WS API troubleshooting / FAQ

How should I set value of site id attribute?

Value of attribute id is only for informative purposes (doesn't affect results of WS API calls) and is mostly used for API usage monitoring reports etc.
Therefore as such it's entirely up to you what value you will choose for attribute id, however we recommend that you try to follow these best practices:

- try to always use the same site id if you are requesting the same location coordinates - this helps to easily visually identify what site / power-plant was concerned within each request
- also try to use different site id for sites with different locations (i.e. do not assign the same id for two different sites)

Other than: value of site id must follow common practice for variable names in programming languages:

- can not start only with letter or underscore
- can only contain characters a..z, A..Z, 0..9, _, .

Can I set minutes precision of timezone i.e. "UTC+07:30" (instead of only whole hours precision)?

Unfortunately this is not possible, WS API currently supports only timezones that are integer multiples of hourse, i.e.: "UTC+07:00" or "UTC+08:00" and timezones like "UTC+07:30" are not supported.

Why 1 day of data in MIN_15 summarization doesn't sum up to 1 day of data in HOURLY or DAILY summarization (for GHI, DNI, GTI, PVOUT...)?

This is because MIN_15, HOURLY and DAILY all use different units on the output, furthermore irradiation data also uses different units on the output than PVOUT.
You can always check the units used for each output column in the metadata section:

For irradiation variables (GHI, DNI, GTI, DIF...) we have:

summarization	unit
MIN_*	W/m2 (Power / m2)
HOURLY	Wh/m2 (Energy / m2)
DAILY	kWh/m2 (Energy / m2)

For PVOUT variable we have:

summarization	unit
MIN_*	kW
HOURLY	kWh
DAILY	kWh

Therefore when summing up values in one day and comparing against different summarizations we need to make sure that they are all in matching units.
For example conversion of GHI MIN_15 to GHI HOURLY should be like this:

SUM(GHI, HOURLY) = SUM(GHI, MIN_15) / 4.0

(Here dividing by 4.0 is consequence of converting average power over four 15 minute intervals into how much energy that power will generate over duration of one hour)

Unable to get WS API working

Quick checklist (things you can quickly try first)

- Make sure you are properly setting content type and using HTTP POST method and setting up correct request body as XML while doing the request (i.e. not calling WS API as url directly from browser)
- Is the url of the API call correct? (e.g.: is it possible that you are accidentally using DataDelivery endpoint instead of PvPlanner or vice versa?)
- Is the provided key correct? (e.g.: in case you have multiple accounts, is the provided key correct for intended call? Didn't you accidentally use key from different account that was intended for different purpose?)
- Try to use the "proven to be working" / simplest / minimum XML requests adapted for you account. You can find minimum and working requests in our technical documentation here:
  - For DataDelivery WS: request / response examples service (you may need to adjust lat/lng, dateFrom, dateTo, processing keys and summarization based on the access permissions of your account)
  - For PvPlanner WS service (you may need to adjust lat/lng)
- If the server answers with response: have you checked the detailed error message? Most of the times the response contains detailed message with hint why the request failed.
  In case error happens the response is in JSON format instead of XML, so you may need to view the response in JSON or Raw view (if you are using third party applications for doing requests like SOAP UI),
  here are examples how the error message may look like:

```
{"message": "Error 403 Access is denied. dateFrom must be >= 2020-02-04"}
```

  Here (assuming that the request was made on 2020-02-04) the message is trying to tell us that we can not access data older then today - this could for example happen if your account has only access to forecast data and you are trying to access historic data. In such case you need to change the dateFrom of the failing request to accommodate this. In case you think you should have access (i.e. it is a configuration problem on our side and you should have access for historic data) please ask for technical support

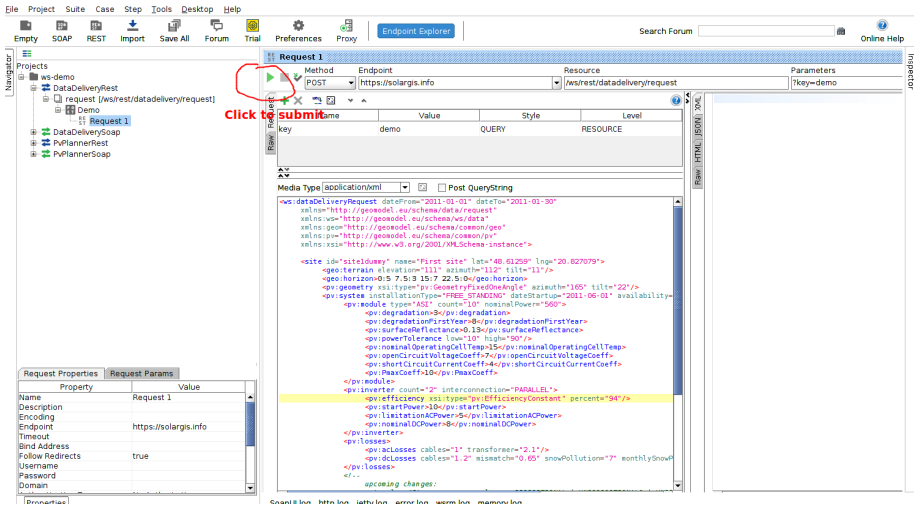
```
{"message":"Error 403 Access is denied. Processing key DNI is not allowed"}
```

  So here the error messages are trying to tell that your account doesn't have access to use processingKey = DNI, so again you need remove the offending key from processingKyes of the request. In case you think you should have access (i.e. it is a configuration problem on our side) please ask for technical support
  These were just examples but there are many more reasons why your requests may fail due to access restrictions (too many calls per day, summarization that is not allowed, requesting more than one month of data per single request and so on.)

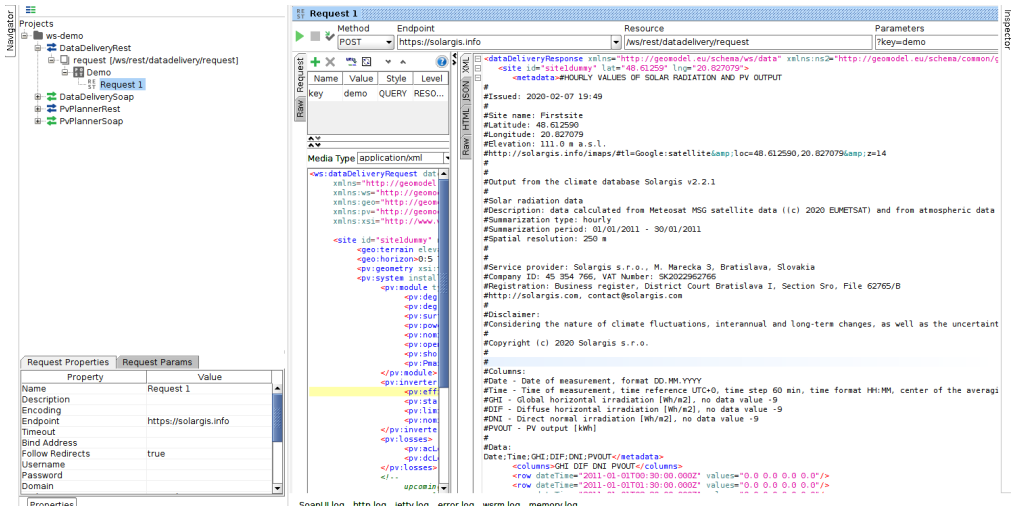
Request still not working / no idea what's wrong

First we need to establish that that you are able to call the request with demo key, in order to do so please follow these steps:

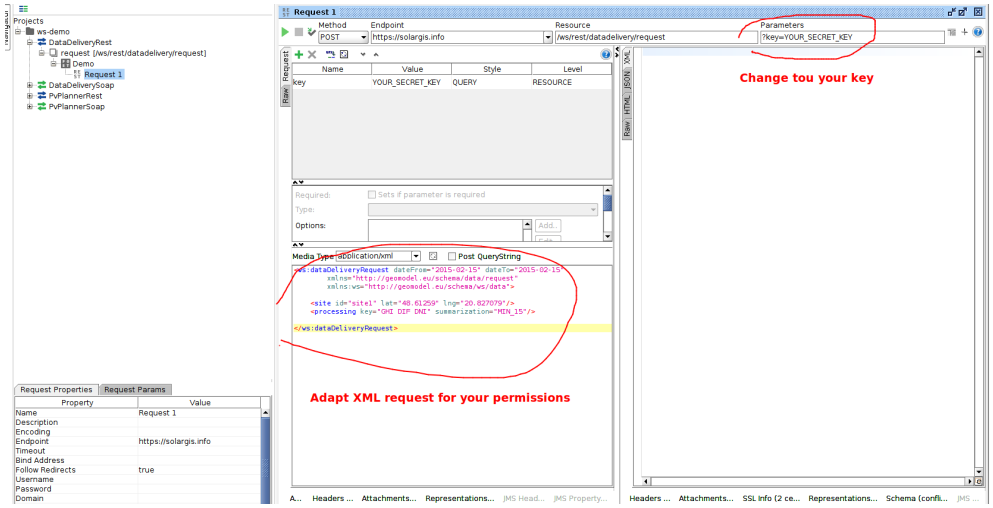
- Download SOAP UI (standalone free application for sending HTTP requests) from here: https://www.soapui.org/
- Install SOAP UI and download following demo project provided by Solargis: ws-demo-soapui-project.xml
- Run SOAP UI and import the ws-demo-soapui-project.xml (File Import Project)
- Depending on whether you have DataDelivery or PvPlanner license you should navigate to either DataDeliveryRest or PvPlannerRest Demo request
  from the imported project. Rest of this guide shows steps for DataDelivery license, however the steps for PvPlanner are very similar.
- Navigate to the DataDeliveryRest Demo request as seen on following image and submit the request with the "play" button:



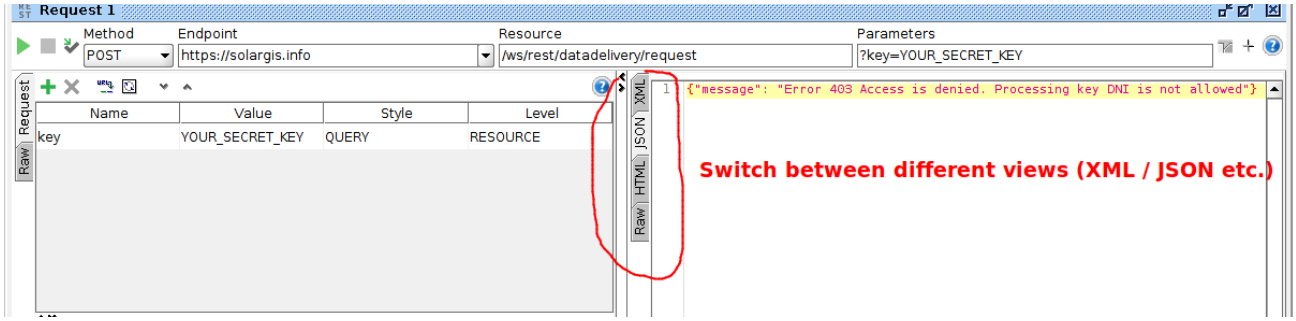
- After the request is finished, you should see HTTP status 200 (OK) and following valid response - this should work immediately out of the box (no need for you to change / configure anything):



- Now that you have established that the demo is working correctly you can start adapting the demo request for your license. In order to do so,
  you need to change key from "demo" to the one you were provided with your WS API account and also to adapt XML body of the request to the one that will work with your key,
  therefore based on your account permissions you may need to change latitude / longitude of the request, dateFrom / dateTo etc. We strongly recommend that you start
  with the simplest requests which are known to be working and build up from there. As already mentioned in "quick checklist" you can find working examples of simple
  DataDelivery or PvPlanner requests here : DataDelivery WS: request / response examples and here: PvPlanner WS



- After you have made the necessary updates you should be able to successfully submit the request and receive correct XML response. If you do not see the expected correct XML response, then you might need to change the view to JSON or RAW.
  Most of the times when the API call fails and you switch to JSON view you will see message like this which provides hint why the request has failed. E.g.: in the following case it suggests that you should remove "DNI" from the list of the processing keys.



- Once you get the simple XML requests working with your secret key then you can attempt to add more complex stuff (i.e. detailed configuration of your PV power plant etc.) We recommend that you proceed in incremental steps and after each change you validate that the request is still working. That way if something breaks you know which change has caused it and you can work around it.
- Afterwards: if some request is not working in your implementation, please first make sure that the same request is also not working in SOAP UI, this is to eliminate potential bugs in your implementation. If the request is not working in your implementation but it is working in SOAP UI then you will need to review your implementation. As it's difficult for us to know what are the problems in your implementation we generally can't provide support regarding your implementation. But in order to solve the issues we recommend to compare that everything in your implementation is 1:1 to the working SOAP UI request, mainly that:
  - the XML content of the request in your implementation is identical to the SOAP UI
  - the url in your implementation matches the url in SOAP UI, including the correct key as the url parameter
  - you are using HTTP header Content-Type set to "application/xml" in your implementation
  - you are using POST method to do the HTTP call in your implementation
  - you are also setting the request body (i.e. parameter of the POST method) to the intended XML request string in your implementation
- If after following these steps you are convinced that some request that is supposed to be working is failing then please ask for our technical support