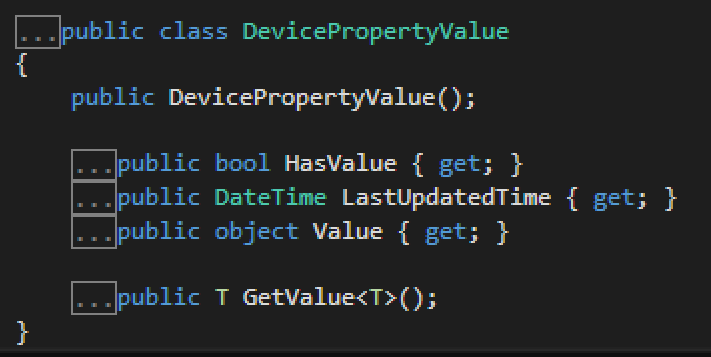
Azure IoT Hub Properties

# Device System Properties

(off of the Device object)

* Data coming from the IoT Hub device registry database

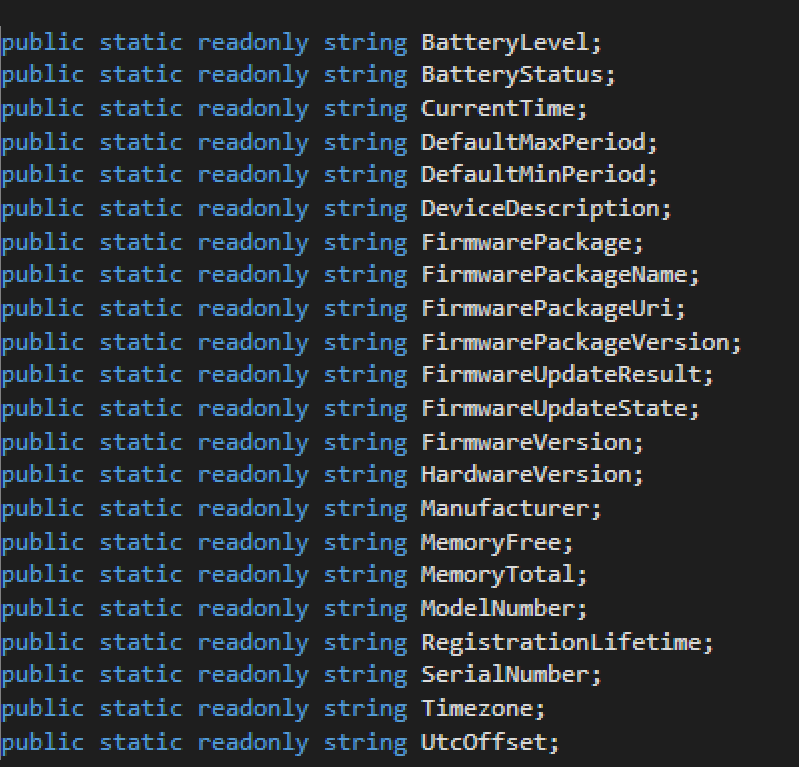
public IReadOnlyDictionary<string, DevicePropertyValue> SystemProperties { get; }



\*\*\* Notice that these are read only (through the getter on Value property)

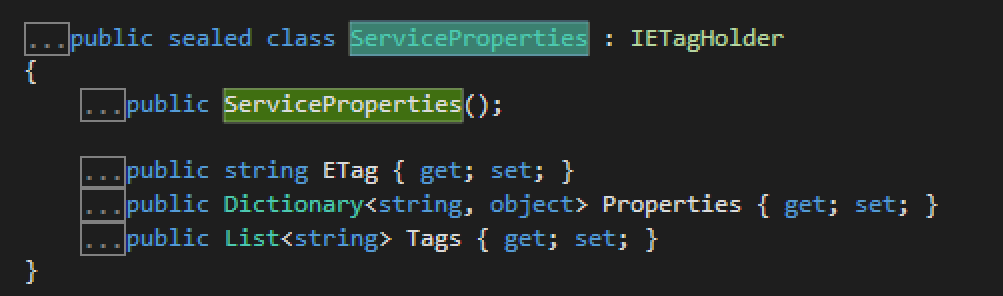
## System Property Names

Not customizable in the public preview timeframe. For GA, building a Device Models API that will enable customization.



# Service Properties

public ServiceProperties ServiceProperties { get; }



\*\*\* Notice that service properties are read/write from the service API.

Forcing a read from the device and updating the value in the IoT Hub device registry:

public abstract Task<JobResponse> ScheduleSystemPropertyReadAsync(string jobId, string deviceId, string propertyName);

Setting a system property from the service is done through a ScheduleSystemPropertyWriteAsync call, which starts a job.

public abstract Task<JobResponse> ScheduleSystemPropertyWriteAsync(string jobId, string deviceId, string propertyName, object value);

# Update Frequency

Based on the LWM2M/CoAP Observe

Min period –Debouncing- Limit device from sending data more than the period defimed by minPeriod. E.g. 5 mins; over time- change=send; elapse=3 mins; change=don’t send; elapse=3 mins; change=send;elapse=10 mins; change=send. Throttling from the device side

Max period – Stay in sync with the device property values. E.g. 6 hours. “Send me the value of the property even if it hasn’t changed”

# Device Reconnecting

On registration, the device automatically updates the state of the device registry.

# Brainstorming Scenarios

1. History of device state
   1. Post Public Preview – Pushing state changes into the eventHub.
2. High level location (Cars in Seattle, Bellevue, Redmond)
3. Desired device state
   1. Using the device job to track current state vs jobs that will update the device and keep track of pending desired device state.
   2. Track different properties on each device that have pending desired state.
4. Multiple mobile users updating a single device
   1. Web service for the mobile apps that in turn has permissions to send job requests to the device. Access control done at the web service.