[←](https://iot.mozilla.org/wot/" \l "toc)

ReSpec6

**Web Thing API**

Unofficial Draft 06 March 2020

**Latest editor's draft:**

<https://iot.mozilla.org/wot/>

**Editor:**

Ben Francis ([Mozilla Corporation](https://mozilla.com/))

Copyright © 2017 Mozilla

Abstract

This document describes a common data model and API for the Web of Things. The [Web Thing Description](https://iot.mozilla.org/wot/#web-thing-description) provides a vocabulary for describing physical devices connected to the World Wide Web in a machine readable format with a default JSON encoding. Common device capabilities can be specified using optional [semantic annotations](https://iot.mozilla.org/wot/#context-member). The [Web Thing REST API](https://iot.mozilla.org/wot/#web-thing-rest-api) and [Web Thing WebSocket API](https://iot.mozilla.org/wot/#web-thing-websocket-api) allow a web client to access the properties of devices, request the execution of actions and subscribe to events representing a change in state..

Status of This Document

This document is draft of a potential specification. It has no official standing of any kind and does not represent the support or consensus of any standards organization.

This document includes a proposed plain JSON serialisation of a [Thing Description](https://w3c.github.io/wot-thing-description/) and a concrete HTTP and WebSockets [protocol binding](https://w3c.github.io/wot-binding-templates/) for the Web of Things. This proposal is intended to complement the current [W3C Web of Things Working Group](https://www.w3.org/WoT/WG/)'s work on an abstract data model and API for the Web of Things, by defining a simple concrete serialisation and protocol binding using existing web technologies which can be used today, and don't require web clients to implement a new [Scripting API](https://w3c.github.io/wot-scripting-api/), multiple non-web IoT protocols, or an RDF-based parser.

While the standardisation process continues, this document will continue to be maintained to reflect the current API implemented by Mozilla's own open source Web of Things implementation through [Project Things](https://iot.mozilla.org/). We encourage developers to continue to implement this API to build web things compatible with Mozilla's [Things Gateway](https://iot.mozilla.org/gateway/), and provide feedback to help us further improve the specification.

**table of contents**

1. [**1.Introduction**](https://iot.mozilla.org/wot/#introduction)
2. [**2.Web Thing Description**](https://iot.mozilla.org/wot/#web-thing-description)
   1. [2.1@context member](https://iot.mozilla.org/wot/#context-member)
   2. [2.2@type member](https://iot.mozilla.org/wot/#type-member)
   3. [2.3id member](https://iot.mozilla.org/wot/#id-member)
   4. [2.4title member](https://iot.mozilla.org/wot/#title-member)
   5. [2.5description member](https://iot.mozilla.org/wot/#description-member)
   6. [2.6properties member](https://iot.mozilla.org/wot/#properties-member)
   7. [2.7actions member](https://iot.mozilla.org/wot/#actions-member)
   8. [2.8events member](https://iot.mozilla.org/wot/#events-member)
   9. [2.9links member](https://iot.mozilla.org/wot/#links-member)
   10. [2.10Property object](https://iot.mozilla.org/wot/#property-object)
   11. [2.11Action object](https://iot.mozilla.org/wot/#action-object)
   12. [2.12Event object](https://iot.mozilla.org/wot/#event-object)
   13. [2.13Link object](https://iot.mozilla.org/wot/#link-object)
   14. [2.14Media Type](https://iot.mozilla.org/wot/#media-type)
   15. [2.15Alternative Encodings](https://iot.mozilla.org/wot/#alternative-encodings)
3. [**3.Web Thing REST API**](https://iot.mozilla.org/wot/#web-thing-rest-api)
   1. [3.1Thing resource](https://iot.mozilla.org/wot/#thing-resource)
   2. [3.2Properties resource](https://iot.mozilla.org/wot/#properties-resource)
   3. [3.3Property resource](https://iot.mozilla.org/wot/#property-resource)
   4. [3.4Actions resource](https://iot.mozilla.org/wot/#actions-resource)
   5. [3.5Action resource](https://iot.mozilla.org/wot/#action-resource)
   6. [3.6ActionRequest resource](https://iot.mozilla.org/wot/#actionrequest-resource)
   7. [3.7Events resource](https://iot.mozilla.org/wot/#events-resource)
   8. [3.8Event resource](https://iot.mozilla.org/wot/#event-resource)
   9. [3.9Things resource](https://iot.mozilla.org/wot/#things-resource)
   10. [3.10Alternative Protocol Bindings](https://iot.mozilla.org/wot/#alternative-protocol-bindings)
4. [**4.Web Thing WebSocket API**](https://iot.mozilla.org/wot/#web-thing-websocket-api)
   1. [4.1Protocol Handshake](https://iot.mozilla.org/wot/#protocol-handshake)
   2. [4.2setProperty message](https://iot.mozilla.org/wot/#setproperty-message)
   3. [4.3requestAction message](https://iot.mozilla.org/wot/#requestaction-message)
   4. [4.4addEventSubscription message](https://iot.mozilla.org/wot/#addeventsubscription-message)
   5. [4.5propertyStatus message](https://iot.mozilla.org/wot/#propertystatus-message)
   6. [4.6actionStatus message](https://iot.mozilla.org/wot/#actionstatus-message)
   7. [4.7event message](https://iot.mozilla.org/wot/#event-message)
5. [**5.Further Information**](https://iot.mozilla.org/wot/#further-information)
6. [**A.References**](https://iot.mozilla.org/wot/#references)
   1. [A.1Normative references](https://iot.mozilla.org/wot/#normative-references)
   2. [A.2Informative references](https://iot.mozilla.org/wot/#informative-references)

1. Introduction

The goal of the Web of Things is to extend the web of pages into a web of things by giving connected devices URLs on the World Wide Web. This will allow the web to be used as a unifying application layer for a decentralized Internet of Things.

Whilst web technologies are already in widespread use on the Internet of Things, this is currently done with mostly proprietary data formats and APIs which require per-vendor integrations to make devices interoperable. In order to promote ad-hoc interoperability on the Internet of Things a shared vocabulary and common API is needed.

In this document we propose a common data model and API for the Web of Things.

2. Web Thing Description

The Thing Description provides a vocabulary for describing physical devices connected to the World Wide Web in a machine readable format with a default JSON encoding.

[EXAMPLE 1](https://iot.mozilla.org/wot/#example-1-a-simple-thing-description): A simple Thing Description

{

"id": "https://mywebthingserver.com/things/switch",

"title": "On/Off Switch",

"description": "A web connected switch",

"properties": {

"on": {

"title": "On/Off",

"type": "boolean",

"description": "Whether the lamp is turned on",

"links": [{"href": "/things/switch/properties/on"}]

}

}

}

[EXAMPLE 2](https://iot.mozilla.org/wot/#example-2-a-more-complex-thing-description-with-semantic-annotations): A more complex Thing Description with semantic annotations

{

"@context": "https://iot.mozilla.org/schemas/",

"@type": ["Light", "OnOffSwitch"],

"id": "https://mywebthingserver.com/things/lamp",

"title": "My Lamp",

"description": "A web connected lamp",

"properties": {

"on": {

"@type": "OnOffProperty",

"type": "boolean",

"title": "On/Off",

"description": "Whether the lamp is turned on",

"links": [{"href": "/things/lamp/properties/on"}]

},

"brightness" : {

"@type": "BrightnessProperty",

"type": "integer",

"title": "Brightness",

"description": "The level of light from 0-100",

"minimum" : 0,

"maximum" : 100,

"links": [{"href": "/things/lamp/properties/brightness"}]

}

},

"actions": {

"fade": {

"@type": "FadeAction",

"title": "Fade",

"description": "Fade the lamp to a given level",

"input": {

"type": "object",

"properties": {

"level": {

"type": "integer",

"minimum": 0,

"maximum": 100

},

"duration": {

"type": "integer",

"minimum": 0,

"unit": "milliseconds"

}

}

},

"links": [{"href": "/things/lamp/actions/fade"}]

}

},

"events": {

"overheated": {

"title": "Overheated",

"@type": "OverheatedEvent",

"type": "number",

"unit": "degree celsius",

"description": "The lamp has exceeded its safe operating temperature",

"links": [{"href": "/things/lamp/events/overheated"}]

}

},

"links": [

{

"rel": "properties",

"href": "/things/lamp/properties"

},

{

"rel": "actions",

"href": "/things/lamp/actions"

},

{

"rel": "events",

"href": "/things/lamp/events"

},

{

"rel": "alternate",

"href": "wss://mywebthingserver.com/things/lamp"

},

{

"rel": "alternate",

"mediaType": "text/html",

"href": "/things/lamp"

}

]

}

2.1 @context member

The @context member is an optional annotation which can be used to provide a URI for a schema repository which defines standard schemas for common "types" of device capabilities. These types can then be referred to using @type annotations elsewhere in the Thing Description.

NOTE

Example experimental schema repositories can be found at <http://iotschema.org/> and <https://iot.mozilla.org/schemas>.

[EXAMPLE 3](https://iot.mozilla.org/wot/#example-3-example-context-member): Example @context member

"@context": "https://iot.mozilla.org/schemas"

2.2 @type member

The @type member is an optional annotation which can be used to provide the names of schemas for types of capabilities a device supports, from a schema repository referred to in the @context member. The value of the @type member is a string or an array of strings which correspond to schema names from a schema repository.

Providing a @type annotation tells a client that it can expect the device to conform to the constraints of that type's schema. This may include certain types of property, action and event the device can be expected to support. A client may make use of this information to aid in automation of or to generate a user interface for known standard device capabilities.

[EXAMPLE 4](https://iot.mozilla.org/wot/#example-4-example-type-member): Example @type member

"@type": ["Light", "OnOffSwitch"]

2.3 id member

The id member provides an identifier of the device in the form of a URI [[RFC3986](https://iot.mozilla.org/wot/#bib-rfc3986)] (e.g. a URL or a URN). This may be the URL of the Thing Description itself, but can be any URI.

[EXAMPLE 5](https://iot.mozilla.org/wot/#example-5-example-id-member): Example id member

"id": "https://mywebthingserver.com/things/lamp"

2.4 title member

The title member is a human friendly string which describes the device. This can be set to any value by the device creator and may include a brand name or model number.

2.5 description member

The description member is a human friendly string which describes the device and its functions. This can be set to any value by the device creator.

2.6 properties member

The properties member is a map of [Property objects](https://iot.mozilla.org/wot/" \l "property-object) which describe the attributes of the device.

2.7 actions member

The actions member is a map of [Action objects](https://iot.mozilla.org/wot/#action-object) which describe functions that can be carried out on a device. Actions are used when the setting of a property alone is not sufficient to affect a required change in state. This may be used to describe a function which takes a period of time to complete, manipulates multiple properties, or has a different outcome based on provided arguments or current state. An example might include a "fade" action which has a specified "duration", a "sequence" action which triggers a sequence of flashing lights or a "toggle" action on a switch.

2.8 events member

The events member is a map of [Event objects](https://iot.mozilla.org/wot/#event-object) which define the types of events which may be emitted by a device. Events can be used where subscribing to a change in property state is not sufficient. This may be used to monitor changes in multiple properties or events which describe something other than a property change. An example might be an event which is emitted when a device is about to reboot.

2.9 links member

The links member provides an array of [Link Object](https://iot.mozilla.org/wot/#link-object)s which link to other resources of a thing.

Examples of links include a link to a [Properties resource](https://iot.mozilla.org/wot/" \l "properties-resource) ("rel":"properties"), an [Actions resource](https://iot.mozilla.org/wot/#actions-resource) ("rel":"actions") or an [Events resource](https://iot.mozilla.org/wot/#events-resource) ("rel":"events"). It can also include links to alternate representations of a thing such as a WebSocket API endpoint or an HTML user interface ("rel":"alternate").

[EXAMPLE 6](https://iot.mozilla.org/wot/#example-6-example-links-member): Example links member

"links": [

{

"rel": "properties",

"href": "/things/lamp/properties"

},

{

"rel": "actions",

"href": "/things/lamp/actions"

},

{

"rel": "events",

"href": "/things/lamp/events"

},

{

"rel": "alternate",

"href": "wss://mywebthingserver.com/things/lamp"

},

{

"rel": "alternate",

"mediaType": "text/html",

"href": "/things/lamp"

}

]

2.10 Property object

A property object describes an attribute of a Thing and is indexed by a property id. A property description may include:

* A primitive type (one of null, boolean, object, array, number, integer or string as per [[json-schema](https://iot.mozilla.org/wot/" \l "bib-json-schema" \o "JSON Schema: core definitions and terminology)])
* A semantic @type (a string identifying a type from the linked [@context](https://iot.mozilla.org/wot/#context-member))
* A unit ([[SI](https://iot.mozilla.org/wot/#bib-si)] unit)
* A title (A string providing a human friendly name)
* A description (A string providing a human friendly description)
* A links array (An array of [Link object](https://iot.mozilla.org/wot/#link-object)s linking to one or more representations of a [Property resource](https://iot.mozilla.org/wot/" \l "property-resource), each with an implied default rel=property.)
* enum (an enumeration of possible values for the property)
* readOnly (A boolean indicating whether or not the property is read-only, defaulting to false)
* A minimum and maximum (numeric values)
* multipleOf (A number indicating what the value must be a multiple of)

[EXAMPLE 7](https://iot.mozilla.org/wot/#example-7-property-object): Property Object

"level" : {

"title": "Level",

"description": "The level of light from 0-100",

"@type": "LevelProperty",

"type": "integer",

"unit": "percent",

"minimum": 0,

"maximum": 100,

"readOnly": false,

"links": [{"href": "/things/lamp/properties/level"}]

}

2.11 Action object

An action object describes a function which can be carried out on a device. An action definition may include:

* A title (A string providing a human friendly name)
* A description (A string providing a human friendly description)
* A links array (An array of [Link object](https://iot.mozilla.org/wot/#link-object)s linking to one or more representations of an [Action resource](https://iot.mozilla.org/wot/#action-resource), each with an implied default rel=action.)
* An input object with a:
  + A primitive type (one of null, boolean, object, array, number, integer or string as per [[json-schema](https://iot.mozilla.org/wot/" \l "bib-json-schema" \o "JSON Schema: core definitions and terminology)])
  + A semantic @type (a string identifying a type from the linked [@context](https://iot.mozilla.org/wot/#context-member))
  + A unit ([[SI](https://iot.mozilla.org/wot/#bib-si)] unit)
  + A minimum and maximum (numeric values)
  + multipleOf (A number indicating what the value must be a multiple of)

[EXAMPLE 8](https://iot.mozilla.org/wot/#example-8-action-object): Action Object

"fade": {

"title": "Fade",

"description": "Fade the lamp to a given level",

"input": {

"@type": "FadeAction",

"type": "object",

"properties": {

"level": {

"type": "integer",

"minimum": 0,

"maximum": 100

},

"duration": {

"type": "integer",

"minimum": 0,

"unit": "milliseconds"

}

}

},

"links": [{"href": "/things/lamp/actions/fade"}]

}

2.12 Event object

An event object describes a kind of event which may be emitted by a device. This may include:

* A primitive type (one of null, boolean, object, array, number, integer or string as per [[json-schema](https://iot.mozilla.org/wot/" \l "bib-json-schema" \o "JSON Schema: core definitions and terminology)])
* A semantic @type (a string identifying a type from the linked [@context](https://iot.mozilla.org/wot/#context-member))
* A unit ([[SI](https://iot.mozilla.org/wot/#bib-si)] unit)
* A title (A string providing a human friendly name)
* A description (A string providing a human friendly description)
* A links array (An array of [Link object](https://iot.mozilla.org/wot/#link-object)s linking to one or more representations of an [Event resource](https://iot.mozilla.org/wot/#event-resource), each with an implied default rel=event).
* A minimum and maximum (numeric values)
* multipleOf (A number indicating what the value must be a multiple of)

[EXAMPLE 9](https://iot.mozilla.org/wot/#example-9-event-object): Event Object

"overheated": {

"title": "Overheated",

"description": "The lamp has exceeded its safe operating temperature",

"@type": "OverheatedEvent",

"type": "number",

"unit": "degree celsius",

"links": [{"href": "/things/lamp/events/overheated"}]

}

2.13 Link object

A link object represents a link relation ([[web-linking](https://iot.mozilla.org/wot/#bib-web-linking)]) and may have a:

* href (a string representation of a URL)
* rel (a string describing a relationship)
* mediaType (a string identifying a media type)

[EXAMPLE 10](https://iot.mozilla.org/wot/#example-10-link-object): Link Object

{

"href": "/things/lamp/properties",

"rel": "properties",

"mediaType": "application/json"

}

2.14 Media Type

The JSON serialization of the Web Thing Description is identified by the Media Type application/td+json.

NOTE

This media type has not yet been registered with IANA.

2.15 Alternative Encodings

The default encoding for the Thing description is JSON, but other encodings such as XML, JSON-LD or CBOR may be used. Alternative encodings are beyond the scope of this specification but may be defined separately.

References to alternative available encodings may be provided using Link HTTP response headers with the alternate link relation. A client can express a preference for an alternative encoding using HTTP content negotiation.

3. Web Thing REST API

The Web Thing API provides a web services based programming interface with a RESTful design for applications to access the properties of devices, request the execution of actions and access a list of events representing a change in state. A default HTTP [[HTTP11](https://iot.mozilla.org/wot/#bib-http11)] protocol binding is defined here.

The Web Thing REST API consists of a number of different types of resources which represent different aspects of a device and can be independently referenced by a URL. The specific URLs and URL structure of resources are defined by the Thing Description. This specification does **not** define a fixed URL structure.

3.1 Thing resource

A thing Resource provides a [Thing Description](https://iot.mozilla.org/wot/" \l "web-thing-description) for a device. A Thing Resource is considered the root resource of a Thing.

The URL of a Thing Resource may be enumerated by the href member of Thing object in a [Things Resource](https://iot.mozilla.org/wot/" \l "things-resource) provided by a gateway or directory, or may be discovered by some other means.

A Thing description is usually read only. An HTTP GET request can be used to retrieve a Thing Description for a Thing.

**Example: Get a description of a Thing**

REQUEST

GET http:*//mythingserver.com/things/pi*

Accept: application/json

RESPONSE

200 OK

{

"id": "https://mywebthingserver.com/things/pi",

"title": "WoT Pi",

"description": "A WoT-connected Raspberry Pi",

"properties": {

"temperature": {

"title": "Temperature",

"type": "number",

"unit": "degree celsius",

"readOnly": true,

"description": "An ambient temperature sensor",

"links": [{"href": "/things/pi/properties/temperature"}]

},

"humidity": {

"title": "Humidity",

"type": "number",

"unit": "percent",

"readOnly": true,

"links": [{"href": "/things/pi/properties/humidity"}]

},

"led": {

"title": "LED",

"type": "boolean",

"description": "A red LED",

"links": [{"href": "/things/pi/properties/led"}]

}

},

"actions": {

"reboot": {

"title": "Reboot",

"description": "Reboot the device"

}

},

"events": {

"reboot": {

"description": "Going down for reboot"

}

},

"links": [

{

"rel": "properties",

"href": "/things/pi/properties"

},

{

"rel": "actions",

"href": "/things/pi/actions"

},

{

"rel": "events",

"href": "/things/pi/events"

},

{

"rel": "alternate",

"href": "wss://mywebthingserver.com/things/pi"

},

{

"rel": "alternate",

"mediaType": "text/html",

"href": "/things/pi"

}

]

}

3.2 Properties resource

A properties resource represents all the properties of a device. The value of all properties can be retrieved with an HTTP GET request.

The URL of a Properties resource can be defined by a properties link relation in the [links member](https://iot.mozilla.org/wot/#links-member) of a Thing Description.

**Example: Get all properties**

REQUEST

GET http:*//mythingserver.com/things/pi/properties*

Accept: application/json

RESPONSE

200 OK

{

"temperature": 21,

"humidity": 50,

"led": true

}

3.3 Property resource

A property resource represents a single property of a device. Some property resources may be read only and some may be writable. The value of a property can be retrieved with an HTTP GET request and updated with an HTTP PUT request.

The URL of a Property resource can be defined by the links section of a [Property object](https://iot.mozilla.org/wot/" \l "property-object) in a Thing Description.

If a Property object does not define a primitive type it may not have a JSON serialisation and may instead return a binary file or stream in response to an HTTP GET request (e.g. an image file or video stream).

**Example: Get a property**

REQUEST

GET http:*//mythingserver.com/things/pi/properties/temperature*

Accept: application/json

RESPONSE

200 OK

{

"temperature": 21

}

**Example: Set a property**

REQUEST

PUT http:*//mythingserver.com/things/pi/properties/led*

{

"led": true

}

Accept: application/json

RESPONSE

200 OK

{

"led": true

}

3.4 Actions resource

An actions resource represents a queue of actions to be executed on a device. A new action is created in the queue with an HTTP POST request and a list of action requests in the queue can be requested with an HTTP GET request.

The URL of an actions resource can be defined by an actions link relation in the [links member](https://iot.mozilla.org/wot/#links-member) of a Thing Description.

Any action supported by the thing can be requested via this top level action queue. If an unsupported action type is requested, the server should respond with a 400 Bad Request response.

**Action Request**

[EXAMPLE 11](https://iot.mozilla.org/wot/#example-11-request-an-action): Request an action

POST https:*//mythingserver.com/things/lamp/actions/*

Accept: application/json

{

"fade": {

"input": {

"level": 50,

"duration": 2000

}

}

}

[EXAMPLE 12](https://iot.mozilla.org/wot/#example-12-response-to-creating-an-action-request): Response to creating an action request

201 Created

{

"fade": {

"input": {

"level": 50,

"duration": 2000

},

"href": "/things/lamp/actions/fade/123e4567-e89b-12d3-a456-426655",

"status": "pending"

}

}

**Actions Queue**

[EXAMPLE 13](https://iot.mozilla.org/wot/#example-13-get-a-list-of-all-action-requests): Get a list of all action requests

GET /things/lamp/actions

Accept: application/json

[EXAMPLE 14](https://iot.mozilla.org/wot/#example-14-action-list-response): Action list response

200 OK

[

{

"fade": {

"input": {

"level": 50,

"duration": 2000

},

"href": "/things/lamp/actions/fade/123e4567-e89b-12d3-a456-426655",

"timeRequested": "2017-01-25T15:01:35+00:00",

"status": "pending"

}

},

{

"reboot": {

"href": "/things/lamp/actions/reboot/124e4568-f89b-22d3-a356-427656",

"timeRequested": "2017-01-24T13:13:33+00:00",

"timeCompleted": "2017-01-24T13:15:01+00:00",

"status": "completed"

}

}

]

3.5 Action resource

An action resource represents a queue of actions of a single action type. A new action is created in the queue with an HTTP POST request and a list of action requests in the queue can be requested with an HTTP GET request.

The URL of an Action resource can be defined by the links section of an [Action object](https://iot.mozilla.org/wot/#action-object) in a Thing Description.

If a client tries to request an action of another type via this resource, the server should respond with a 400 Bad Request response.

**Action Request**

[EXAMPLE 15](https://iot.mozilla.org/wot/#example-15-request-an-action): Request an action

POST https:*//mythingserver.com/things/lamp/actions/fade*

Accept: application/json

{

"fade": {

"input": {

"level": 50,

"duration": 2000

}

}

}

[EXAMPLE 16](https://iot.mozilla.org/wot/#example-16-response-to-creating-an-action-request): Response to creating an action request

201 Created

{

"fade": {

"input": {

"level": 50,

"duration": 2000

},

"href": "/things/lamp/actions/fade/123e4567-e89b-12d3-a456-426655"

"status": "pending"

}

}

**Action Queue**

[EXAMPLE 17](https://iot.mozilla.org/wot/#example-17-get-a-list-of-all-action-requests-for-a-given-action): Get a list of all action requests for a given action

GET /things/lamp/actions/fade

Accept: application/json

[EXAMPLE 18](https://iot.mozilla.org/wot/#example-18-action-list-response): Action list response

200 OK

[

{

"fade": {

"input": {

"level": 50,

"duration": 2000

},

"href": "/things/lamp/actions/fade/123e4567-e89b-12d3-a456-426655",

"timeRequested": "2017-01-25T15:01:35+00:00",

"status": "pending"

}

},

{

"fade": {

"input": {

"level": 100,

"duration": 2000

},

"href": "/things/lamp/actions/fade/123e4567-e89b-12d3-a456-426655",

"timeRequested": "2017-01-24T11:02:45+00:00",

"timeCompleted": "2017-01-24T11:02:46+00:00",

"status": "completed"

}

}

]

3.6 ActionRequest resource

An action request resource represents an individual action request for a given action. The current status of an action request can be retrieved with an HTTP GET request, updated with an HTTP PUT request and deleted with an HTTP DELETE request.

The URL of an Action request resource can be defined by the href member of an action request object in an action queue.

**Action Request Status**

[EXAMPLE 19](https://iot.mozilla.org/wot/#example-19-get-the-status-of-an-action-request): Get the status of an action request

GET /things/lamp/actions/fade/123e4567-e89b-12d3-a456-426655

Accept: application/json

[EXAMPLE 20](https://iot.mozilla.org/wot/#example-20-action-request-status-response): Action request status response

200 OK

{

"fade": {

"input": {

"level": 50,

"duration": 2000

},

"href": "/things/lamp/actions/fade/123e4567-e89b-12d3-a456-426655",

"timeRequested": "2017-01-25T15:01:35+00:00",

"status": "pending"

}

}

**Cancel an Action Request**

[EXAMPLE 21](https://iot.mozilla.org/wot/#example-21-cancel-an-action-request-response): Cancel an action request response

DELETE /things/lamp/actions/fade/123e4567-e89b-12d3-a456-426655

[EXAMPLE 22](https://iot.mozilla.org/wot/#example-22-action-cancellation-response): Action cancellation response

204 No Content

3.7 Events resource

An events resource provides a log of all events recently emitted by a device. An events resource is usually read only.

The URL of an Events resource can be defined by an events link relation in the [links member](https://iot.mozilla.org/wot/#links-member) of a Thing Description.

**Event Log**

[EXAMPLE 23](https://iot.mozilla.org/wot/#example-23-events-request): Events Request

GET /things/lamp/events

Accept: application/json

[EXAMPLE 24](https://iot.mozilla.org/wot/#example-24-events-response): Events Response

200 OK

[

{

"overheated": {

"data": 102,

"timestamp": "2017-01-25T15:01:35+00:00"

}

},

{

"reboot": {

"timestamp": "2017-01-24T13:02:45+00:00"

}

}

]

3.8 Event resource

An event resource provides a log of events recently emitted by a device for a particular event type. An event resource is usually read only.

The URL of an Event resource can be defined by the links section of an [Event object](https://iot.mozilla.org/wot/#event-object) in a Thing Description.

**Event Log**

[EXAMPLE 25](https://iot.mozilla.org/wot/#example-25-event-request): Event Request

GET /things/lamp/events/overheated

Accept: application/json

[EXAMPLE 26](https://iot.mozilla.org/wot/#example-26-event-response): Event Response

200 OK

[

{

"overheated": {

"data": 102,

"timestamp": "2017-01-25T15:01:35+00:00"

}

},

{

"overheated": {

"data": 101,

"timestamp": "2017-01-24T13:02:45+00:00"

}

}

]

3.9 Things resource

A things resource represents a collection of web things exposed by a particular Web of Things server. This resource is not needed for individual web things, for which the Thing Description is the entry point. It is used only by Web of Things gateways and services which expose a directory of web things.

**Example: Get a list of things**

REQUEST

GET /things

Accept: application/json

RESPONSE

200 OK

[

{

"id": "https://mywebthingserver.com/things/pi",

"title": "WoT Pi",

"description": "A WoT-connected Raspberry Pi",

"properties": {

"temperature": {

"title": "Temperature",

"type": "number",

"unit": "degree celsius",

"readOnly": true,

"description": "An ambient temperature sensor",

"links": [{"href": "/things/pi/properties/temperature"}]

},

"humidity": {

"title": "Humidity",

"type": "number",

"unit": "percent",

"readOnly": true,

"links": [{"href": "/things/pi/properties/humidity"}]

},

"led": {

"title": "LED",

"type": "boolean",

"description": "A red LED",

"links": [{"href": "/things/pi/properties/led"}]

}

},

"actions": {

"reboot": {

"title": "Reboot",

"description": "Reboot the device"

}

},

"events": {

"reboot": {

"description": "Going down for reboot"

}

},

"href": "/things/pi"

}

]

3.10 Alternative Protocol Bindings

The default protocol binding for the Web Thing REST API is HTTP(S). Bindings to alternative application protocols (e.g. CoAP) may be used, but these bindings are beyond the scope of this specification. A Web Thing API protocol binding may also be layered on top of a non-Internet application protocol by use of a gateway.

4. Web Thing WebSocket API

The Web Thing WebSocket API complements the REST API to provide a realtime mechanism to make multiple requests and be notified of events as soon as they happen, by keeping a WebSocket [[WEBSOCKETS-PROTOCOL](https://iot.mozilla.org/wot/#bib-websockets-protocol)] open on the Web Thing. The "webthing" WebSocket subprotocol defined here has a simple set of message types and a JSON response format consistent with the Web Thing REST API.

4.1 Protocol Handshake

To open a WebSocket on a Thing, an HTTP GET request is upgraded to a WebSocket using a standard WebSocket protocol handshake [[WEBSOCKETS-PROTOCOL](https://iot.mozilla.org/wot/#bib-websockets-protocol)] and the "webthing" subprotocol. The WebSocket URL for a Web Thing is specified in the links member of the Web Thing Description.

REQUEST

GET wss:*//mythingserver.com/things/robot*

Host: mythingserver.com

Origin: https:*//mythingserver.com*

Upgrade: websocket

Connection: Upgrade

Sec-WebSocket-Key: x3JJHMbDL1EzLkh9GBhXDw==

Sec-WebSocket-Protocol: webthing

Sec-WebSocket-Version: 13

RESPONSE

HTTP 101 Switching Protocols

Upgrade: websocket

Connection: Upgrade

Sec-WebSocket-Accept: HSmrc0sMlYUkAGmm5OPpG2HaGWk=

Sec-WebSocket-Protocol: webthing

A WebSocket can be opened from a web page using the JavaScript [WebSocket API](https://www.w3.org/TR/websockets/) which will take care of the handshake detailed above and allow messages to be sent and received.

EXAMPLE

const socket = new WebSocket('wss://mywebthingserver/things/robot', 'webthing');

4.2 setProperty message

The setProperty message type is sent from a web client to a Web Thing in order to set the value of one or more of its properties. This is equivalent to a PUT request on a Property resource URL using the REST API, but with the WebSocket API a property value can be changed multiple times in quick succession over an open socket and multiple properties can be set at the same time.

EXAMPLE

{

"messageType": "setProperty",

"data": {

"leftMotor": 100

}

}

4.3 requestAction message

The requestAction message type is sent from a web client to a Web Thing to request an action be carried out on a Web Thing. This is equivalent to a POST request on an Actions resource URL using the REST API, but multiple actions can be requested at the same time or in quick succession over an open socket.

EXAMPLE

{

"messageType": "requestAction",

"data": {

"goForward": {},

}

}

4.4 addEventSubscription message

The addEventSubscription message type is sent from a web client to a Web Thing to allow a client to subscribe to a particular event type, defined by the events member of a Web Thing description. This is similar to adding an event listener in JavaScript, but events are received as an event message over the Web Thing WebSocket API.

EXAMPLE

{

"messageType": "addEventSubscription",

"data": {

"motion": {}

}

}

4.5 propertyStatus message

The propertyStatus message type is sent from a Web Thing to a web client whenever a property of a Web Thing changes. The payload data of this message is in the same format as a response to a GET request on Property resource using the REST API, but the message is pushed to the client whenever a property changes and can include multiple properties at the same time.

EXAMPLE

{

"messageType": "propertyStatus",

"data": {

"led": true

}

}

4.6 actionStatus message

The actionStatus message type is sent from a Web Thing to a web client when the status of a requested action changes. The payload data is consistent with the format of an Action resource in the REST API, but messages are pushed to the client as soon as the status of an action changes.

EXAMPLE

{

"messageType": "actionStatus",

"data": {

"grab": {

"href": "/actions/grab/123e4567-e89b-12d3-a456-426655",

"status": "completed",

"timeRequested": "2017-01-24T11:02:45+00:00",

"timeCompleted": "2017-01-24T11:02:46+00:00"

}

}

}

4.7 event message

The event message type is sent from a Web Thing to a web client when an event occurs on the Web Thing. The payload data is consistent with the format of an Event resource in the REST API but messages are pushed to the client as soon as an event occurs.

EXAMPLE

{

"messageType": "event",

"data": {

"motion": {

"timestamp": "2017-01-24T13:02:45+00:00"

}

}

}

5. Further Information

[Web of Things Integration Patterns](https://docs.google.com/document/d/1H3coHbb3Bwd02_NJi4KEBONByUkq92_HsTk1IpfmACY/edit?usp=sharing)  
Advice on different design patterns for integrating connected devices with the Web of Things.

A. References

A.1 Normative references

**[HTTP11]**

[*Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing*](https://httpwg.org/specs/rfc7230.html). R. Fielding, Ed.; J. Reschke, Ed.. IETF. June 2014. Proposed Standard. URL: <https://httpwg.org/specs/rfc7230.html>

**[json-schema]**

[*JSON Schema: core definitions and terminology*](https://tools.ietf.org/html/draft-zyp-json-schema). K. Zyp. Internet Engineering Task Force (IETF). 31 January 2013. Internet-Draft. URL: <https://tools.ietf.org/html/draft-zyp-json-schema>

**[SI]**

[*SI Brochure: The International System of Units (SI), 8th edition*](http://www.bipm.org/en/publications/si-brochure/). BIPM. 2014. URL: <http://www.bipm.org/en/publications/si-brochure/>

**[web-linking]**

[*Web Linking*](https://tools.ietf.org/html/rfc5988). IETF. URL: <https://tools.ietf.org/html/rfc5988>

**[WEBSOCKETS-PROTOCOL]**

[*The WebSocket Protocol*](https://tools.ietf.org/html/rfc6455). I. Fette; A. Melnikov. IETF. December 2011. Proposed Standard. URL: <https://tools.ietf.org/html/rfc6455>

A.2 Informative references

**[RFC3986]**

[*Uniform Resource Identifier (URI): Generic Syntax*](https://tools.ietf.org/html/rfc3986). T. Berners-Lee; R. Fielding; L. Masinter. IETF. January 2005. Internet Standard. URL: <https://tools.ietf.org/html/rfc3986>

[↑](https://iot.mozilla.org/wot/#title)