# Device Class

public class Device : IETagHolder

{

public Device();

public Device(string id);

public AuthenticationMechanism Authentication { get; set; }

public int CloudToDeviceMessageCount { get; internal set; }

//

// Summary:

// Configuration value that is sent to the device

public string Configuration { get; internal set; }

public DeviceConnectionState ConnectionState { get; internal set; }

public DateTime ConnectionStateUpdatedTime { get; internal set; }

//

// Summary:

// Custom properties that are defined in the device model. It also contains

// Auto Generated properties for the Device

public IReadOnlyDictionary<string, DevicePropertyValue> CustomProperties { get; internal set; }

public string ETag { get; set; }

public string GenerationId { get; internal set; }

public string Id { get; internal set; }

public bool IsManaged { get; internal set; }

public DateTime LastActivityTime { get; internal set; }

//

// Summary:

// Properties that are mastered by the Service

public ServiceProperties ServiceProperties { get; internal set; }

public DeviceStatus Status { get; set; }

public string StatusReason { get; set; }

public DateTime StatusUpdatedTime { get; internal set; }

//

// Summary:

// Device Properties that are defined in the System Model. These are common

// for all the devices.

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

// Summary:

// Returns the Property value if available from CustomProperties Exceptions:

// KeyNotFoundException if property is not found InvalidOperationException if

// property does not have a value InvalidCastException if the Cast is not valid

//

// Parameters:

// propertyName:

//

// Type parameters:

// T:

public T GetCustomPropertyValue<T>(string propertyName);

//

// Summary:

// Returns the Property value if available from SystemProperties Exceptions:

// KeyNotFoundException if property is not found InvalidOperationException if

// property does not have a value InvalidCastException if the Cast is not valid

//

// Parameters:

// propertyName:

//

// Type parameters:

// T:

public T GetSystemPropertyValue<T>(string propertyName);

}

// Summary:

// Properties that are managed by the Service. The Contents cannot be changed

// once created

public sealed class ServiceProperties : IETagHolder

{

public ServiceProperties();

public string ETag { get; set; }

//

// Summary:

// Custom list of Properties.

public Dictionary<string, object> Properties { get; set; }

//

// Summary:

// Tags for the Device

public List<string> Tags { get; set; }

}

// Summary:

// Defines the Predefined System Property Names that can be looked up in the

// SystemProperties.

public static class SystemPropertyNames

{

// Summary:

// BatteryLevel (int)

public static readonly string BatteryLevel;

//

// Summary:

// BatteryStatus (String)

public static readonly string BatteryStatus;

//

// Summary:

// CurrentTime (DateTime)

public static readonly string CurrentTime;

//

// Summary:

// DefaultMaxPeriod (int)

public static readonly string DefaultMaxPeriod;

//

// Summary:

// DefaultMinPeriod (int)

public static readonly string DefaultMinPeriod;

//

// Summary:

// DeviceDescription (String)

public static readonly string DeviceDescription;

//

// Summary:

// FirmwarePackage (String)

public static readonly string FirmwarePackage;

//

// Summary:

// FirmwarePackageName (String)

public static readonly string FirmwarePackageName;

//

// Summary:

// FirmwarePackageUri (String)

public static readonly string FirmwarePackageUri;

//

// Summary:

// FirmwarePackageVersion (String)

public static readonly string FirmwarePackageVersion;

//

// Summary:

// FirmwareUpdateResult (int)

public static readonly string FirmwareUpdateResult;

//

// Summary:

// FirmwareUpdateState (String)

public static readonly string FirmwareUpdateState;

//

// Summary:

// FirmwareVersion (String)

public static readonly string FirmwareVersion;

//

// Summary:

// HardwareVersion (String)

public static readonly string HardwareVersion;

//

// Summary:

// Manufacturer (String)

public static readonly string Manufacturer;

//

// Summary:

// MemoryFree (int)

public static readonly string MemoryFree;

//

// Summary:

// MemoryTotal (int)

public static readonly string MemoryTotal;

//

// Summary:

// ModelNumber (String)

public static readonly string ModelNumber;

//

// Summary:

// RegistrationLifetime (int)

public static readonly string RegistrationLifetime;

//

// Summary:

// SerialNumber (String)

public static readonly string SerialNumber;

//

// Summary:

// Timezone (String)

public static readonly string Timezone;

//

// Summary:

// UtcOffset (String)

public static readonly string UtcOffset;

}

# Query API

//

// Summary:

// Retrieves specified number of devices matching supplied tags. Results are

// not ordered.

//

// Parameters:

// tags:

// Tags to match on devices

//

// maxCount:

// Maximum number of devices to return

//

// Returns:

// The list of matching devices

public abstract Task<System.Collections.Generic.IEnumerable<Device>> QueryDevicesAsync(string[] tags, int maxCount);

# Jobs API

//

// Summary:

// Enum for the status of a JobClient job

//

public enum JobStatus

{

Unknown = 0,

Enqueued = 1,

Running = 2,

Completed = 3,

Failed = 4,

Cancelled = 5,

}

// Summary:

// Provides current job report when fetched

public class JobResponse

{

public JobResponse();

// Summary:

// System generated. Ignored at creation. Represents the time the job stopped

// processing.

public DateTime? EndTimeUtc { get; internal set; }

//

// Summary:

// System generated. Ignored at creation. If status == failure, this

// represents a string containing the reason.

public string FailureReason { get; set; }

//

// Summary:

// System generated. Ignored at creation.

public string JobId { get; internal set; }

//

// Summary:

// System generated. Ignored at creation. Represents the percentage of completion.

public int Progress { get; internal set; }

//

// Summary:

// System generated. Ignored at creation.

public DateTime? StartTimeUtc { get; internal set; }

//

// Summary:

// System generated. Ignored at creation.

public JobStatus Status { get; internal set; }

//

// Summary:

// Required. The type of job to execute.

public JobType Type { get; internal set; }

}

//

// Summary:

// Creates an JobClient from the Iot Hub connection string.

//

// Parameters:

// connectionString:

// The Iot Hub connection string.

//

// Returns:

// An JobClient instance.

public static JobClient CreateFromConnectionString(string connectionString);

//

// Summary:

// Starts a job to sets the firmware package URI and begin the update per LWM2M spec

//

// Parameters:

// jobClient:

// The JobClient instance used to create the job.

// jobId:

// A unique identifier for the job. Typically a GUID.

// deviceId:

// The id of the device for which you want to set the URI.

// value:

// The URI for the firmware package.

//

// Returns:

// An JobClient instance.

public static Task<JobResponse> SetFirmwarePackageUriAsync(this JobClient jobClient, string jobId, string deviceId, string value);

//

// Summary:

// Gets the job with the specified ID.

//

// Parameters:

// jobId:

public abstract Task<JobResponse> GetJobAsync(string jobId);