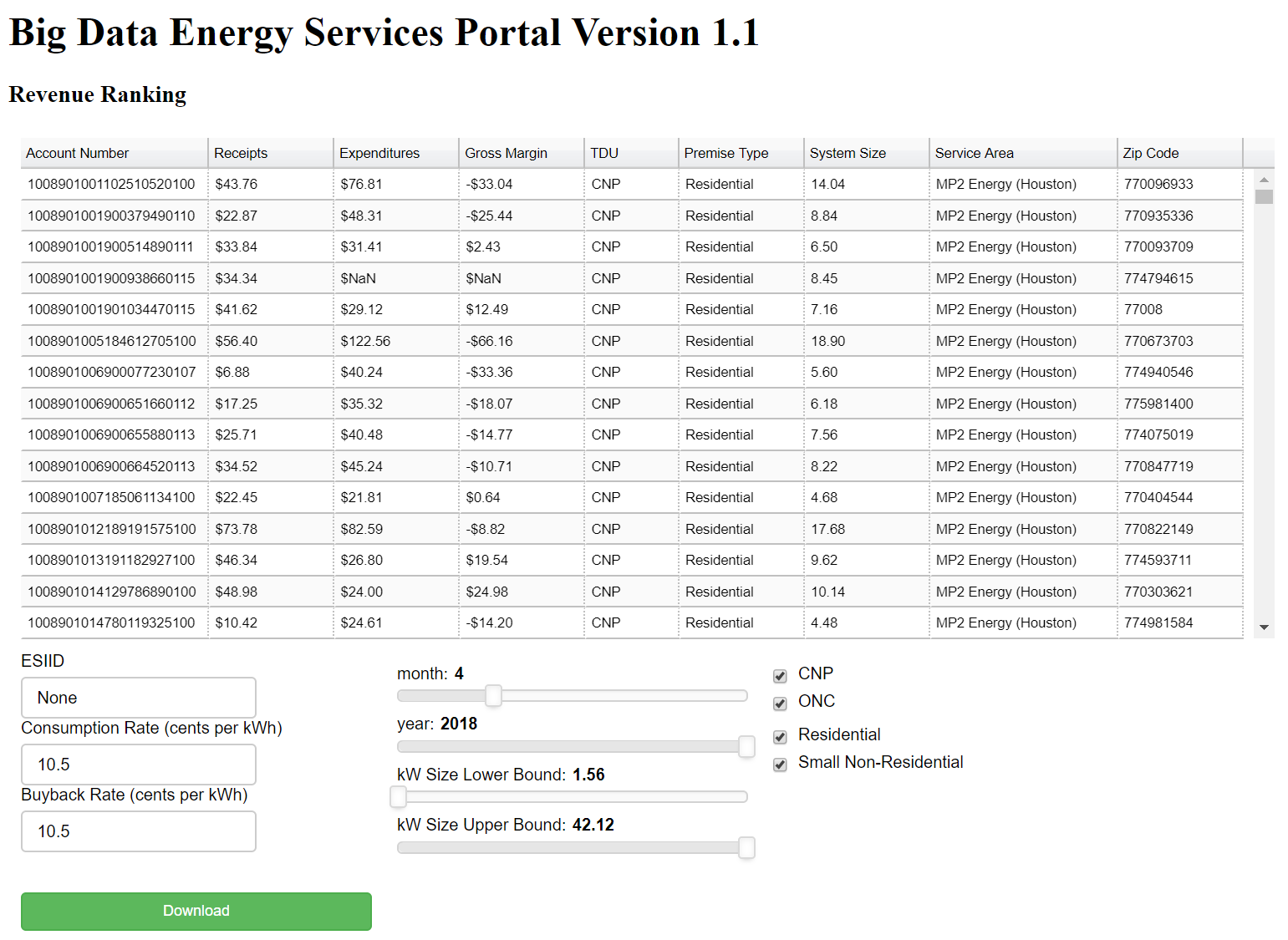
1. TDSP
2. Flow Date Range
3. Premise Type
4. System Size
5. ESIID

The user must define the consumption rate and buyback rate. If a non-numeric number is entered, the default (10.5 cents per kWh) will be used instead.

Wholesale energy costs are based on the kWh the flows from the grid to the site times the prevailing real time market settlement price. TDSP charges are based on the kWh that flows from the grid to the site time the user input. If a non-numeric number is entered, the default (5 cents per kWh) will be used instead. Solar payments are the kWh that flows from the site to the grid times the buyback rate. Wholesale sellback is the kWh that flows from the site to the grid times the prevailing real time market settlement price.

All fields in the download file have been defined above, in other visualizations.

## Revenue Ranking



The revenue ranking chart allows users to sort all solar customers based on any field in the header. This allows users to isolate high value and low value customers.

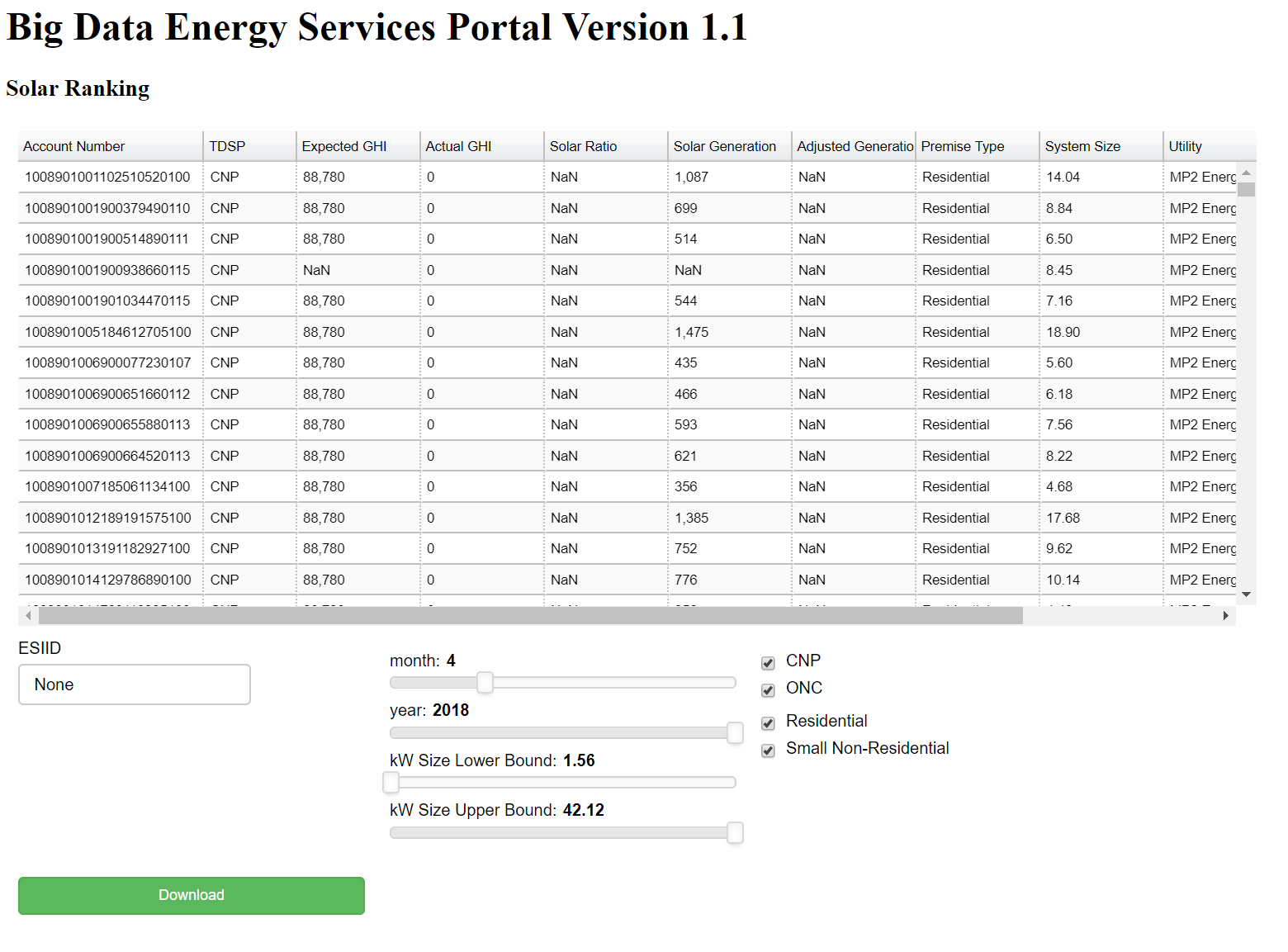
The user must define the consumption rate and buyback rate. If a non-numeric number is entered, the default (10.5 cents per kWh) will be used instead.

The user has the option to filter based on:

1. TDSP
2. Flow Month and Flow Year
3. Premise Type
4. System Size
5. ESIID

The data used in this table follows all assumptions listed in prior sections of this document.

## Solar Ranking



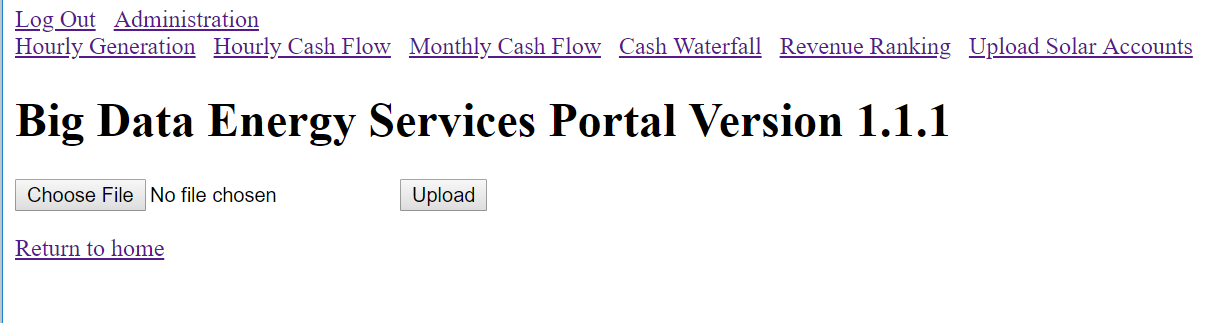
The revenue ranking chart allows users to sort all solar customers based on any field in the header. This allows users to isolate high and low output systems.

The user has the option to filter based on:

1. TDSP
2. Flow Month and Flow Year
3. Premise Type
4. System Size
5. ESIID

The data used in this table follows all assumptions listed in prior sections of this document.

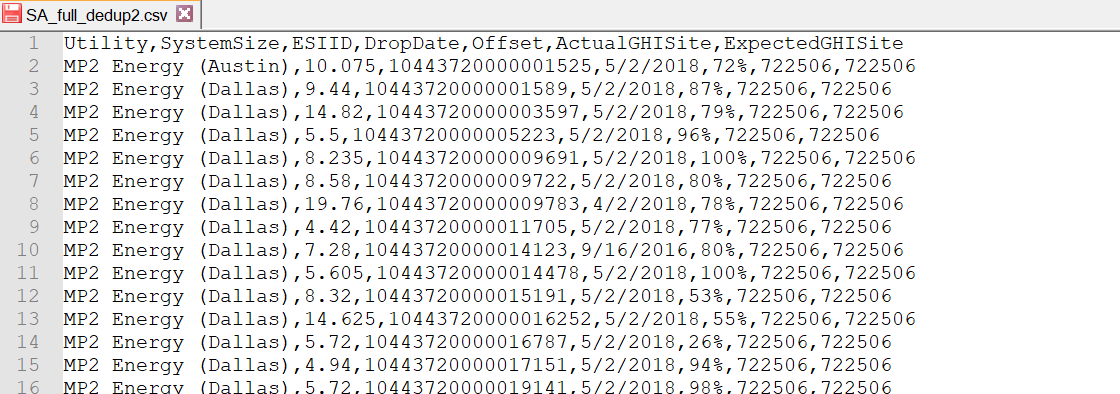
## Upload Solar Accounts

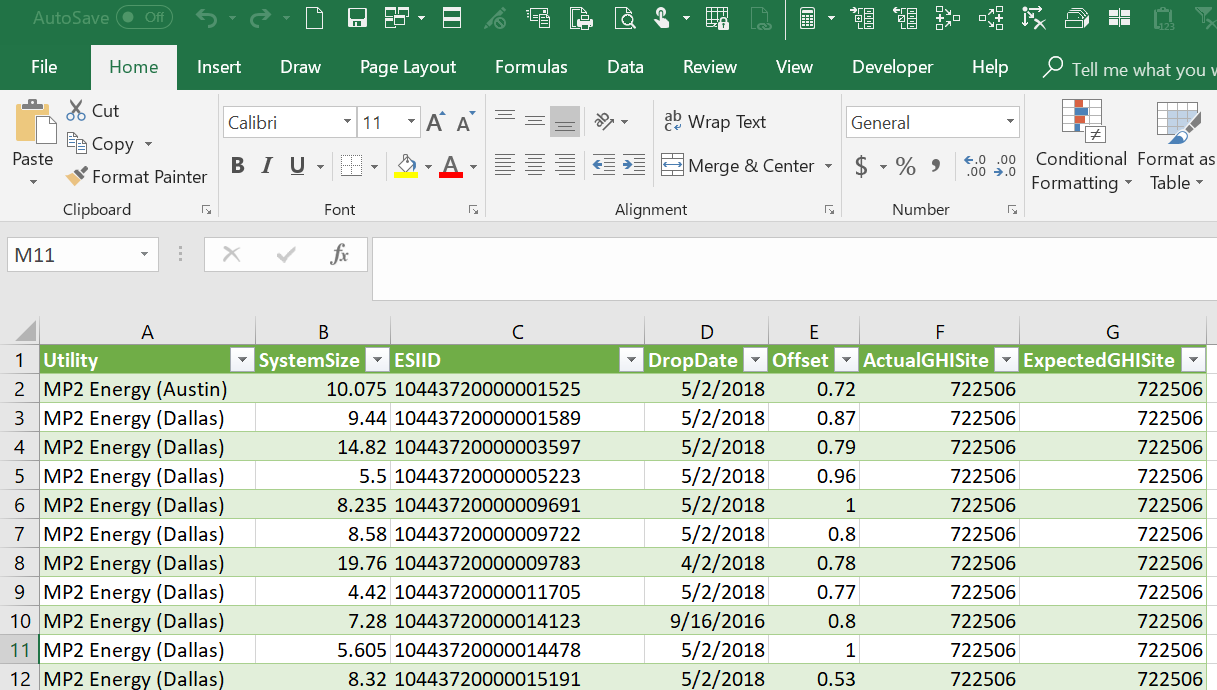


The user can upload a CSV file that contains solar accounts. This action will flush and fill. After doing so the new accounts will be active on the analytics portal the following day since database updates occur overnight.

The format of the CSV file is critical, and there is some error trapping to ensure it’s correct. The error trapping should not be considered comprehensive.

The format of the file (shown in a text editor and Excel)





It’s critical that the ESIID column doesn’t get truncated to scientific notation if you use MS Excel. Please reference MS Excel documentation to import properly to avoid the translation. It’s also critical that the ActualGHISite and ExpectedGHISite contain the values 722506. These can’t be blank.

## SSL Authentication

The site is secured with SSL Authentication. The certificate must be renewed from time to time.

# Data Flow

1. An external BDES process loads SMT data into the CustomerUsage15 table for prior day
2. An external BDES process loads actual irradiance into the ActualIrradiance table for prior day
3. Once both of those complete successfully, a daily cron job automates the rest
   1. Create SolarGeneration table
      1. Solely a query-driven process
      2. Takes 10-20 minutes.
      3. Could possibly be optimized if necessary, although speed won’t be an issue
   2. Create Hourly table
      1. SolarGeneration task must complete before launching this process
      2. Complex merges – can’t easily be done via SQL
      3. Memory Intensive Python program (probably 4-8 GB of RAM)
      4. Not CPU intensive
      5. Takes about 10-20 minutes
      6. Optimizing further is possible but will require a non-trivial amount of development time for some parts. Other parts (queries) may be easier to optimize

# Notes

All code is saved on GitHub, including setup files for nginx, bokeh server, gunicorn, SSH, and sudoers.

There is a file named Server\_Instruction.txt that shows how to instantiate this application to a new server.

User login sessions expire after five minutes of inactivity with an expiration warning happening at four minutes. These are adjustable via the settings.py file.

## To Do

1. Implement an emergency recovery process should the machine crash, including transfer to code to BDES
2. Change Data flow such that SolarGeneration is a stored procedure rather than run on the virtual machine. Also debug the query – may be dropping NULLs
3. Migrate credentials and database (SSL -encrypted database and application credentials instead of Jonathan’s credentials)

## Possible Add-Ons

1. Branding / CSS Styling along with Favicon
2. Better Hover options
3. Timestamp on data files displayed so the user knows the timestamp of the last available data
4. Allow users to update their own passwords
5. Keep track of user statistics -- logins, session times, tracking pages
6. Better monitoring (for cron job data load, errors, etc)