Network Working Group	C. Jennings
Internet-Draft	Cisco
Intended status: Experimental Protocol	March 14, 2011
Expires: September 15, 2011	

Media Type for Sensor Markup Language (SENML)
draft-jennings-senml-05

Abstract

This specification defines media types for representing simple sensor measurements in JSON. A simple sensor, such as a temperature sensor, could use this media type in protocols such as HTTP to transport the values of a sensor.
The IETF has been notified of intellectual property rights claimed in regard to some or all of the specification contained in this document. For more information consult the online list of claimed rights.

Status of this Memo

Copyright Notice

Table of Contents

1. Overview

Connecting sensors to the internet is not new, and there have been many protocols designed to facilitate it. This specification defines new media types for carrying simple sensor information in a protocol such as HTTP or CoAP[I-D.ietf-core-coap]. This format was designed so that processors with very limited capabilities could easily encode a sensor reading into the media type, while at the same time a server parsing the data could relatively efficiently collect a large number of sensor readings. There are many types of more complex measurements and readings that this media type would not be suitable for. A decision was made not to carry most of the meta data about the sensor in this media type to help reduce the size of the data and improve efficiency in decoding.

JSON[RFC4627] was selected as a basis for the encoding as it represents a widely understood way of encoding data that is popular in current web based APIs and represents reasonable trade-offs between extensibility, simplicity, and efficiency.
The data is structured as a single JSON object (with attributes) that contains an array of measurements. Each measurement is a JSON object that has attributes such as a unique identifier for the sensor, the time the measurement was made, and the current value. For example, the following shows a measurement from a temperature gauge in JSON syntax.

```
{"m":[{ "n": "0017f202a5c5-Temp", "v":23.5, "u":"degC" }]}
```

In the example above, the array in the object has a single measurement for a sensor named "0017f202a5c5-Temp" with a temperature of 23.5 degrees Celsius.

2. Requirements and Design Goals

The design goal is to be able to send simple sensor measurements in small packets on mesh networks from large numbers of constrained devices. Keeping the total size under 80 bytes makes this easy to use on a wireless mesh network. It is always difficult to define what small code is, but there is a desire to be able to implement this in roughly 1 KB of flash on a 8 bit microprocessor. Experience with Google power meter and other large scale deployments has indicated strongly that the solution needs to support allowing multiple measurements to be batched into a single HTTP request. This "batch" upload capability allows the server side to efficiently support a large number of devices. The multiple measurements could be from multiple related sensors or from the same sensor but at different times.

3. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 *[RFC2119]*.

4. Semantics

Each media type caries a single JSON object that represents a set of measurements. This object contains several optional attributes described below and a mandatory array of one or more measurements.

bn:
This is a base name string that is perpended to the names found in the measurements. This attribute is optional.

bt: A base time that is added to the time found in a measurement. This attribute is optional.

ver: Version number of media type format. This attribute is optional positive integer and defaults to 1 if not present.

m: Array of measurements. Mandatory and there must be at least one measurement in the array.

Each measurement contains several attributes, some of which are optional and some of which are mandatory.
[IEEE.754.1985]. The number of significant digits in any measurement is not relevant, so a reading of 1.1 has exactly the same semantic meaning as 1.10. If the value has an exponent, the "e" MUST be in lower case. The mantissa SHOULD be less than 19 characters long and the exponent SHOULD be less than 5 characters long. This allows time values to have better than micro second precision over the next 100 years.

n: Name of sensor. When appended to the "bn" attribute, this must result in a globally unique identifier for the sensor.

u: Units for the sensor value. Optional. Acceptable values are specified in Section 7.1

v: Value of the sensor. Optional if an s value is present, otherwise required.

s: Integrated sum of the sensor values over time. Optional. This attribute is in the units specified in the u value multiplied by seconds.

t: Time when measurement was made. Optional.

unc: The uncertainty in the measurement, that uses the same units as the base; if absent, the value is unknown (i.e., don't assume that this is zero). Optional.

c: The confidence of the measurement, as a probability between 0.0 and 1.0; if absent, this can be considered to be 0.95. Optional.

ut Update time. A time in seconds that represents the maximum time before this sensor will provide an updated reading. This can be used to detect the failure of sensors or communications path from the sensor. Optional.

The bt, v, s, and t attributes are floating point numbers. Systems receiving measurements MUST be able to process the range of numbers that are representable as an IEEE double-precision floating-point numbers

Systems reading one of the JSON objects MUST check for the ver attribute. If this value is a version number larger than the version which system understands, the system SHOULD NOT use this JSON object. This allows the version number to indicate that the object contains mandatory to understand attributes. New version numbers can only be defined in RFC which updates this specification or it successors.

The n value is concatenated to the bn value to get the name of the sensor. The resulting name needs to uniquely identity and differentiate the sensor from all others. If the name contains 48 bits of random material, or 48 bits of material that is procedurally assigned in a unique way, it is considered to be good enough uniqueness. One way to achieve this uniqueness is to include a EUI-48 identifier (A MAC address) or some other 48 bit identifier that is guaranteed uniqueness (such as a 1-wire address) that is assigned to the device. UUIDs [RFC4122] are another way to generate a unique name.

The resulting concatenated name MUST consist only of characters out of the set "A" to "Z", "a" to "z", "0" to "9", "-", ":", ".", or "_" and it MUST start with a character out of the set "A" to "Z", "a" to "z", or "0" to "9". This restricted character set was chosen so that these names can be directly used as in other types of URI including segments of an HTTP path with no special encoding. [RFC5952] contains advice on encoding an IPv6 address in a name.

If either the bt or t value is missing, the missing attribute is considered to have a value of zero. The bt and t values are added together to get the time of measurement. A time of zero indicates that the sensor does not know the absolute time and the measurement was made roughly "now". A negative value is used to indicate seconds in the past from roughly "now". A positive value is used to indicate the number of seconds, excluding leap seconds, since the start of the year 1970 in UTC .

Representing the statistical characteristics of measurements can be very complex. This specification only provides a very course grain description in the c and unc attributes. Future specification may add new attributes to provide better information about the statistical properties of the measurement. For example, attributes to specify a distribution and it's parameters could be added or a attributes to carry additional properties such as the estimated root mean square error.

5. Syntax

All of the data is UTF-8, but since this is for machine to machine communications on constrained systems, only characters with code points between U+0001 and U+007F are allowed which corresponds to the ASCII[RFC0020] subset of UTF-8.

The contents MUST consist of exactly one JSON object as specified by [RFC4627]. This object MAY contain a "bn" attribute with a value of type string. This object MAY contain a "bt" attribute with a value of type number. The object MAY contain other attribute value pairs. The object MUST contain exactly one "m" attribute with a value of type array. The array MUST have one or more measurement objects.
Inside each measurement object the "n" and "u" attribute are of type string and the "t", "v", and "s" attributes are of type number.

5.1. Simple Example

The following shows a temperature reading taken approximately "now" by a 1-wire sensor device that was assigned the unique 1-wire address of 0x000801EF221E:

```
{"m":[{ "n": "000801EF221E-Temp", "v":23.5 }]}
```

5.2. Complex Example

The following example show the voltage at Tue Jun 8 18:01:16 UTC 2010 along with the current at that time and at each second for the previous 5 seconds. The device has a MAC address of 0017f202b5c4.

```
{"m":[
     { "n": "voltage", "u": "V",
           "v": 120.1, "anExtension": 0.0 },
     { "n": "current", "t": -5, "v": 1.2 },
     { "n": "current", "t": -4, "v": 1.30 },
     { "n": "current", "t": -3, "v": 0.14e1 },
     { "n": "current", "t": -2, "v": 1.5 },
     { "n": "current", "t": -1, "v": 1.6 },
     { "n": "current", "t": 0,   "v": 1.7 }
       ],
 "bn": "0017f202a5c4-",
 "bt": 1276020076,
  "someExtensions": "a value"
}
```

6. Usage Considerations

The measurements support sending both the current value of a sensor as well as the an integrated sum. For many types of measurements, the sum is more useful than the current value. For example, an electrical meter that measures the energy a given computer uses will typically want to measure the cumulative amount of energy used. This is less prone to error than reporting the power each second and trying to have something on the server side sum together all the power measurements. If the network between the sensor and the meter goes down over some period of time, when it comes back up, the cumulative sum helps reflect what happened while the network was down. A meter like this would typically

report a measurement with the units set to watts, but it would put the sum of energy used in the "s" attribute of the measurement. It might optionally include the current power in the "v" attribute.
While the benefit of using the integrated sum is fairly clear for measurements like power and energy, it is less obvious for something like temperature. Reporting the sum of the temperature makes it easy to compute averages even when the individual temperature values are not reported frequently enough to compute accurate averages. Implementors are encouraged to report the cumulative sum as well as the raw value of a given sensor.
Applications that use the cumulative sum values need to understand they are very loosely defined by this specification, and depending on the particular sensor implementation may behave in unexpected ways. Applications should be able to deal with the following issues:

1. Many sensors will allow the cumulative sums to "wrap" back to zero after the value gets sufficiently large.

2. Some sensors will reset the cumulative sum back to zero when the device is reset, loses power, or is replaced with a different sensor.

3. Applications cannot make assumptions about when the device started accumulating values into the sum.

Typically applications can make some assumptions about specific sensors that will allow them to deal with these problems. A common assumption is that for sensors whose measurement values are always positive, the sum should never get smaller; so if the sum does get smaller, the application will know that one of the situations listed above has happened.

7. IANA Considerations

Note to RFC Editor: Please replace all occurrences of "RFC-AAAA" with the RFC number of this specification.

7.1. Units Registry

IANA will create a registry of unit symbols. The primary purpose of this registry is to make sure that symbols uniquely map to give type of measurement. Definitions for many of these units can be found in [NIST822] and [BIPM].

Symbol	Description	Reference
m	meter	RFC-AAAA
kg	kilogram	RFC-AAAA
s	second	RFC-AAAA

Symbol	Description	Reference
A	ampere	RFC-AAAA
K	kelvin	RFC-AAAA
cd	candela	RFC-AAAA
mol	mole	RFC-AAAA
Hz	hertz	RFC-AAAA
rad	radian	RFC-AAAA
sr	steradian	RFC-AAAA
N	newton	RFC-AAAA
Pa	pascal	RFC-AAAA
J	joule	RFC-AAAA
W	watt	RFC-AAAA
C	coulomb	RFC-AAAA
V	volt	RFC-AAAA
F	farad	RFC-AAAA
Ohm	ohm	RFC-AAAA
S	siemens	RFC-AAAA
Wb	weber	RFC-AAAA
T	tesla	RFC-AAAA
H	henry	RFC-AAAA
degC	degrees Celsius	RFC-AAAA
lm	lumen	RFC-AAAA
lx	lux	RFC-AAAA
Bq	becquerel	RFC-AAAA
Gy	gray	RFC-AAAA
Sv	sievert	RFC-AAAA
kat	katal	RFC-AAAA
pH	pH acidity	RFC-AAAA
%	Value of a switch. A value of 0.0 indicates the switch is off while 100.0 indicates on.	RFC-AAAA
count	counter value	RFC-AAAA
%RH	Relative Humidity	RFC-AAAA
m2	area	RFC-AAAA
l	volume in liters	RFC-AAAA
m/s	velocity	RFC-AAAA
m/s2	acceleration	RFC-AAAA

Symbol	Description	Reference
l/s	flow rate in liters per second	RFC-AAAA
W/m2	irradiance	RFC-AAAA
cd/m2	luminance	RFC-AAAA
Bspl	bel sound pressure level	RFC-AAAA
bit/s	bits per second	RFC-AAAA
lat	degrees latitude. Assumed to be in WGS84 unless another reference frame is known for the sensor.	RFC-AAAA
lon	degrees longitude. Assumed to be in WGS84 unless another reference frame is known for the sensor.	RFC-AAAA

New entries can be added to the registration by either Expert Review or IESG Approval as defined in [RFC5226]. Experts should exercise their own good judgement but need to consider the following guidelines:

1. There needs to be a real and compelling use for any new unit to be added.

2. Units should define the semantic information and be chosen carefully. Implementors need to remember that the same word may be used in different real-life contexts. For example, degrees when measuring latitude have no semantic relation to degrees when measuring temperature; thus two different units are needed.

3. These measurements are produced by computers for consumption by computers. The principle is that conversion has to be easily be done when both reading and writing the media type. The value of a single canonical representation outweighs the convenience of easy human representations or loss of precision in a conversion.

4. Use of SI prefixes such as "k" before the unit is not allowed. Instead one can represent the value using scientific notation such a 1.2e3.

5. For a given type of measurement, there will only be one unit type defined. So for length, meters are defined and other lengths such as mile, foot, light year are not allowed. For most cases, the SI unit is preferred.

6. Symbol names that could be easily confused with existing common units or units combined with prefixes should be avoided. For example, selecting a unit name of "mph" to indicate something that had nothing to do with velocity would be a bad choice, as "mph" is commonly used to mean miles per hour.

7. The following should not be used because the are common SI prefixes: Y, Z, E, P, T, G, M, k, h, da, d, c, n, u, p, f, a, z, y, Ki, Mi, Gi, Ti, Pi, Ei, Zi, Yi.

8. The following units should not be used as they are commonly used to represent other measurements Ky, Gal, dyn, etg, P, St, Mx, G, Oe, Gb, sb, Lmb, ph, Ci, R, RAD, REM, gal, bbl, qt, degF, Cal, BTU, HP, pH, B/s, psi, Torr, atm, at, bar, kWh.

9. The unit names are case sensitive and the correct case needs to be used, but symbols that differ only in case should not be allocated.

10. A number after a unit typically indicates the previous unit raised to that power, and the / indicates that the units that follow are the reciprocal. A unit should have only one / in the name.

7.2. Media Type Registration

The following registrations are done following the procedure specified in [RFC4288] and [RFC3023].
Note to RFC Editor: Please replace all occurrences of "RFC-AAAA" with the RFC number of this specification.

7.2.1. senml+json Media Type Registration

To: ietf-types@iana.org
Subject: Registration of media type application/senml+json
Type name: application
Subtype name: senml+json
Required parameters: none
Optional parameters: none
Encoding considerations: Must be encoded as using a subset of the encoding allowed in [RFC4627]. Specifically, only the ASCII[RFC0020] subset of the UTF-8 characters are allowed. This simplifies implementation of very simple system and does not impose any significant limitations as all this data is meant for machine to machine communications and is not meant to be human readable.
Security considerations: Sensor data can contain a wide range of information ranging from information that is very public, such the outside temperature in a given city, to very private information that requires integrity and confidentiality protection, such as patient health information. This format does not provide any security and instead relies on the transport protocol that carries it to provide security. Given applications need to look at the overall context of how this media type will be used to decide if the security is adequate.
Interoperability considerations: Applications should ignore any JSON key value pairs that they do not understand. This allows backwards

compatibility extensions to this specification. The "ver" field can be used to ensure the receiver supports a minimal level of functionality needed by the creator of the JSON object.
Published specification: RFC-AAAA
Applications that use this media type: The type is used by systems that report electrical power usage and environmental information such as temperature and humidity. It can be used for a wide range of sensor reporting systems.
Additional information:
Magic number(s): none
File extension(s): senml
Macintosh file type code(s): none
Person & email address to contact for further information: Cullen Jennings <c.jennings@ieee.org>
Intended usage: COMMON
Restrictions on usage: None
Author: Cullen Jennings <c.jennings@ieee.org>
Change controller: IESG

8. Security Considerations

See Section 9.Further discussion of security proprieties can be found in Section 7.2.

9. Privacy Considerations

Sensor data can range from information with almost no security considerations, such as the current temperature in a given city, to highly sensitive medical or location data. This specification provides no security protection for the data but is meant to be used inside another container or transport protocol such as S/MIME or HTTP with TLS that can provide integrity, confidentiality, and authentication information about the source of the data.

10. Acknowledgement

I would like to thank Lisa Dusseault, Joe Hildebrand, Lyndsay Campbell, Martin Thomson, John Klensin, Bjoern Hoehrmann, and Carsten Bormann for their review comments.

11. References

11.1. Normative References

[RFC4627]	Crockford, D., "The application/json Media Type for JavaScript Object Notation (JSON)", RFC 4627, July 2006.
[RFC3023]	Murata, M., St. Laurent, S. and D. Kohn, "XML Media Types", RFC 3023, January 2001.

[RFC4288]	Freed, N. and J. Klensin, "Media Type Specifications and Registration Procedures", BCP 13, RFC 4288, December 2005.
[RFC2119]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
[IEEE. 754.1985]	Institute of Electrical and Electronics Engineers, "Standard for Binary Floating-Point Arithmetic", IEEE Standard 754, August 1985.
[RFC5226]	Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

11.2. Informative References

[I-D.ietf-core-coap]	Shelby, Z, Frank, B and D Sturek, "Constrained Application Protocol (CoAP)", Internet-Draft draft-ietf-core-coap-04, January 2011.
[BIPM]	Bureau International des Poids et Mesures, "The International System of Units (SI)", 8th edition, 2006 , .
[NIST822]	Thompson, A. and B. Taylor, "Guide for the Use of the International System of Units (SI)", NIST Special Publication 811, 2008 Edition , .
[RFC5952]	Kawamura, S. and M. Kawashima, "A Recommendation for IPv6 Address Text Representation", RFC 5952, August 2010.
[RFC4122]	Leach, P., Mealling, M. and R. Salz, "A Universally Unique IDentifier (UUID) URN Namespace", RFC 4122, July 2005.
[RFC0020]	Cerf, V., "ASCII format for network interchange", RFC 20, October 1969.

Author's Address

Cullen Jennings Jennings Cisco 170 West Tasman Drive San Jose, CA 95134 USA Phone: +1 408 421-9990 EMail: fluffy@cisco.com