API Doc

How to use:

1. For each JSON given and received, it will show the JSON in it’s “compact” format. Please copy and past the JSON into the left window of this link: <http://www.jsoneditoronline.org/>
2. Make sure that the arrays used for both the backend and frontend match EXACTLY as they are written in this document
3. Each API endpoint will call to a url. Use this exact url when creating it on the backend as that is where it will be called in the front end.

Matching Arrays:

1. Directions: [“n”, “ne”, “e”, “se”, “s”, “sw”, “w”, “nw”, “flat”]
2. Panel Types: [“Monocrystalline”, “Polycrystalline”, “Amorphous Thin Film”]

Notes:

1. Panel Size is measured in kWh
2. All measurements given and received are in METERS
3. Differences in panel types can be found here: <http://www.pvsolarchina.com/difference-between-monocrystalline-polycrystalline-and-amorphous-thin-film-solar-cell.html>

/////////////////////////////////////////////////

API1: Lat Lon Api

Front End Call:

submitLocation (GET)

URL: /submitLocation?lat=-27.469516&lon=153.02323

http://jotm-solar-calc.appspot.com/submitLocation

Backend Process:

1. take the lat lon from the call and save into variable
2. check that the lat lon combination isn’t already in the database, and if it is, then send back a valid JSON else continue through this list
3. take the lat, lon and call the following apis based on the first of each month of the previous year (12 + 12: so that’s a total of 24 calls) [http://api.wunderground.com/api/3a152325c1969bdf/history\_20110601/astronomy/q/-27.469516,153.02323.json](http://api.wunderground.com/api/3a152325c1969bdf/history_20120601/astronomy/q/-27.469516,153.02323.json) [http://api.wunderground.com/api/3a152325c1969bdf/history\_20110601/q/-27.469516,153.02323.json](http://api.wunderground.com/api/3a152325c1969bdf/history_20120601/q/-27.469516,153.02323.json)
4. For each result record the total amount of sunlight in hours (sun set - sun rise), and the min and max temperatures of each day and put it all into the database along with the lat and lon as the unique id (probably will still have to have a key field, but will use lat and lon to determine whether or not that location has been looked up previously as in step 2).
5. Do a check that the lat lon combination resides in the database, and if it does, send back a JSON as outlined in “expected result”

Expected Result:

{"status":"valid"}

/////////////////////////////////////////////////

API2: Results Api

Front End Call:

getResults (GET)

URL: /getResults?info=\”{JSON}\”

http://jotm-solar-calc.appspot.com/solar\_calculation?jsonParameter=

Given JSON:

{"house":{"currency":"AUD","country":"Australia","lat":"123.123","lon":"-123.123","wire":12.1,"electricityCompany":"Origin","tariff":0.023,"usagePerQuarter":[1234,1234,1234,1234],"cost":0.029,"roofSection":[{"sectionId":0,"sectionName":"Above the Porch","angle":7.2,"direction":2,"width":2.2,"length":2.2,"numberOfCurrentPanels":1,"ageOfCurrentPanels":0,"typeOfCurrentPanels":0,"sizeOfCurrentPanels":4.9},{"sectionId":1,"sectionName":"Beside the Kitchen","angle":27.3,"direction":4,"width":5,"length":5,"numberOfCurrentPanels":0,"ageOfCurrentPanels":0,"typeOfCurrentPanels":0,"sizeOfCurrentPanels":0},{"sectionId":2,"sectionName":"Above the Garage","angle":0,"direction":13,"width":4,"length":6,"numberOfCurrentPanels":0,"ageOfCurrentPanels":0,"typeOfCurrentPanels":0,"sizeOfCurrentPanels":0}]},"scenarios":[{"name":"budget","typeOfNewPanels":1,"sizeOfNewPanels":75,"numberOfPanelsPerSection":[2,3,0],"quotedPrice":0},{"name":"crazy","typeOfNewPanels":2,"sizeOfNewPanels":165,"numberOfPanelsPerSection":[5,8,0],"quotedPrice":0}]}

Backend Process:

1. Take the JSON and put all values to be used for calculations into variables
2. Look up Lat and Lon in the database and take the daylight and temperature data (note: if the Lat Lon is not in the database for whatever reason, run the submit location code again)
3. Calculate the results for each scenario and give back the expected JSON

Expected Result:

{"result":[{"name":"budget","quotedPrice":0,"quarterlyResults":[{"powerGenerated":1234,"powerUsed":1234,"powerBought":1234,"excessPowerGenerated":1234},{"powerGenerated":1234,"powerUsed":1234,"powerBought":1234,"excessPowerGenerated":1234},{"powerGenerated":1234,"powerUsed":1234,"powerBought":1234,"excessPowerGenerated":1234},{"powerGenerated":1234,"powerUsed":1234,"powerBought":1234,"excessPowerGenerated":1234}],"powerGeneratedPerYear":1234,"powerUsedPerYear":1234,"powerBoughtPerYear":1234,"excessPowerGeneratedPerYear":1234,"breakEvenTime":{"years":2,"months":3},"moneySavedAfter1Year":1234,"moneySavedAfter5Years":1234,"graphURL":"http://www.graph.com"},{"name":"crazy","quotedPrice":0,"quarterlyResults":[{"powerGenerated":1234,"powerUsed":1234,"powerBought":1234,"excessPowerGenerated":1234},{"powerGenerated":1234,"powerUsed":1234,"powerBought":1234,"excessPowerGenerated":1234},{"powerGenerated":1234,"powerUsed":1234,"powerBought":1234,"excessPowerGenerated":1234},{"powerGenerated":1234,"powerUsed":1234,"powerBought":1234,"excessPowerGenerated":1234}],"powerGeneratedPerYear":1234,"powerUsedPerYear":1234,"powerBoughtPerYear":1234,"excessPowerGeneratedPerYear":1234,"breakEvenTime":{"years":2,"months":3},"moneySavedAfter1Year":1234,"moneySavedAfter5Years":1234,"graphURL":"http://www.graph.com"}],"status":"valid"}