

InSight: Mars Weather Service API

(Last Updated: 8/28/2019)

NASA's InSight Mars lander takes continuous weather measurements (temperature, wind, pressure) on the surface of Mars at Elysium Planitia, a flat, smooth plain near Mars' equator. Please note that there are sometimes problems with the sensors on Mars that result in missing data! If you see a long gap, a search result may bring up more information on whether it is a long-lasting problem. Summaries of these data are available at <https://mars.nasa.gov/insight/weather/>.

This API provides per-Sol summary data for each of the last seven available Sols (Martian Days). As more data from a particular Sol are downlinked from the spacecraft (sometimes several days later), these values are recalculated, and consequently may change as more data are received on Earth. Additionally, please note that wind and other sensor data may not exist for certain date ranges. You can check out <https://mars.nasa.gov/insight/weather/> and scroll down to the 'seasonal weather report' you'll see the gaps where no data exists for some sensors.

This API is maintained and provided by NASA Jet Propulsion Laboratory and Cornell University. If you find bugs in this API, please use the contact form found at <https://mars.nasa.gov/feedback/>. The rate limit for this API is every hour no more than 2000 hits for each individual IP.

Defining the Data

The summary data are provided as an object in a JSON stream, a formal definition of JSON is [RFC 7159](#). Appendix A contains an abridged, typical JSON stream for this API as an example.

API URL

https://api.nasa.gov/insight_weather/?api_key=DEMO_KEY&feedtype=json&ver=1.0

Quick start

Object keys are indicated by [blue](#) text

If we assume the entire JSON object that is returned by the API is in a variable named **JSO**

Get a top-level Sol key from the top-level key [sol_keys](#), put it in a variable named **sol**

- Javascript: **JSO**.[sol_keys](#)[0] through **JSO**.[sol_keys](#)[6]
- Some other languages (e.g. Python), typical:
 - **JSO**[["sol_keys"](#)][0] through **JSO**[["sol_keys"](#)][6]
 - assumes seven Sols are present)

Per-Sol average values, linear sensors

- **JSO**[**sol**].[AT.av](#) - atmospheric temperature, degrees Celsius
- **JSO**[**sol**].[PRE.av](#) - atmospheric pressure, Pascals
- **JSO**[**sol**].[HWS.av](#) - horizontal wind speed, metres per second

Per-sol 16-wind compass point of most common wind direction e.g. N for North or ESE for East-SouthEast

- **JSO**[**sol**].[WD.most_common.compass_point](#)

Top-level keys

Key	Description
<SOL>	Object; key is a string of one or more decimal digits
E.g. "259"	Object contains summary data for Sol <SOL> e.g. "259": {...} => summary data for Sol 259
sol_keys	Array of strings e.g. ["259","260","...", "265"] Sols, as strings, with enough data in at least one channel (temperature, pressure or wind) to meet validity criteria. This array indicates which top-level <SOL> keys, described above, are present.
validity_checks	Object. Provenance of evaluations whether enough data are present to adequately represent each Sol. The criterion is at least one sample present in 18, out of 24, hours in a Sol, for each sample. N.B. an "hour" refers to a martian hour i.e. one-twenty-fourth of a Sol

Per-Sol weather data object (key = <SOL>)

- N.B. a Sol will be present at the top level, and in the sol_keys object, if there is at least one sensor with data for that Sol that meet the validity criterion
- N.B. only data from sensors that meet the validity criterion will be present under each Sol
 - Data from all sensors (AT; HWS; PRE; WD) may not be present under each Sol at the top level
 - Data from at least one sensor will be present under each Sol at the top level

Key	Description
AT	Object; per-Sol atmospheric temperature sensor summary data
HWS	Object; per-Sol horizontal wind speed sensor summary data
PRE	Object; per-Sol atmospheric pressure sensor summary data
WD	Object; per-Sol wind direction sensor summary data
Season	String; per-Sol season on Mars; one of ["winter", "spring", "summer", "fall"]
First_UTC	Time of first datum, of any sensor, for the Sol (UTC; YYYY-MM-DDTHH:MM:SSZ)
Last_UTC	Time of last datum, of any sensor, for the Sol (UTC; YYYY-MM-DDTHH:MM:SSZ)

Per-Sol, per-sensor data object (top-level key = [AT](#) or [HWS](#) or [PRE](#))

Key	Description
av	Average of samples over the Sol (°F for AT; m/s for HWS; Pa for PRE)
ct	Total number of recorded samples over the Sol
mn	Minimum data sample over the sol (same units as av)
mx	Maximum data sample over the sol (same units as av)

Per-Sol, per-sensor data object for wind direction (top-level key = [WD](#))

Key	Description
<compass_pt_no>	Object; key is a string of one or two decimal digits indicating the ordinal of a 16-wind compass rose from North clockwise, e.g. "1", "2", ..., "16" N.B. These data could be used to create a wind rose histogram e.g. see this website .
most_common	Object or null; usually duplicates whichever of the <compass_pt_no> objects has the highest count (ct) The most_common key will always be present in any Sol, but it may have the value null if the wind direction data for that Sol do not pass the validity checks.

Per-Sol wind direction compass point object (key [<compass_pt_no>](#) or [most_common](#) in [WD](#))

Key	Description
compass_degrees	Number; the compass direction of the center of the compass point; degrees <ul style="list-style-type: none"> N.B. the wind is blowing <i>from</i> this direction $\pm 11.25^\circ$
compass_point	String; the name of the compass point e.g. "N" for North, or "ESE" for East-SouthEast
compass_right	Number; the positive-right (positive-east), horizontal component of a unit vector indicating the direction of the compass point
compass_up	Number; the positive--up (positive-north), vertical component of a unit vector indicating the direction of the compass point
ct	Number; the number of samples for the Sol in the 11.25° around this compass point

Provenance of validity checks (top-level key "[validity_checks](#)"; see Caveats below)

Key	Description
<SOL>	Object; key is a string of one or more decimal digits There will typically be more Sol keys here than in the top-level <SOL> keys
sol_hours_required	Number; typically 18; number of hours with at least one sensor datum required for that Sol's sensor data to be considered valid
sols_checked	Array of strings e.g. ["258","260", "...", "265"] Sols, as strings, that were checked against the validity criterion. This array indicates which <SOL> keys are present in the validity_checks object.

Per-Sol validity checks (key [<SOL>](#) in top-level object "[validity_checks](#)"; see Caveats below)

Key	Description
AT	Object; per-Sol validity check for atmospheric temperature sensor
HWS	Object; per-Sol validity check for horizontal wind speed sensor
PRE	Object; per-Sol validity check for atmospheric pressure sensor
WD	Object; per-Sol validity check for wind direction sensor

Per-Sol, per-sensor validity check (key [AT](#) or [HWS](#) or [PRE](#) or [WD](#) in previous table; see Caveats below)

Key	Description
sol_hours_with_data	Array of numbers; number indicating which hours have at least one datum recorded for this Sol and sensor; values are 0 through 23.
valid	Boolean value; true if there are at least <sol_hours_required> hours (typically 18) with at least one datum recorded for this Sol and sensor.

Caveats

- Everything under [validity_checks](#) key is for debugging by API providers; these data will not be of interest to typical Mars Weather Data API consumers.

Appendix A

Sample JSON data, abridged, re-arranged, and annotated

N.B. all blue text delimited on the left by blue #s are annotations

```
{  
  "sol_keys": [ "259", "260", "261", "262", "263", "264", "265" ],  
  "259": {  
    "AT": { "av": -71.233, "ct": 326642, "mn": -101.024, "mx": -27.149 },  
    "HWS": { "av": 4.35, "ct": 154146, "mn": 0.156, "mx": 17.617 },  
    "PRE": { "av": 761.006, "ct": 163012, "mn": 742.1498, "mx": 780.3891 },  
    "WD": {  
      "most_common": { "compass_degrees": 202.5, "compass_point": "SSW", "compass_right": -0.382683432365,  
        "compass_up": -0.923879532511, "ct": 28551 },  
      "8": { "compass_degrees": 180.0, "compass_point": "S", "compass_right": 0.0,  
        "compass_up": -1.0, "ct": 17699 },  
      "9": { "compass_degrees": 202.5, "compass_point": "SSW", "compass_right": -0.382683432365,  
        "compass_up": -0.923879532511, "ct": 28551 },  
      "10": { "compass_degrees": 225.0, "compass_point": "SW", "compass_right": -0.707106781187,  
        "compass_up": -0.707106781187, "ct": 27124 }  
    },  
    "First_UTC": "2019-08-19T08:03:59Z", "Last_UTC": "2019-08-20T08:43:34Z", "Season": "winter"  
  },  
  "260": {  
    "AT": { "av": -75.95, "ct": 300789, "mn": -101.715, "mx": -28.634 },  
    "PRE": { "av": 762.462, "ct": 149206, "mn": 741.1254, "mx": 777.796 },  
    "WD": { "most_common": null }  
    "First_UTC": "2019-08-20T08:43:34Z", "Last_UTC": "2019-08-21T09:23:09Z", "Season": "winter"  
  },  
  }  
}
```

Start of JSON stream

Array of Sols with data

Start of summary data for Sol 259

Atmospheric temperature data for Sol 259

Horizontal wind speed data for Sol 259

Pressure data for Sol 259

Wind direction summary for Sol 259

N.B. only a subset of the WD keys are shown in this sample

Wind direction data for most common compass point

N.B. count (key "ct") is 28551, which is the number of

WD readings in this compass point

Wind direction data for compass point 8=South; count is less

than that for most common point; points 1-7 and 11-16 are

excluded in this example to save space, but the counts could

be used to display a wind rose histogram c.f. [this website](#).

Wind direction data for compass point 9=SSW

N.B. count (key "ct") is 28551, which matches that of the

most common key above i.e. this is the same point

Miscellaneous provance: UTC range; season.

Start of summary data for Sol 260

N.B. Sol 260 has no valid data for HWS, and WD has null

for most_common; see validity_checks below for why

```
"261": {...},
"262": {...},
"263": {...},
"264": {...},
"265": {...},
```

```
### Summary data for Sols 261 and later are excluded
### in this example
```

```
"validity_checks": {
```

```
### Start of validity check data
```

```
  "sol_hours_required": 18,
```

```
### Input parameter to validity check algorithm: sensor
### data need to be present in at least 18 of 24 (martian)
### hours of a Sol for summary data to be present for that
### sensor in that Sol above
```

```
  "sols_checked": ["258", "259", "260", "261", "262", "263", "264", "265"]
```

```
### N.B. to get seven Sols of data, eight days are
### checked, because the most recent Sol is
### usually incomplete. As it turns out in this
### example, there are data available for at least
### one sensor for all Sols from 265 back to 259,
### so summary data for Sol 258 are not included
### above.
```

```
"258": {
```

```
### Validity check result data for Sol 258 are present,
### even though its summary data are not used
```

```
  "AT": { "sol_hours_with_data": [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23],
    "valid": true },
```

```
  "HWS": { "sol_hours_with_data": [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23],
    "valid": true },
```

```
  "PRE": { "sol_hours_with_data": [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23],
    "valid": true },
```

```
  "WD": { "sol_hours_with_data": [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23],
    "valid": true }
},
```

```
"259": {
```

```
### Start of validity check result data for Sol 259; data are
### present for all 24 hours for all sensors, so all sensor
### keys here (AT, HWS, PRE, WD) have valid = true
```

```
  "AT": { "sol_hours_with_data": [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23],
    "valid": true },
```

```
  "HWS": { "sol_hours_with_data": [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23],
    "valid": true },
```

```
  "PRE": { "sol_hours_with_data": [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23],
    "valid": true },
```

```
  "WD": { "sol_hours_with_data": [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23],
    "valid": true }
},
```

```
"260": {
```

```
### Start of validity check result data for Sol 259
```

```
"AT": { "sol_hours_with_data": [0,1,2,3,4,5,6,7,8,9,10,11,12,15,16,17,18,19,20,21,22,23],
        "valid": true },
```

```
### AT data are present in 22 of 24 hours, which is
### greater than 18 (sol_hours_required) above, so
### valid = true, and AT summary data are present for
### Sol 260 above
```

```
"HWS": { "sol_hours_with_data": [ 0,1,2,3,4,5,6,7,15,16,17,18,19,20,21,22,23],
        "valid": false },
```

```
### HWS data are present in 17 of 24 hours, which is
### less than 18 (sol_hours_required) above, so
### valid = false, and HWS summary data are not present
### for Sol 260 above
```

```
"PRE": { "sol_hours_with_data": [ 0,1,2,3,4,5,6,7,8,9,10,11,12,15,16,17,18,19,20,21,22,23],
        "valid": true },
```

```
### PRE data are present in 22 of 24 hours,
### so valid = true
```

```
"WD": { "sol_hours_with_data": [ 0,1,2,3,4,5,6,7,15,16,17,18,19,20,21,22,23],
        "valid": false }
```

```
### WE data are present in 17 of 24 hours,
### so valid = false
```

```
},
```

```
"261": {...},
```

```
### Validity check result data for Sols 261 and later are
### excluded for this example
```

```
"262": {...},
```

```
"263": {...},
```

```
"264": {...},
```

```
"265": {...}
```

```
}
```

```
### End of validity_check result data
```

```
}
```

```
### End of JSON stream
```