

T2TRG | IETF 104

Constrained RESTful Application Language



Hypermedia

JSON Hypertext Application Language

draft-kelly-json-hal-08

JSON Hyper-Schema

draft-handrews-json-schema-hyperschema-01

Profiled Hypertext Application Language

draft-montoya-phtal-00

Constrained RESTful Application Language

draft-hartke-t2trg-coral-08

Web Linking

RFC 8288

CoRE Link Format

RFC 6690

Home Documents for HTTP APIs

draft-nottingham-json-home-06

The Constrained RESTful Application Language (CoRAL) defines a data model and interaction model as well as two specialized serialization formats for the description of typed connections between resources on the Web (“links”), possible operations on such resources (“forms”), as well as simple resource metadata.

- Hypermedia Data & Interaction Model
- Compact, binary serialization format suitable for constrained environments
- Lightweight, textual serialization format easy to read and write for humans

“What is the resource?”

“What can you do with the resource?”

“How does the resource relate to other resources?”

Carsten's Coffee Machine SPLAT Example

```
#using <http://coreschema.org/ccm#>

isBrewing    true
progress     0.5
count        122
readyName    "size 110, strength 50%"
brewingName  "size 230, strength 100%"
maxOrders    10
create       -> POST <f/brew?create>
queue-item   <f/brew/2/> {
  state       "paused"
  cancel      -> DELETE <>
  unpause     -> POST <s/order/state> [ payload "brewing" ]
}
queue-item   <f/brew/1/> {
  state       "queued"
  cancel      -> DELETE <>
  pause       -> POST <s/order/state> [ payload "paused" ]
}
queue-item   <f/brew/0/> {
  state       "queued"
  cancel      -> DELETE <>
  pause       -> POST <s/order/state> [ payload "paused" ]
}
```

Michael's Light Switch W3C Thing Description

```
{
  "@context": [
    "http://www.w3.org/ns/thing",
    "http://iotschema.org",
    "http://www.w3.org/2011/08/jsonld"
  ],
  "@id": "http://159.203.213.90:1880",
  "@type": [
    "http://iotschema.org/Light",
    "http://iotschema.org/BinarySwitch",
    "http://iotschema.org/Level"
  ],
  "properties": {
    "switchState": {
      "@type": "http://iotschema.org/SwitchState",
      "writeproperty": true
    },
    "currentLevel": {
      "@type": "http://iotschema.org/CurrentLevel",
      "writeproperty": true
    },
    "turnOn": {
      "@type": "http://iotschema.org/TurnOn",
      "writeproperty": true
    },
    "turnOff": {
      "@type": "http://iotschema.org/TurnOff",
      "writeproperty": true
    },
    "setLevel": {
      "@type": "http://iotschema.org/SetLevel",
      "writeproperty": true
    }
  }
}
```

```
#using <http://coreapps.org/td#>
#using iot = <http://iotschema.org/>
#base <http://159.203.213.90:1880>
```

```
id "urn:uuid:2d5e84f6-85c9-4436-b53f-c0669dfd1603"
type <http://iotschema.org/Light>
type <http://iotschema.org/BinarySwitch>
type <http://iotschema.org/Level>
name "Lamp"
iot:SwitchState </light> {
  contentType "application/json"
  writeproperty -> POST </light> [ contentType "application/json" ]
}
iot:CurrentLevel </light> {
  contentType "application/json"
  writeproperty -> POST </light> [ contentType "application/json" ]
}
iot:TurnOn -> POST </light> [ contentType "application/json" ]
iot:TurnOff -> POST </light> [ contentType "application/json" ]
iot:SetLevel -> POST </light> [ contentType "application/json" ]
```

IPSO Reusable Resources

```
http://coreapps.org/ipsodigital-input-state> // The current state of a digital input.
http://coreapps.org/ipsodigital-input-counter> // The cumulative value of active state detected.
http://coreapps.org/ipsodigital-input-polarity> // The polarity of a digital input as a Boolean (0 = Normal, 1 = Reversed)
http://coreapps.org/ipsodigital-input-debounce> // The debounce period in ms.
http://coreapps.org/ipsodigital-input-edge-selection> // The edge selection as an integer (1 = Falling edge, 2 = Rising edge, 3 = Both Rising and Falling edge)
http://coreapps.org/ipsodigital-input-counter-reset> // Reset the Counter value
http://coreapps.org/ipsocurrent-time> // Unix Time. A signed integer representing the number of seconds since Jan 1st, 1970 in the UTC time zone.
http://coreapps.org/ipsorfractional-time> // For shorter times of a fraction of a second (i.e. 0.23).
http://coreapps.org/ipsomin-value> // The minimum measured value along the X axis expressed in the unit defined by the "Sensor Units" resource if present.
http://coreapps.org/ipsomax-value> // The maximum measured value along the X axis expressed in the unit defined by the "Sensor Units" resource if present.
http://coreapps.org/ipsomax-y-value> // The minimum measured value along the Y axis expressed in the unit defined by the "Sensor Units" resource if present.
http://coreapps.org/ipsomax-z-value> // The maximum measured value along the Y axis expressed in the unit defined by the "Sensor Units" resource if present.
http://coreapps.org/ipsomax-x-value> // The minimum measured value along the Z axis expressed in the unit defined by the "Sensor Units" resource if present.
http://coreapps.org/ipsomax-y-value> // The maximum measured value along the Z axis expressed in the unit defined by the "Sensor Units" resource if present.
http://coreapps.org/ipsolatitude> // The decimal notation of latitude, e.g. -43.5723 (World Geodetic System 1984).
http://coreapps.org/ipsolongitude> // The decimal notation of longitude, e.g. 153.21768 (World Geodetic System 1984).
http://coreapps.org/ipsolocalization> // The accuracy of the position in meters.
http://coreapps.org/ipsovelocisty> // The velocity of the device as defined in IGP 23.032 (IAD specification. This set of values may not be available if the device is static.
http://coreapps.org/ipsotimestamp> // The timestamp of when the location measurement was performed.
http://coreapps.org/ipsomin-limits> // The minimum value that can be measured by the sensor.
http://coreapps.org/ipsomax-limits> // The maximum value that can be measured by the sensor.
http://coreapps.org/ipsodelay-duration> // The duration of the time delay.
http://coreapps.org/ipsodelay> // Audio Clip that is playable (i.e. short audio recording indicating the floor in an elevator).
http://coreapps.org/ipsotriggers> // Trigger Initiating actuation.
http://coreapps.org/ipsoduration> // The duration of the sound once trigger.
http://coreapps.org/ipsotimer-on-off-time> // The off time when On/Off control exists on.
http://coreapps.org/ipsotimer-mode> // Type of timer pattern used by the timer. 1: One-shot, 2: On-Time or Interval, 3: Time delay on pick-up, 4: Time Delay on Drop-Out, 0: disables the timer.
http://coreapps.org/ipsotext> // A string of text.
http://coreapps.org/ipsok-coordinate> // X Coordinate.
http://coreapps.org/ipsoy-coordinate> // Y Coordinate.
http://coreapps.org/ipsoclear-display> // Command to clear the display.
http://coreapps.org/ipsolcontrast> // Proportional control, integer value between 0 and 100 as a percentage.
http://coreapps.org/ipsoincrease-input-state> // Indicates an increase control action.
http://coreapps.org/ipsodecrease-input-state> // Indicates a decrease control action.
http://coreapps.org/ipsocounters> // Counts the number of times the timer output transitions from 0 to 1.
http://coreapps.org/ipsocalibration-offset> // Calibration offset value to be used to additively correct the measured value of the sensor.
http://coreapps.org/ipsocurrent-position> // Current position or desired position of a positioner actuator.
http://coreapps.org/ipsotransition-time> // The time expected to move the actuator to the new position.
http://coreapps.org/ipsoremaining-time> // The time remaining in an operation.
http://coreapps.org/ipsoup-counter> // Counts the number of times the increase control has been operated. Writing a 0 resets the counter.
http://coreapps.org/ipsodown-counter> // Counts the times the decrease control has been operated. Writing a 0 resets the counter.
http://coreapps.org/ipsodigital-state> // The current state of the timer output.
http://coreapps.org/ipsocumulative-time> // The total time in seconds that the timer input is true. Writing a 0 resets the time.
http://coreapps.org/ipsomax-coordinate> // The highest X coordinate the display supports before wrapping to the next line.
http://coreapps.org/ipsomax-y-coordinate> // The highest Y coordinate the display supports before wrapping to the next line.
http://coreapps.org/ipsomulti-state-input> // The current state of a Multi-state input or selector.
http://coreapps.org/ipsollevel> // Used to represent a level control such as audio volume.
http://coreapps.org/ipsodigital-output-state> // The current state of a digital output.
http://coreapps.org/ipsodigital-output-polarity> // The polarity of a digital output as a Boolean (False = Normal, True = Reversed)
http://coreapps.org/ipsodigital-input-current-value> // The current value of the analog input expressed in the unit defined by the "Sensor Units" resource if present.
http://coreapps.org/ipsomin-measured-value> // The minimum value measured by the sensor since it is ON or Reset, expressed in the unit defined by the "Sensor Units" resource if present.
http://coreapps.org/ipsomax-measured-value> // The maximum value measured by the sensor since it is ON or Reset, expressed in the unit defined by the "Sensor Units" resource if present.
http://coreapps.org/ipsomin-range-value> // The minimum value that can be measured by the sensor, expressed in the unit defined by the "Sensor Units" resource if present.
http://coreapps.org/ipsomax-range-value> // The maximum value that can be measured by the sensor, expressed in the unit defined by the "Sensor Units" resource if present.
http://coreapps.org/ipsolreset-min-max-measured-values> // Reset the Min and Max Measured Values to current value.
http://coreapps.org/ipsolanalogue-output-current-value> // The current value of the analog output.
http://coreapps.org/ipsolsensor-value> // Last or Current Measured Value from the Sensor expressed in the unit defined by the "Sensor Units" resource if present.
http://coreapps.org/ipsolmeasurement-units> // Measurement Units Definition e.g. "Cel" for Temperature in degrees Celsius.
http://coreapps.org/ipsolx-value> // The measured value along the X axis expressed in the unit defined by the "Sensor Units" resource if present.
http://coreapps.org/ipsoly-value> // The measured value along the Y axis expressed in the unit defined by the "Sensor Units" resource if present.
http://coreapps.org/ipsolz-value> // The measured value along the Z axis expressed in the unit defined by the "Sensor Units" resource if present.
http://coreapps.org/ipsolcompass-direction> // The compass direction.
http://coreapps.org/ipsolcolour> // A string representing a value in the color space defined by the "Sensor Units" resource if present.
http://coreapps.org/ipsolapplication-type> // The application type of the sensor or actuator as a string, for instance "Air Pressure".
http://coreapps.org/ipsolsensor-type> // The type of the sensor (for instance PIR type)
http://coreapps.org/ipsolinstantaneous-active-power> // The current active power
http://coreapps.org/ipsolmin-measured-active-power> // The minimum active power measured by the sensor since it is ON
http://coreapps.org/ipsolmax-measured-active-power> // The maximum active power measured by the sensor since it is ON
http://coreapps.org/ipsolmin-range-active-power> // The minimum active power that can be measured by the sensor
http://coreapps.org/ipsolmax-range-active-power> // The maximum active power that can be measured by the sensor
http://coreapps.org/ipsolcumulative-active-power> // The cumulative active power since the last cumulative energy reset or device start
http://coreapps.org/ipsolactive-power-calibration> // Request an active power calibration by writing the value of a calibrated load.
http://coreapps.org/ipsolinstantaneous-reactive-power> // The current reactive power
http://coreapps.org/ipsolmin-measured-reactive-power> // The minimum reactive power measured by the sensor since it is ON
http://coreapps.org/ipsolmax-measured-reactive-power> // The maximum reactive power measured by the sensor since it is ON
http://coreapps.org/ipsolmin-range-reactive-power> // The minimum active power that can be measured by the sensor
http://coreapps.org/ipsolmax-range-reactive-power> // The maximum reactive power that can be measured by the sensor
http://coreapps.org/ipsolcumulative-reactive-power> // The cumulative reactive power since the last cumulative energy reset or device start
http://coreapps.org/ipsolreactive-power-calibration> // Request a reactive power calibration by writing the value of a calibrated load.
http://coreapps.org/ipsolpower-factor> // If applicable, the power factor of the current consumption.
http://coreapps.org/ipsolcurrent-calibration> // Read or Write the current calibration coefficient
http://coreapps.org/ipsolreset-cumulative-energy> // Reset both cumulative active/reactive power
http://coreapps.org/ipsolevent-identifier> // The event identifier as a string.
http://coreapps.org/ipsolstart-time> // Time when the load control event will start started.
http://coreapps.org/ipsolduration-in-min> // The duration of the load control event.
http://coreapps.org/ipsolcriticality-level> // The criticality of the event. The device receiving the event will react in an appropriate fashion for the device.
http://coreapps.org/ipsolavg-load-adjpct> // Defines the maximum energy usage of the receiving device, as a percentage of the device's normal maximum energy usage.
http://coreapps.org/ipsolduty-cycle> // Defines the duty cycle for the load control event, i.e. what percentage of time the receiving device is allowed to be on.
http://coreapps.org/ipsolon-off> // This resource represents an on/off actuator, which can be controlled, the setting of which is a Boolean value where True is On and False is Off.
http://coreapps.org/ipsoldimmer> // This resource represents a light dimmer setting, which has an Integer value between 0 and 100 as a percentage.
http://coreapps.org/ipsolon-time> // The time in seconds that the device has been turned on. Writing a value of 0 resets the counter.
http://coreapps.org/ipsomulti-state-output> // A string describing a state for multiple level output such as Pilot Wire
http://coreapps.org/ipsoloff-time> // The time in seconds the Off command was sent. Writing a value of 0 resets the counter.
http://coreapps.org/ipsolset-point-value> // The setpoint value expressed in the unit defined by the "Sensor Units" resource if present..
http://coreapps.org/ipsolbusy-to-clear-delay> // Delay from the detection state to the clear state in ms
http://coreapps.org/ipsolclear-to-busy-delay> // Delay from the clear state to the busy state in ms.
http://coreapps.org/ipsolbitmap-inputs> // Integer in which each of the bits are associated with specific digital input value. Represented as a binary signed integer in network bytes order, and in two's complement representation.
http://coreapps.org/ipsolbitmap-inputs-reset> // Reset the Bitmap
http://coreapps.org/ipsolbitmap-description> // The semantics / description of each bit as a string. First instance describes the least significant bit, second instance the second least significant bit, etc
```

.well-known/ core Link Format

```
#using <http://coreapps.org/core#>
#using iana = <http://www.iana.org/assignments/relation/>

rd-item </sensors> {
  ct 40
  title "Sensor Index"
}
rd-item </sensors/temp> {
  rt "temperature-c"
  if "sensor"
  iana:describedby <http://www.example.com/sensors/t123>
  iana:alternate </t>
}
rd-item </sensors/light> {
  rt "light-lux"
  if "sensor"
}
```

<https://datatracker.ietf.org/doc/draft-hartke-t2trg-coral/>

<https://datatracker.ietf.org/doc/draft-hartke-t2trg-ciri/>

<https://datatracker.ietf.org/doc/draft-hartke-t2trg-coral-reef/>

<https://datatracker.ietf.org/doc/draft-hartke-t2trg-data-hub/>

<https://github.com/ektrah/coral>

Photo credits:

“Morning in the anemone forest” by FotoFloridian

<https://flic.kr/p/W2HdTS> (CC BY-NC 2.0)