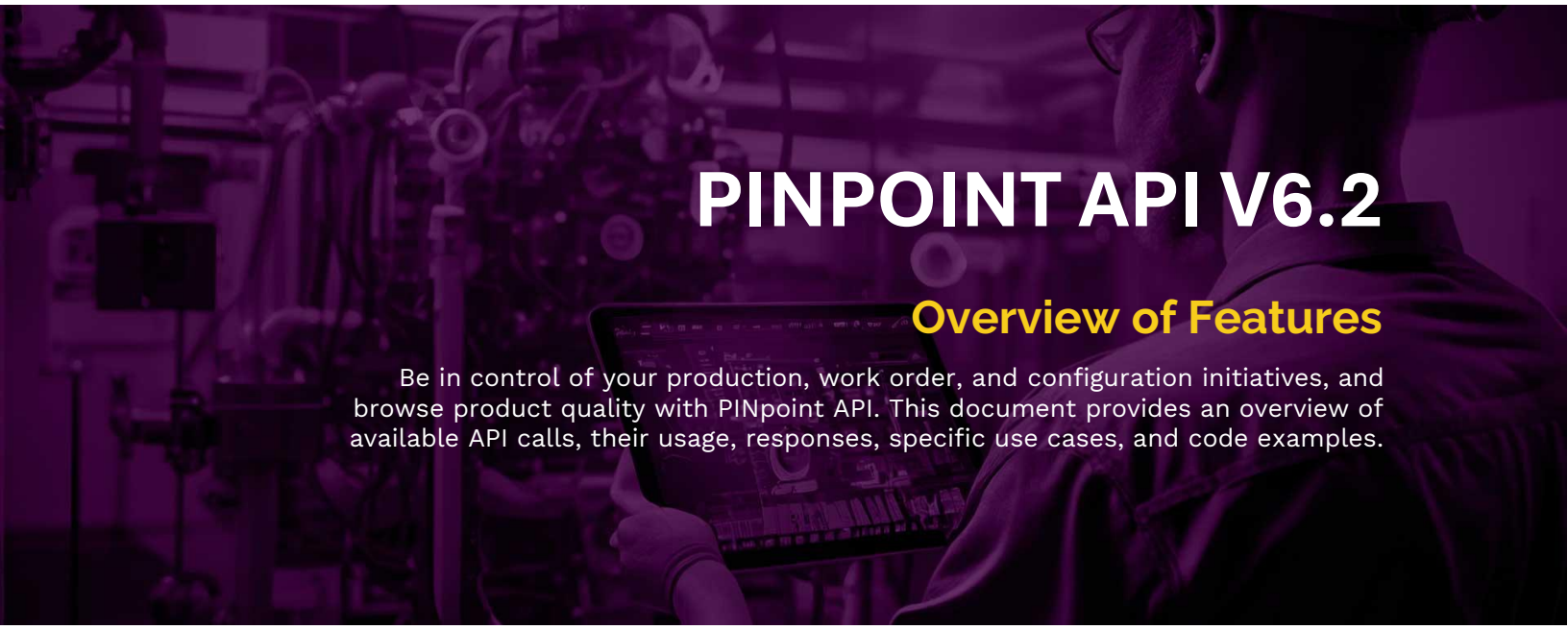


PINPOINT API V6.2

Overview of Features

Be in control of your production, work order, and configuration initiatives, and browse product quality with PINpoint API. This document provides an overview of available API calls, their usage, responses, specific use cases, and code examples.

PINpoint Information Systems Inc.
techsupport@pinpointinfo.com, sales@pinpointinfo.com

Table of Contents

Table of Contents 2

Overview 4

- Server Technology 4
- Security 4
- Common Properties 8
- Common HTTP Return Codes 9
- Validation 10
- Rate Limiting 11
- Endpoint Deprecation 11

Configuration 12

- Configuration/ValidateParts **POST** 12
- Configuration/ValidateStationParts **POST** 13

Production 15

- Production/LoadSerialNumber **POST** 15
- Production/UnloadSerialNumber **POST** 17
- Production/EnablePart **POST** 18
- Production/EnableProcess **POST** 19
- Production/DisablePart **POST** 20
- Production/DisableProcess **POST** 21
- Production/SubmitWorkOrder **POST** 22
- Production/WorkOrders **DELETE** 27
- Production/TesterResults **POST** 28
- Production/OperationValues **GET** 30

Production/Quality 31

- Production/Quality/Product **GET** 31
- Production/Quality/ProductDetail **GET** 41

Production/Quality/ProductionAdvisory 46

- Production/Quality/ProductionAdvisory **GET** 46
- Production/Quality/ProductionAdvisory/{productionAdvisoryId} **GET** 47
- Production/Quality/ProductionAdvisory **POST** 48
- Production/Quality/ProductionAdvisory **PATCH** 49

Production/Quality/Issues 50

- Production/Quality/Issues **GET** 50
- Production/Quality/Issues/{defectId} **GET** 51
- Production/Quality/Issues **POST** 52
- Production/Quality/Issues **PATCH** 54

Production/Quality/Issues/{defectId}/Copy **POST** 55
Production/Quality/Issues/Status **POST** 56
Production/Quality/Issues/Note **POST** 57
Superseded Endpoints 58
Configuration/ValidateProducts **POST** (Superseded) 58
Configuration/ValidateStationProducts **POST** (Superseded) 58
Production/EnableProduct **POST** (Superseded) 58
Production/DisableProduct **POST** (Superseded) 58
Production/DeleteWorkOrder **POST** (Superseded) 58
Integration Endpoints 59
Support 60

Overview

The PINpoint API is a customer facing API that resides in PINpoint's Web Service which exposes certain abilities to control production and configuration related tasks, work order submissions, and to browse and interact with product quality. The PINpoint API is broken down into functional areas pertaining to the tasks that may be performed.

The API methods are considered RPC style unless otherwise specified and accept JSON request body contents. **XML requests are deprecated in the PINpoint API.** HTTP headers describing the content type and expected content response type should be provided. An access token is to be present for each call and is retrievable via PINpoint Security Server (more details in the security section).

Server Technology

- ASP.NET Web API
- JSON serialized payloads
- HTTP/HTTPS (**HTTPS recommended**)

Security

The API will be available only on the customer's intranet. There may be any number of machines that will call the API, and they can come from any domain that has access to the web server hosting the API.

Each call to the API must include an OAuth 2.0 access token that is to be retrieved from the PINpoint Security Server with the correct credentials and scopes to ensure that the call is allowed. The access token will be attached to each request as a bearer token. These tokens are revocable via a limited timeframe in which they can be used. Once revoked, or once expired, a new access token must be requested.

A PINpoint team member will provide the client ID, client secret, and required scopes to retrieve an access token from the PINpoint Security Server for use in PINpoint API.

Below is an example of how one can be retrieved. PINpoint can also provide a sample .NET 6/7/8 console application to provide further clarity.

```
/// <summary>
/// Retrieves an OAuth 2.0 access token for use with calls to the PINpoint API.
/// </summary>
/// <param name="cancellationToken">The cancellation token.</param>
/// <returns>An access token response.</returns>
1 reference
private static async Task<AccessTokenResponse?> RetrieveAccessTokenAsync(string clientId, string clientSecret, CancellationToken cancellationToken)
{
    AccessTokenResponse? accessTokenResponse = null;

    try
    {
        var accessTokenUrl = "https://localhost:44396/PinPoint.SecurityServer/connect/token";

        using HttpClient httpClient = new();
        using FormUrlEncodedContent content = new(new List<KeyValuePair<string, string>>(4)
        {
            new KeyValuePair<string, string>("grant_type", "client_credentials"),
            new KeyValuePair<string, string>("scope", "external_api"),
            new KeyValuePair<string, string>("client_id", clientId),
            new KeyValuePair<string, string>("client_secret", clientSecret)
        });
        content.Headers.Clear();
        content.Headers.Add("Content-Type", "application/x-www-form-urlencoded");

        HttpResponseMessage accessTokenHttpResponse = await httpClient.PostAsync(accessTokenUrl, content, cancellationToken);

        if (accessTokenHttpResponse?.IsSuccessStatusCode == true)
        {
            Stream responseStream = await accessTokenHttpResponse.Content.ReadAsStreamAsync(cancellationToken);
            accessTokenResponse = await JsonSerializer.DeserializeAsync<AccessTokenResponse>(responseStream, cancellationToken: cancellationToken);
        }
    }
    catch (Exception exception)
    {
        Console.WriteLine(exception.ToString());
    }

    return accessTokenResponse;
}
```

This example comes from the available .NET sample application using standard libraries available in the .NET framework. The **AccessTokenResponse** class is an abbreviated construct that contains standard OAuth 2.0 fields as shown here:

```
/// <summary>
/// Encapsulates the access token response from the PINpoint Security Server.
/// </summary>
/// <remarks>
/// https://www.oauth.com/oauth2-servers/access-tokens/access-token-response/
/// RFC 6750 part 4: https://www.rfc-editor.org/rfc/rfc6750
/// </remarks>
7 references
internal sealed class AccessTokenResponse
{
#pragma warning disable IDE1006 // Naming Styles
    /// <summary>
    /// The access token.
    /// </summary>
    3 references
    public string? access_token { get; set; }

    /// <summary>
    /// The token type.
    /// </summary>
    2 references
    public string? token_type { get; set; }

    /// <summary>
    /// Seconds
    /// </summary>
    2 references
    public int? expires_in { get; set; }
#pragma warning restore IDE1006 // Naming Styles

    /// <summary>
    /// Dump contents.
    /// </summary>
    1 reference
    public override string ToString()
    {
        return $"{nameof(access_token)}: '{access_token}', {nameof(token_type)}: '{token_type}', {nameof(expires_in)}: '{expires_in}'";
    }
}
```

The available fields can be observed in the JSON response from the security server during an authentication request. These references below outline the applicable fields and the related RFC:

- https://www.oauth.com/oauth2-servers/access-tokens/access-token-response/
- https://www.rfc-editor.org/rfc/rfc6750

Apply the received access token as a bearer token header for subsequent calls to PINpoint API:

```
/// <summary>
/// Performs the request to PINpoint API.
/// </summary>
/// <typeparam name="T">The object type.</typeparam>
/// <param name="accessTokenResponse">The retrieved access token.</param>
/// <param name="targetUrl">The target URL.</param>
/// <param name="value">The request contents.</param>
/// <param name="cancellationToken">The cancellation token.</param>
/// <returns>Boolean indicating success.</returns>
2 references
private static async Task<bool> MakeHttpRequest<T>(AccessTokenResponse? accessTokenResponse, string targetUrl, T value, CancellationToken cancellationToken)
{
    var isSuccessful = false;

    try
    {
        using HttpClient httpClient = new();
        httpClient.DefaultRequestHeaders.Accept.Add(new MediaTypeWithQualityHeaderValue("application/json"));
        httpClient.DefaultRequestHeaders.TryAddWithoutValidation("Authorization", $"Bearer {accessTokenResponse!.access_token}");

        HttpResponseMessage httpResponseMessage = await httpClient.PostAsJsonAsync(targetUrl, value, cancellationToken);

        isSuccessful = await VerifyResponse(httpResponseMessage, cancellationToken);
    }
    catch (Exception exception)
    {
        Console.WriteLine(exception.ToString());
    }

    return isSuccessful;
}

/// <summary>
/// Verifies that the response was successful, otherwise optionally prints the errors.
/// </summary>
/// <param name="httpResponseMessage">The response message.</param>
/// <param name="cancellationToken">The cancellation token.</param>
/// <returns>Boolean indicating success.</returns>
1 reference
private static async Task<bool> VerifyResponse(HttpResponseMessage httpResponseMessage, CancellationToken cancellationToken)
{
    var isSuccessful = false;

    try
    {
        if (httpResponseMessage?.IsSuccessStatusCode == true)
            isSuccessful = true;
        else
        {
            isSuccessful = false;

            // Optionally, we can capture the result and interrogate validation errors.
            //Stream responseStream = await httpResponseMessage!.Content.ReadAsStreamAsync(cancellationToken);
            //_ = await JsonSerializer.DeserializeAsync<SubmitWorkOrder>(responseStream, cancellationToken: cancellationToken);

            var responseString = await httpResponseMessage!.Content.ReadAsStringAsync(cancellationToken);
            Console.Write(responseString);
        }
    }
    catch (Exception exception)
    {
        Console.WriteLine(exception.ToString());
    }

    return isSuccessful;
}
```

Common Properties

These properties apply to all endpoints.

- **URL format**: `[full path to PINpoint web service]/api/v1/External/[endpoint path]`. For example, the full URL for `Production/LoadSerialNumber` would be `http(s)://YourServerName/PinPoint.WebService/api/v1/External/Production/LoadSerialNumber`
- A **Content-Type** of **application/json**
- An **Accept** of **application/json**
- An OAuth 2.0 bearer token retrieved from the PINpoint Security Server.
- **Path segments (or path parameters)** are parts of the URL path, separated by slashes (/), used to identify specific resources. For example, in …`/api/v1/issues/5`, the `5` is a path segment that identifies a particular issue (by its DefectId). Path parameters are required and form part of the resource's unique address.
- **Query parameters** are key-value pairs appended to the end of a URL after a question mark (?), typically used to filter, sort, or paginate results. For example, `…/External/Production/Quality/ProductionAdvisory?Type=Shortage` uses `Type=Shortage` as a query parameter to filter issues. The order in which the query string parameters are provided does not matter. Spaces and special characters may be used in query string parameters. Query parameters are optional and do not identify specific resources.
- **JSON body (request body)** is the data sent within the body of an HTTP request, usually in JSON format, to provide additional information needed to create or update resources. For example, when creating a new issue, a POST request might include a JSON body like `{"title": "Bug report", "description": "Details about the bug"}`. The request body is not part of the URL and is typically used for more complex or structured data.
- JSON property identifiers are case insensitive. For example, `"OrderNumber": null` could be provided as `"ORDERNUMBER": null`, `"ordernumber": null`, etc…
- A successful call usually echoes the request data, with some more information specific to the endpoint.
- An unsuccessful call usually echoes the request data with validation errors and an HTTP error code.

Common HTTP Return Codes

These HTTP return codes apply to all endpoints. Each endpoint might have return codes or meanings to return codes that are unique to itself which are specified in their respective sections.

Code	**Reason**
200 OK	The request was successfully processed.
400 Bad Request	The request contains syntactical errors, a mandatory parameter was not provided, a namespace or property was misspelled, or a parameter has exceeded the allowable size limit.
401 Unauthorized	The required access token credentials are not provided, or they are invalid.
403 Forbidden	The required credentials were provided but they do not have access to the API call or resource. They may be missing the correct scopes.
422 Unprocessable Entity	The request was received successfully, but the server was unable to fulfill it. The response may include validation errors in line with the properties that caused them.
500 Internal Server Error	An unexpected error has occurred. Please attempt the request again. If further responses are returned with this error code, please contact a PINpoint representative. Error messages will be returned in the response body.
503 Service Unavailable	The server is not ready to handle the request. The server may be down for maintenance or is currently overloaded. Please try the request again later. A Pp-Api-EstimatedAvailabilityIn header may be supplied in the response denoting the estimated wait time.

Validation

When a request is received, it undergoes validation against a predefined set of rules. If any rules are violated, the specific broken rules are returned as part of the response payload. These validation errors are included as child elements within the response data structure.

Responses containing validation errors will use the HTTP status code **422 Unprocessable Entity** (or sometimes **400 Bad Request**) to clearly indicate that the request was well-formed but contains semantic errors or invalid parameters that prevent processing.

Each error entry in the response includes:

- **Error:** Describes the nature of the validation failure. The specific error types applicable to each endpoint are detailed in their respective sections.
- **Property:** A string identifying the property in the request data that caused the validation error.

This structure allows clients to easily identify which fields require correction and the reason for each validation failure.

Common Validation Errors:

Error	**Reason**
Required	A required property is missing from the request.
Unknown/NotFound	The specified resource could not be located.
Invalid	The provided data is invalid or does not meet the expected format.
Ambiguous	The request found multiple possible matches. Provide additional data to narrow down the targeted entity.
InvalidFormat	The value doesn't meet the configured format rules.
Duplicate	A resource with the same unique identifier or attributes already exists.
Unsupported	The requested operation is not supported by the API.

Example Error Response Structure:

```
{
   "Parts": [
      {
         "PartNumber": "KL933352",
         "ValidationErrors": [
            {
               "Error": "Unknown",
               "Property": "PartNumber"
            }
         ]
      },
      {
         "PartNumber": "ZZ1134AB",
         "ValidationErrors": [
            {
               "Error": "Unknown",
               "Property": "PartNumber"
            }
         ]
      }
   ]
}
```

This indicates that part numbers KL933352 and ZZ1134AB are unknown in the PinPoint database.

Refer to each endpoint's documentation for a list of possible error types and their meanings.

Rate Limiting

Depending on the functional area, requests are served on a first-come-first-serve basis. The default rate limit for every endpoint is **120 requests per minute**. This is reflected in the **appsettings.json** file.

For example:

```
{
	"Endpoint": "*External/Production/LoadSerialNumber",
	"Period": "1m",
	"Limit": 120
}
```

To change the limit of this endpoint to **5 requests per second**:

```
{
	"Endpoint": "*External/Production/LoadSerialNumber",
	"Period": "1s",
	"Limit": 5
}
```

You can configure multiple rate limits for the same endpoint simultaneously, each using a different time window. The Period string supports "1s" (1 second), "1m" (1 minute), "1h" (1 hour), and "1d" (1 day).

Endpoint Deprecation

Starting with version 6.0, certain API endpoints have been deprecated and should not be used to develop new integrations. However, superseded endpoints remain available to ensure backward compatibility. When a call is made to a superseded endpoint, it will be automatically redirected to the corresponding new endpoint without requiring any changes to your existing JSON payloads. For instance, if your payload contains a "products" field, it will be converted to "parts" during the redirection process.

Endpoint/Path	**Superseded By**
Superseded Endpoints	
Configuration/ValidateProducts	ValidateParts
Configuration/ValidateStationProducts	ValidateStationParts
Production/EnableProduct	EnablePart
Production/DisableProduct	DisablePart
Production/DeleteWorkOrder	WorkOrders

Configuration

Configuration/ValidateParts **POST**

Ensures a collection of parts are configured.

CALL DETAILS

- The pre-v6 version of this endpoint, *ValidateProducts*, is expected to be deprecated in the future but is currently still supported and will be redirected to this endpoint. The JSON body doesn't require any change, meaning there's no need to change *Products* to *Parts*. A call that would succeed in a pre-v6 version should also succeed in v6.0.

BUSINESS RULES

- A part is identified by a unique part number.

MODEL HIERARCHY

Parts *(Part[])*

MODEL HIERARCHY IN DETAIL

PART

- *PartNumber* (string (255), **required**)

SAMPLE

```
{
   "Parts": [
      {
         "PartNumber": "KL933352"
      },
      {
         "PartNumber": "ZZ1134AB"
      },
      {
         "PartNumber": "JK911232"
      }
   ]
}
```

This will validate and ensure part numbers KL933352, ZZ1134AB, and JK911232 are configured.

Configuration/ValidateStationParts **POST**

Ensures a collection of parts are configured and that ordering those parts will result in a process being selected at the station.

CALL DETAILS

- The pre-v6 version of this endpoint, *ValidateStationProducts*, is expected to be deprecated in the future but is currently still supported and will be redirected to this endpoint. The JSON body doesn't require any change, meaning there's no need to change *Products* to *Parts*. A call that would succeed in a pre-v6 version should also succeed in v6.0.

BUSINESS RULES

- A Part is identified by a unique Part Number.

MODEL HIERARCHY

StationParts *(StationPart[])*

Parts *(Part[])*

MODEL HIERARCHY IN DETAIL

STATION PART

- *Station* (string (255), **required**)

PART

- *PartNumber* (string (255), **required**)

SAMPLE

```
{
   "StationParts": [
      {
         "Station": "Start Station 1",
         "Parts": [
            {
               "PartNumber": "KL933352"
            },
            {
               "PartNumber": "ZZ1134AB"
            },
            {
               "PartNumber": "JK911232"
            }
         ]
      },
      {
         "Station": "Start Station 2",
         "Parts": [
            {
               "PartNumber": "KL933352"
            },
            {
               "PartNumber": "ZZ1134AB"
            },
            {
               "PartNumber": "JK911232"
            }
         ]
      }
   ]
}
```

This will validate and ensure part numbers KL933352, ZZ1134AB, and JK911232 are configured, and a process will be selected at Start Station 1 and Start Station 2.

VALIDATION

Error	**Reason**
InvalidFormat	The value doesn't meet the configured format rules.
Required	A value required to perform the request was not provided.
Unknown	The station or part number could not be found and are not configured.
NotAtStation	The part number is not configured at its respective station and ordering those parts will not result in a process being selected at the station.

Production

Production/LoadSerialNumber **POST**

Loads a serial number into a station.

BUSINESS RULES

- If a station contains a pending or empty item, the item will be reclassified (replaced) as the loaded serial number.
- If a different serial number is currently loaded at the station, that serial number will be unloaded and placed into the next queue, and the requested serial number will be loaded.
- If the serial number belongs to a work order that has not been released to production, the work order will be released to production. Once released, the serial numbers contained in the work order will become queued to their respective routes. Once queued, the requested serial number will be loaded into the station.
- If the serial number belongs to a route that is inactive or unloaded, an InactiveOrUnloaded Route validation error will be returned.
- Serial numbers may not be loaded into peer stations.
- Depending on SmartBuild setting, **Serial Number Unique Per Serial Number Controller**, serial numbers may be reused across multiple routes. Regardless of this setting, serial numbers may be reused for the same part on a route if the existing serial number has been rejected.
- If multiple eligible serial numbers are found to be loaded at the station, there is an eligibility order that determines which to load:

    - First, the serial number that is ready for work,
    - If none found, the serial number that is started,
    - If none found, the serial number that has not been worked on.
    - If none found, the serial number that is rejected,
    - Finally, the serial number that is completed.
        - If the same serial numbers have the same status, the serial number created first will be chosen.

MODEL HIERARCHY

- *SerialNumber* (string (128), **required**)
- *Station* (string (255), **required**)
- *PartNumber* (string (255), **optional**)
    - top level part number used to handle proper serial number selection when multiple eligible serial numbers are found on a single route
- *Route* (string (255), **optional**)
    - used to handle proper serial number selection when multiple eligible serial numbers are found across routes

The top-level part number and route should be provided in the request, when applicable, to avoid validation errors in case they are required.

In simple configuration setups, for example when unique serial numbers and unique stations are used in routes, the serial number and station are enough parameters to correctly determine the serial number to be loaded. The part number and/or route are ignored in these cases.

For example, if the same serial number is used for multiple part numbers on a single route, a part number will be required in the request. If the same serial number is used on different routes, the route will be required in the request. If the same serial number is used for multiple part numbers, across multiple routes, both the part number and route are required in the request.

SAMPLE

```
{
   "SerialNumber": "710111",
   "PartNumber": "KL933352",
   "Route": "Main Line",
   "Station": "Start Station 1"
}
```

This will attempt to load serial number 710111 into Start Station 1. Depending on the complexity of how serial numbers are reused per route or across routes, the route and part number (top level) may be required. PINpoint will inform the consumer when more data is needed to ensure the correct serial number is chosen to load at the desired station.

VALIDATION

Error	**Reason**
InvalidFormat	The value doesn't meet the configured format rules.
Unknown	The serial number, station, or route could not be found.
Required	A value required to perform the request was not provided.
AlreadyAtStation	The same serial number currently exists at the intended station.
Invalid	The request is attempted for stations that do not allow it (peer stations for example).
InternalTrackingError	An unexpected response was returned during underlying communication among components.
Ambiguous	The serial number and station combination exist among multiple routes. More data may be required in the request. PINpoint will inform the consumer of what additional data is required. Including a part number (top level) and route is recommended to ensure the correct serial number is chosen for the intended station in a single request.
InactiveOrUnloaded	A serial number is intended to be loaded on a station that belongs to an inactive or unloaded route. Please check the SmartBuild configuration.
NotOnRoute	A route was required in the request, due to the serial number being found on multiple routes, but the serial number was not found on the route supplied in the request.

Production/UnloadSerialNumber **POST**

Unloads a serial number from a station.

BUSINESS RULES

- If a serial number is provided, then the station will only be unloaded if it is at the station.
- If no serial number is provided in the request, PINpoint will unload whatever serial number is currently loaded at the requested station.
- If the serial number is present on a route that is inactive or unloaded, the route name will be returned in the response.
- Any serial number that is unloaded will be placed into the next queue.

MODEL

- *SerialNumber* (string (128), **optional**)
- *Station* (string (255), **required**)

It is recommended to include the serial number in the request because it is possible to configure a station across multiple routes. The serial number will ensure no accidental removal of part(s) an operator is currently working on.

SAMPLE

```
{
   "SerialNumber": "710111",
   "Station": "Start Station 1"
}
```

This will attempt to unload serial number 710111 from Start Station 1 and place it into its next queue. If the serial number is not currently present at the station, no action is performed, and a validation error will be present.

VALIDATION

Error	**Reason**
InvalidFormat	The value doesn't meet the configured format rules.
Unknown	The station could not be found.
Required	A value required to perform the request was not provided.
NotAtStation	The serial number provided is currently not loaded at the station.
Invalid	The request is attempted for stations that do not allow it (peer stations for example).
InternalTrackingError	An unexpected response was returned during underlying communication among components.
InactiveOrUnloaded	A serial number is intended to be unloaded on a station that belongs to an inactive or unloaded route. Please check the SmartBuild configuration.

Production/EnablePart **POST**

Enables a part for a serial number that is currently loaded at a station.

BUSINESS RULES

- A part is identified by a unique part number.
- When a part becomes enabled, the process steps defined in SmartBuild for that part are now capable of being worked on by an operator.
- Parts may only be enabled at stations that can show process steps to an operator. Automated stations and Andon Only stations may not accept an enable part request. Validations errors will be returned.
- Process steps may be defaulted to be enabled/disabled when a serial number is at a station via the SmartBuild station configuration setting **Enable Steps On Product Load**.
- Assuming the loaded serial number is not complete, if a part becomes enabled at a station, the station's cycle timer will start/resume as during normal process and appropriate time will be allocated into a **working** cycle state.

MODEL

- *SerialNumber* (string (128), **required**)
- *Station* (string (255), **required**)
- *PartNumber* (string (255), **required**)

SAMPLE

```
{
   "SerialNumber": "710111",
   "PartNumber": "KL933352",
   "Station": "Start Station 1"
}
```

This will attempt to enable part KL933352 (identified by its part number) for serial number 710111 at station Start Station 1.

VALIDATION

Error	**Reason**
InvalidFormat	The value doesn't meet the configured format rules.
Required	A value required to perform the request was not provided.
Unknown	The station could not be found.
Invalid	The station does not accept part enabling requests. The station could be an Andon Only station or Automated Station.
NotAtStation	The serial number provided is currently not loaded at the station.
NoProcessStepsDefined	No process was configured for part at the intended station, please check the SmartBuild configuration settings to ensure process steps exist and are active (process step revisions).
NotConnected	The station is currently not running or not connected to the tracking Data Manager services.
MessageDeliveryFailure	The station was unable to receive the part enable request.

Production/EnableProcess **POST**

Enables a process for a serial number that is currently loaded at a station.

BUSINESS RULES

- Processes are identified by their process names.
- When a process becomes enabled, the process steps defined in SmartBuild for that process are now capable of being worked on by an operator.
- Processes may only be enabled at stations that can show process steps to an operator. Automated stations and Andon Only stations may not accept an enable process request. Validations errors will be present.
- Process steps may be defaulted to be enabled/disabled when a serial number is at a station via the SmartBuild station configuration setting **Enable Steps On Product Load**.
- Assuming the loaded serial number is not complete, if a process becomes enabled at a station, the station's cycle timer will start/resume as during normal process and appropriate time will be allocated into a **working** cycle state.

MODEL

- *SerialNumber* (string (128), **required**)
- *ProcessName* (string (255), **required**)
- *ProcessName* (string (255), **required**)

SAMPLE

```
{
   "SerialNumber": "710111",
   "ProcessName": "TP 1",
   "Station": "Start Station 1"
}
```

This will attempt to enable process TP 1 (identified by its process name) for serial number 710111 at station Start Station 1.

VALIDATION

Error	**Reason**
InvalidFormat	The value doesn't meet the configured format rules.
Required	A value required to perform the request was not provided.
Unknown	The station could not be found.
Invalid	The station does not accept process enabling requests. The station could be an Andon Only station or Automated Station.
NotAtStation	The serial number provided is currently not loaded at the station.
NoProcessStepsDefined	No process step was configured for process at the intended station, please check the SmartBuild configuration settings to ensure process steps exist and are active (process step revisions).
NotConnected	The station is currently not running or not connected to the tracking Data Manager services.
MessageDeliveryFailure	The station was unable to receive the process enable request.

Production/DisablePart **POST**

Disables a part for a serial number that is currently loaded at a station.

BUSINESS RULES

- A part is identified by a unique part number.
- When a part becomes disabled, the process steps defined in SmartBuild for that part are no longer capable of being worked on by an operator.
- Parts may only be disabled at stations that can show process steps to an operator. Automated stations and Andon Only stations may not accept a disable part request. Validations errors will be present.
- Process steps may be defaulted to enable/disabled when a serial number is loaded from the SmartBuild station configuration setting **Enable Steps On Product Load**.
- Assuming the loaded serial number is not complete, if a part becomes disabled at a station, and no other parts are enabled, the station's cycle timer will stop as during normal process and appropriate time will be allocated into a **suspended** cycle state.

MODEL

- *SerialNumber* (string (128), **required**)
- *Station* (string (255), **required**)
- *PartNumber* (string (255), **required**)

SAMPLE

```
{
   "SerialNumber": "710111",
   "PartNumber": "KL933352",
   "Station": "Start Station 1"
}
```

This will attempt to disable part KL933352 (identified by its part number) for serial number 710111 at station Start Station 1.

VALIDATION

Error	**Reason**
InvalidFormat	The value doesn't meet the configured format rules.
Required	A value required to perform the request was not provided.
Unknown	The station could not be found.
Invalid	The station does not accept part disable requests. The station could be an Andon Only Station or Automated Station.
NotAtStation	The serial number provided is currently not loaded at the station.
NoProcessStepsDefined	No process was configured for part at the intended station, please check the SmartBuild configuration settings to ensure process steps exist and are active (process step revisions).
NotConnected	The station is currently not running or not connected to the tracking Data Manager services.
MessageDeliveryFailure	The station was unable to receive the part disable request.

Production/DisableProcess **POST**

Disables a process for a serial number that is currently loaded at a station.

BUSINESS RULES

- Processes are identified by their process names.
- When a process becomes disabled, the process steps defined in SmartBuild for that process are no longer capable of being worked on by an operator.
- Processes may only be disabled at stations that can show process steps to an operator. Automated stations and Andon Only stations may not accept a disable process request. Validations errors will be present.
- Process steps may be defaulted to enable/disabled when a serial number is loaded from the SmartBuild station configuration setting **Enable Steps On Product Load**.
- Assuming the loaded serial number is not complete, if a process becomes disabled at a station, and no other processes are enabled, the station's cycle timer will stop as during normal process and appropriate time will be allocated into a **suspended** cycle state.

MODEL

- *SerialNumber* (string (128), **required**)
- *Station* (string (255), **required**)
- *ProcessName* (string (255), **required**)

SAMPLE

```
{
   "SerialNumber": "710111",
   "ProcessName": "TP 1",
   "Station": "Start Station 1"
}
```

This will attempt to disable process TP 1 (identified by its process name) for serial number 710111 at station Start Station 1.

VALIDATION

Error	**Reason**
InvalidFormat	The value doesn't meet the configured format rules.
Required	A value required to perform the request was not provided.
Unknown	The station could not be found.
Invalid	The station does not accept process disable requests. The station could be an Andon Only Station or Automated Station.
NotAtStation	The serial number provided is currently not loaded at the station.
NoProcessStepsDefined	No process was configured for process at the intended station, please check the SmartBuild configuration settings to ensure process steps exist and are active (process step revisions).
NotConnected	The station is currently not running or not connected to the tracking Data Manager services.
MessageDeliveryFailure	The station was unable to receive the process disable request.

Production/SubmitWorkOrder **POST**

Creates a new work order or replace an existing work order if the work order number already exists and has not been started. A work order gets replaced when the same order number has been detected in the system and there are differences between the current request being submitted and the existing work order.

CALL DETAILS

- The request body contains the work order name, and a set of parts with any amount of child parts.
- A successful call echoes request data and returns a complete work order, with work order number and serial numbers assigned (as applicable).
- An unsuccessful call echoes the request data with validation errors and an HTTP error code.

BUSINESS RULES

- The serial number and quantity fields are mutually exclusive. When a serial number is provided, the quantity will be 1, and the quantity supplied will be ignored. When only quantity is provided, the system will assign serial numbers.
- Serial number uniqueness will be validated based on the **Serial Number Unique Per Serial Number Controller** attribute in PINpoint SmartBuild.
- If the **Auto Generate Work Order Numbers** attribute is set to true in PINpoint SmartBuild, then when a work order number is not provided, it will be automatically generated. If **Auto Generate Work Order Numbers** is false, a work order number will not be automatically generated and must be provided in the work order.
- Work order number uniqueness will be validated against all other work orders already in the database that were not created with this API. Work orders created with this API will be replaced by subsequent requests to submit a work order with the same work order number assuming it has not already been started and the new submission has changes present.
- Work orders can only be replaced before they have been started. Once a work order is started, further modifications will be prohibited by the API, however some manipulation can still be done by the PINpoint Tracking Web app.
- When a route is specified, the system will ignore the default route for the part, and use the specified route.
- Child parts detected as features will not have serial numbers.
- Child parts that are not detected as features will have auto-generated serial numbers.
- Contextual Data Item (CDI) Rules
    - When a CDI of the top-level part is not known to the PINpoint database, it will be added and/or associated to the top-level part.
    - When a CDI of a child part is not known to the PINpoint database, it will be ignored and won't be added to the work order or the database. The call won't fail because of this.
- Bill of Material (BOM) Rules
    - When a child part does not have a Bill of Material (BOM) configured in SmartBuild, it will be interpreted as a feature. Their child parts will be created as per their configured routes and quantities. When a child part has a BOM, a new work order item will be created for the BOM only.
    - All part numbers specified in the work order (top level PN and PN(s) identified as 'features') must have an enabled relationship to the route specified in the request or the route otherwise derived based on the top level PN.
    - If a part has a corresponding BOM in SmartBuild, the BOM will be used to insert additional parts as per BOM quantities. Serial numbers will be automatically assigned.
- Some integrations may perform bulk submissions of work orders at regular intervals. Those regular submissions may contain work orders that have already been submitted (identical), and even already started. To help prevent bulk work order submissions from failing in an unmonitored job scenario, identical work order submissions will be **skipped**, returning a successful status response, and allowing subsequent work order submissions to continue.

MODEL HIERARCHY

A work order has any number of peer parts, and each part may have any number of contextual data fields and any number of child parts.

Order

Parts (Part[])

ContextualDatas (ContextualData[])

ChildParts (ChildPart[])

MODEL HIERARCHY IN DETAIL

ORDER

- *OrderNumber* (string (128), **required** or **optional**)
  - Based on PINpoint SmartBuild configuration, Auto Generate Work Order Numbers attribute
- *ReleaseDate* (date time, with UTC offset, **optional**)
  - When not supplied or supplied with a date in the past, will be handled as to be released at current plant date/time).
  - If Tracking DM option “Sort by Provisioning Date” is set, a past date will not be modified.
- *ReleaseToProduction* (bool, **optional**)
  - If provided, and false, the work order will not be released to production regardless of ReleaseDate. In order to release it, the work order should be resubmitted with the flag set true, or the Release to Production toggle can be turned on for the work order in the PINpoint Tracking Web Application, or by loading the serial number in a station.
  - If provided, and true, the work order will be released to production on the specified ReleaseDate.
- ***Parts*** (list of **Part**, **required**)

PART

- *PartNumber* (string (255), **required**)
- *SerialNumber* (string (128), **required** if quantity is omitted, otherwise **optional**)
- *Quantity* (integer, **required** if serial number is omitted, otherwise **optional**)
- *Route* (string, **optional**)
- ***ChildParts*** (list of **ChildPart**, **optional**)
- ***ContextualDatas*** (list of **ContextualData**, **optional**)

CONTEXTUAL DATA

- *Name* (string (64), **required**)
- *Value* (string (524288), **required**)

CHILD PART

- *PartNumber* (string (255), **required**)

SAMPLE

Request:


```
{
  "OrderNumber": null,
  "ReleaseDate": "2018-06-07T09:00:00-04:00",
  "ReleaseToProduction": true,
  "Parts": [
    {
      "PartNumber": "834837",
      "SerialNumber": "710111",
      "Quantity": null,
      "Route": "Main Line",
      "ContextualDatas": [
        {
          "Name": "Color",
```

```
          "Value": "Red"
        },
        {
          "Name": "MaterialNumber",
          "Value": "14"
        }
      ],
      "ChildParts": [
        {
          "PartNumber": "993221"
        }
      ]
    },
    {
      "PartNumber": "CT90011",
      "SerialNumber": null,
      "Quantity": 2,
      "Route": null,
      "ContextualDatas": [
        {
          "Name": "Texture",
          "Value": "B11"
        }
      ],
      "ChildParts": [
        {
          "PartNumber": "CT95322"
        },
        {
          "PartNumber": "CT95391"
        }
      ]
    }
  ]
}
```


This will create a work order where a work order number of AC912001 is automatically assigned that will be released to production on June 7, 2018 at 9AM, which is 4 hours behind UTC. There are two peer parts, each with child parts.

The first part is for part number 834837, and since a serial number is provided, there is no need to generate it in the call. A route is specified so the item will be built on Main Line. It has two contextual data fields which will be associated with the item. It has a single child part for part number 993221. This child part doesn't get a serial number. Additionally, the part has a BOM configured in SmartBuild that states each time the part is added to a work order, also add two items for part number GC89991. This happens automatically during the call.

The second part is part number CT90011. Since there is a quantity given and no serial number is specified, the system will create 2 work order items and automatically generate serial numbers for both. There is no route specified, so the system will build the part on the default route as per SmartBuild configuration. The part has 2 child parts.

Response:


```
{
  "OrderNumber": "AC912001",
  "ReleaseDate": "2018-06-07T09:00:00-04:00",
  "ReleaseToProduction": true,
  "Parts": [
    {
      "PartNumber": "834837",
      "SerialNumber": "710111",
      "Quantity": null,
      "Route": "Main Line",
      "ContextualDatas": [
        {
          "Name": "Color",
```

```
            "Value": "Red"
          },
          {
            "Name": "MaterialNumber",
            "Value": "14"
          }
        ],
        "ChildParts": [
          {
            "PartNumber": "993221"
          }
        ]
      },
      {
        "PartNumber": "GC89991",
        "SerialNumber": "KL550113",
        "Quantity": null,
        "Route": "Main Line",
        "ContextualDatas": null
      },
      {
        "PartNumber": "CT90011",
        "SerialNumber": "AF70001",
        "Quantity": null,
        "Route": "Sub Assembly",
        "ContextualDatas": [
          {
            "Name": "Texture",
            "Value": "B11"
          }
        ],
        "ChildParts": [
          {
            "PartNumber": "CT95322"
          },
          {
            "PartNumber": "CT95391"
          }
        ]
      },
      {
        "PartNumber": "CT90011",
        "SerialNumber": "AF70002",
        "Quantity": null,
        "Route": "Sub Assembly",
        "ContextualDatas": [
          {
            "Name": "Texture",
            "Value": "B11"
          }
        ],
        "ChildParts": [
          {
            "PartNumber": "CT95322"
          },
          {
            "PartNumber": "CT95391"
          }
        ]
      }
    ]
  }
```


In certain scenarios, additional information will be brought back in the response. If a **null** work order number is provided, the auto-generated work order number will be included in the response. If a **null** serial number is provided for any of the top-level parts (and the quantity requested is a single unit), the auto-generated serial number will be included in the response. If a **null** route is given for a top-level part, the chosen (default) route will be included in the response. Any configured contextual data that is required and not provided for the submitted parts will be interpolated in the response for the parts that they are required for, stating validation errors.

HTTP RETURN CODES

The following table contains HTTP Return Codes unique to this endpoint. Common HTTP Return Codes can be found at **Common HTTP Return Codes**.

Code	Reason
200 OK	The work order was successfully submitted. Please note, there is a special "**skipped**" case where this code would be returned which is identified in the Business Rules.

VALIDATION

Error	Reason
Duplicate	An entry with this value already exists.
InvalidFormat	The value doesn't meet the configured format rules.
AlreadyStarted	Work order has been started and cannot be replaced.
Required	Value is missing, or for a numeric data field it isn't a positive whole number.
Unknown	Used for part number and route only. The part number or route doesn't exist.
NoDefaultRoute	Used for part number only. No route was specified, and part number doesn't have default route configured.
CannotBuildPartOnRoute	When both part number and route are known, but not associated together in SmartBuild Build view. The route may also be inactive or unloaded.
BomContextualDataRequired	Only applies to products with a BOM configured in SmartBuild. If the products in the BOM have contextual data that is required, and don't have defaults, the submit cannot go through.
ChildPartRouteDisabled	The route specified or the default route for the top-level part is found but disabled for a child part.
NoChildPartRouteFound	Either no routes are defined for the feature, or the route specified or the default route for the top-level part is not found for a child part.

Production/WorkOrders **DELETE**

Deletes a work order by using a work order number.

QUERY STRING PARAMETERS

- *OrderNumber* (string (128), **required**)

VALIDATION

Error	**Reason**
Required	A value required to perform the request was not provided.
NotFound	The order number was not present in the PINpoint system.
InvalidFormat	The value doesn't meet the configured format rules.
AlreadyStarted	The work order could not be deleted because it's already being worked on.

Production/TesterResults **POST**

Submits a test result for a Broadcast Tester device for a serial number that is currently loaded at a station.

BUSINESS RULES

- The Data Manager service must be running, and its device engine has started.
- A serial number must be loaded at the station, subscribed to a device, and awaiting a test result.
- The device name must match the supplied Device ID.
- Test results are unsolicited, they do not need to match a corresponding request ID.
- Only a single overall result is supported.

MODEL HIERARCHY

- *DeviceId* (string (255), **required**)
- *SerialNumber* (string (255), **required**)
- *Result* (TesterResultEnum)
- *Timestamp* (DateTime (UTC), **optional**)

MODEL HIERARCHY IN DETAIL

```
public enum TesterResultEnum
{
    Fault = 0,
    Fail,
    Pass,
    Ignore
}
```

SAMPLE

Request:

```
{
   "DeviceId": "BroadcastTester",
   "Result": "Pass",
   "SerialNumber": "00014240"
}
```

Response:

```
{
   "Message": "Success",
   "Success": true,
   "ErrorCode": 0
}
```

VALIDATION

The incoming request will be checked for rules, and broken rules will be sent back to the caller in the Error Code property, when the Success state is false. These broken rules will be present on a response containing a **200 HTTP** error code (OK).

Error	**Reason**
GeneralFailureSendingResult (94)	An application error has occurred.
GeneralFailureRequestingInformation (100)	Unused
SocketError (96)	Unused
SocketTimeout (95)	Unused

CannotDetermineDataManager (98)	Cannot find the Data Manager service responsible for listening for the requests for the provided Device ID.
TesterNotFound (97)	No device with a matching Device ID was found.
NullOrInvalidArgument (99)	The value doesn't meet the configured format rules.

SAMPLE VALIDATION ERROR


```
{
   "Message": "TesterNotFound",
   "Success": false,
   "ErrorCode": 97,
   "Parameters": null
}
```

Production/OperationValues **GET**

Gets scan operations values for a serial number. Scan operations must have an External ID to be considered by this query. Only the last valid value for each operation will be returned.

BUSINESS RULES

- Scan operations must have an External ID to be considered.
- Only valid values will be returned.
- It does not matter if the serial number has been rebuilt, reloaded, or reset; the returned value will be the most recent value.
- The station where the operation belongs to must be active.

QUERY STRING PARAMETERS

- *SerialNumber* (string (128), **required**)
  - The serial number of the top-level part
- *Line* (string (255), **optional**)
  - The line to filter where operations are found

RESPONSE MODEL

- *SerialNumber* (echoed from the request)
- *LineName* (echoed from the request)
- *OperationValue[]*
  - *Operation* (the operation with External ID and a valid value)
  - *Value* (the value for the operation)

SAMPLE RESPONSE

```
{
   "SerialNumber": "WAKA230004",
   "OperationValues": [
      {
         "Operation": "My external scan operation ID",
         "Value": "Value for external operation"
      }
   ]
}
```

VALIDATION

There are no validation errors, in case no value has been found, the method will return an object with an empty collection.

SAMPLE NO VALUE FOUND

```
{
   "SerialNumber": "WRONG_SN",
   "OperationValues": []
}
```

Production/Quality

Production/Quality/Product **GET**

Returns product quality with accompanying child product quality that matches the filters.

QUERY STRING PARAMETERS

- *SerialNumber* (string (128), **required**)
  - the serial number of the top-level part
- *PartNumber* (string (255), **optional**)
  - the part number of the top-level part to filter the result when multiple products of the same serial number are used in a production line
- *Line* (string (255), **optional**)
  - the line to filter where the top-level products are found
- *Route* (string (255), **optional**)
  - If known and provided, this parameter will trump the line to filter where the top-level parts are found. This is a more concise filter.

BUSINESS RULES

- Product Quality and Child Product Quality are described as follows:
  - NotSet, the product quality has not yet been calculated.
  - Green, the product is shippable.
  - Yellow, the product has an assembly issue and may still be considered shippable.
  - Red, the product has an assembly issue and should not be considered shippable.
- Product assembly issues that determine product quality include:
  - Rejects
  - Bypasses
  - Defects
  - Incomplete stations
- The severity of an assembly issue and how it impacts the product quality calculation may be configured on the hardware view in SmartBuild.
- Product Quality and/or Child Product Quality are calculated based on a worst-case scenario; red trumps yellow, yellow trumps green, green trumps grey (which is considered NotSet).

RESPONSE MODEL HIERARCHY

A response will echo the requested parameters and will include the top-level product quality as well as accompanying child product quality. The top-level product qualities may recursively include additional child product qualities depending on how the route/line is configured. For a linear line, only a single level of product qualities will be included in the response. For a composite line, multiple levels will be included in the response, and the number of levels will equal the number of geometric composite branches.

ProductQualityRequest

ProductQualities (list of **ProductQuality**)

ChildProductQualities (list of **ProductQuality**)

...additional recursive ChildProductQualities...

RESPONSE MODEL HIERARCHY IN DETAIL

PRODUCT QUALITY REQUEST

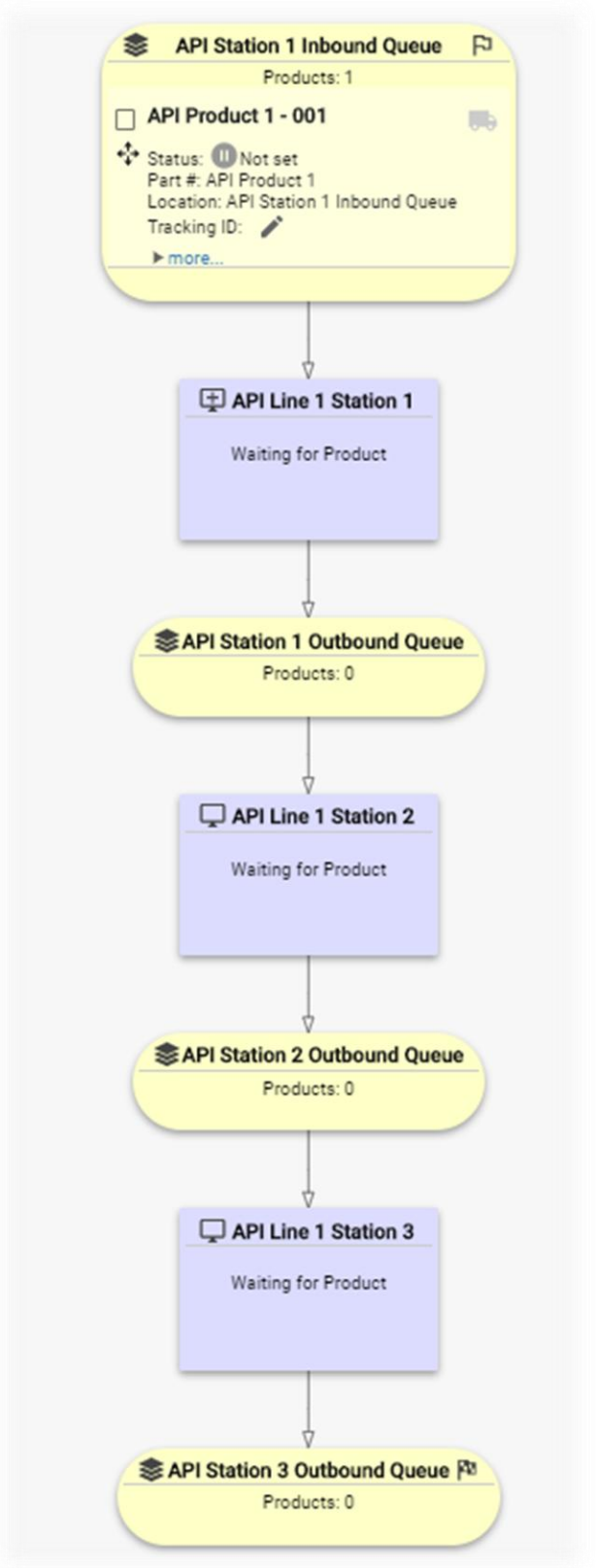
- *SerialNumber* (echoed from the request)
- *PartNumber* (echoed from the request)

- *Line* (echoed from the request)
- *Route* (echoed from the request)
- ***ProductQuality***

PRODUCT QUALITY

- *Route*
  - The route PINpoint has found the serial number to be present on
- *Location*
  - The current location name of where the serial number is
  - This may be a **queue**, a **station**, or a **graveyard** location (a graveyard meaning the product has been completed and either is available to be merged into the next main assembly or is awaiting to me merged
- *Status*
  - The status of the serial number
- *ProductQuality* (string)
  - The main product quality of a linear line/route, or the main product quality of a composite branch
- *ChildProductQualities* (list of **ProductQuality**)
  - The product quality of this product's child product, this will only be present for products which have children

LINEAR LINE EXAMPLE (NOT STARTED)



REQUEST:

http(s)://*YourServerName/PinPoint.WebService/api/v1/External/****Production****/****Quality****/*Product?**PartNumber=API Product 1**&**Line=API Line 1**&**SerialNumber=API Product 1 – 001**

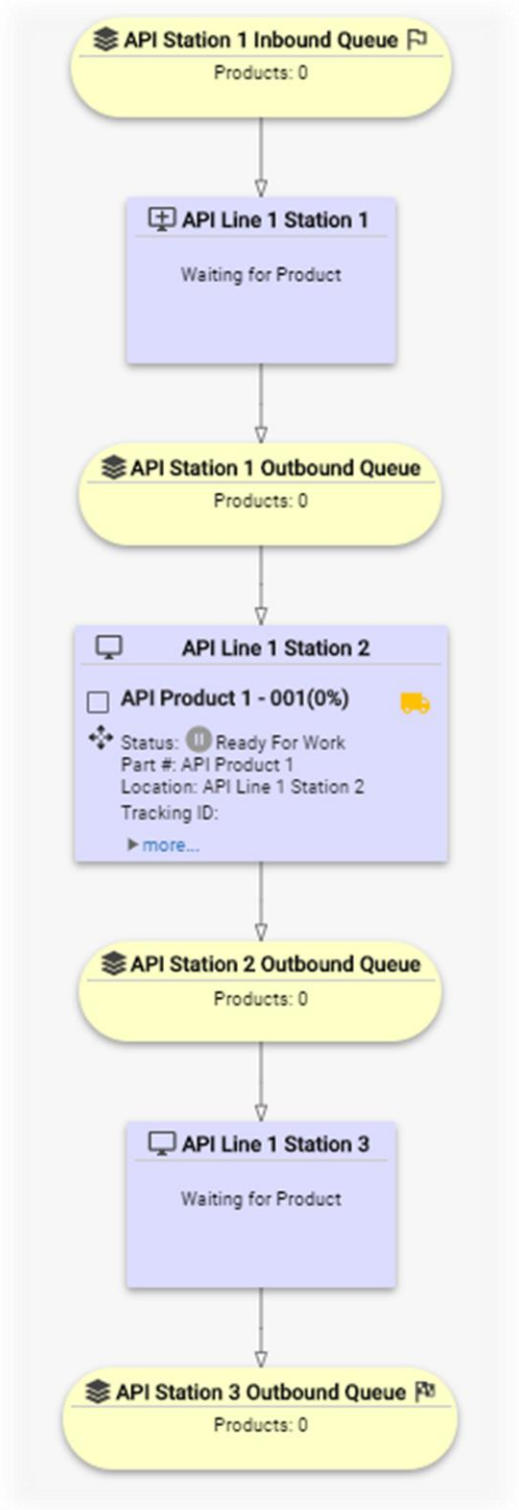
RESPONSE:


```
{
   "SerialNumber": "API Product 1 - 001",
   "PartNumber": "API Product 1",
   "Line": "API Line 1",
   "ProductQualities": [
      {
         "Route": "API Route 1",
         "Location": "API Station 1 Inbound Queue",
         "Status": "NotSet",
         "ProductQuality": "NotSet",
         "ChildProductQualities": []
      }
   ]
}
```


The root Product Qualities node is the overall product quality. It considers the aggregation of all assembly issues that were raised during the assembly of the top-level product.

Both linear and composite lines will have a root Product Qualities node. However, for composite lines, there will be an additional Child Product Quality property.

LINEAR LINE EXAMPLE (STARTED)



REQUEST:

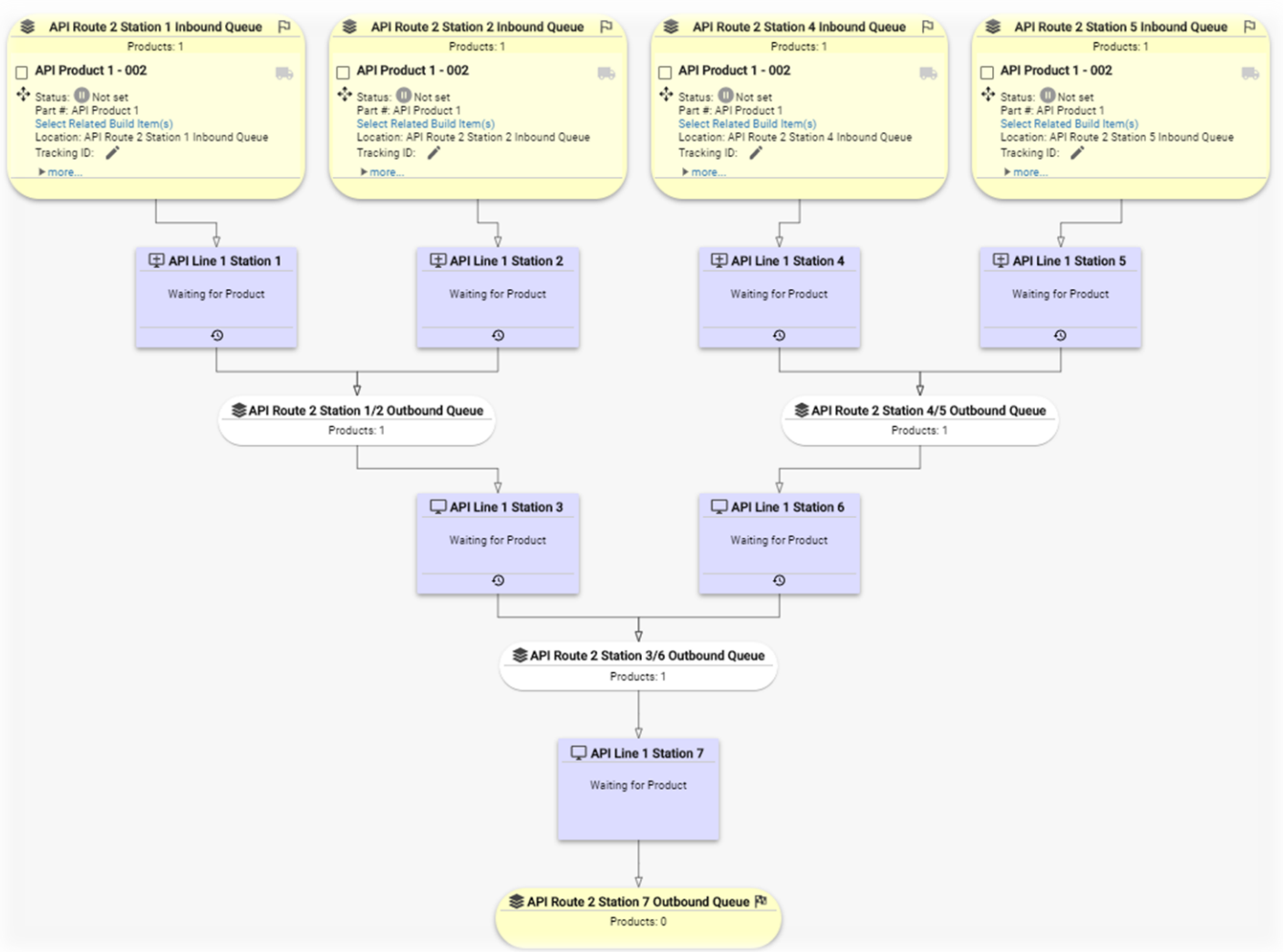
http(s)://*YourServerName/PinPoint.WebService/api/v1/External/****Production/Quality****/Product?**PartNumber=API Product 1**&**Line=API Line 1**&**SerialNumber=API Product 1 - 001**

RESPONSE:


```
{
   "SerialNumber": "API Product 1 - 001",
   "PartNumber": "API Product 1",
   "Line": "API Line 1",
   "ProductQualities": [
      {
         "Route": "API Route 1",
         "Location": "API Line 1 Station 2",
         "Status": "Started",
         "ProductQuality": "Yellow",
         "ChildProductQualities": []
      }
   ]
}
```


There are no child product qualities for this request because the line is linear.

COMPOSITE LINE EXAMPLE (NOT STARTED)



REQUEST:

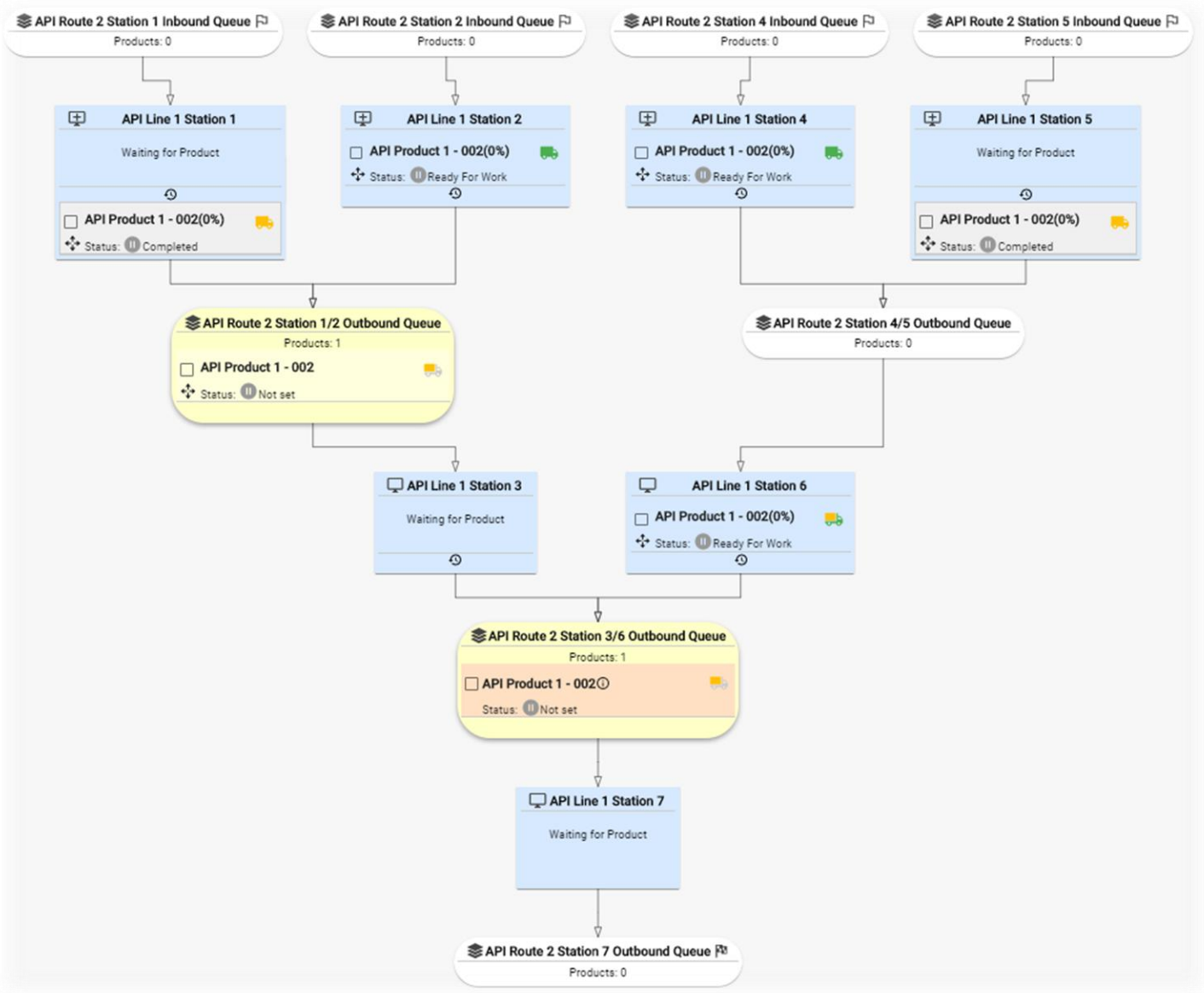
http(s)://*YourServerName/PinPoint.WebService/api/v1/External/****Production****/****Quality****/*Product?**PartNumber=API Product 1**&**Line=API Line 1**&**SerialNumber=API Product 1 - 002**

RESPONSE:


```
{
   "SerialNumber": "API Product 1 - 002",
   "PartNumber": "API Product 1",
   "Line": "API Line 1",
   "ProductQualities": [
      {
         "Route": "API Route 2",
         "Location": "API Route 2 Station 3/6 Outbound Queue",
         "Status": "NotSet",
         "ProductQuality": "NotSet",
         "ChildProductQuality": "NotSet",
         "ChildProductQualities": [
            {
               "Route": "API Route 2",
               "Location": "API Route 2 Station 4/5 Outbound Queue",
               "Status": "NotSet",
               "ProductQuality": "NotSet",
               "ChildProductQuality": "NotSet",
               "ChildProductQualities": [
                  {
                     "Route": "API Route 2",
                     "Location": "API Route 2 Station 4 Inbound Queue",
                     "Status": "NotSet",
                     "ProductQuality": "NotSet",
                     "ChildProductQualities": []
                  },
                  {
                     "Route": "API Route 2",
                     "Location": "API Route 2 Station 5 Inbound Queue",
                     "Status": "NotSet",
                     "ProductQuality": "NotSet",
                     "ChildProductQualities": []
                  }
               ]
            },
            {
               "Route": "API Route 2",
               "Location": "API Route 2 Station 1/2 Outbound Queue",
               "Status": "NotSet",
               "ProductQuality": "NotSet",
               "ChildProductQuality": "NotSet",
               "ChildProductQualities": [
                  {
                     "Route": "API Route 2",
                     "Location": "API Route 2 Station 1 Inbound Queue",
                     "Status": "NotSet",
                     "ProductQuality": "NotSet",
                     "ChildProductQualities": []
                  },
                  {
                     "Route": "API Route 2",
                     "Location": "API Route 2 Station 2 Inbound Queue",
                     "Status": "NotSet",
                     "ProductQuality": "NotSet",
                     "ChildProductQualities": []
                  }
               ]
            }
         ]
      }
   ]
}
```


In this example, each child product quality is populated per composite branch as well as their children. Currently, nothing on the line has started to be worked on.

COMPOSITE LINE EXAMPLE (STARTED)



REQUEST:

http(s)://*YourServerName/PinPoint.WebService/api/v1/External/****Production****/****Quality****/*Product?**PartNumber=API Product 1**&**Line=API Line 1**&**SerialNumber=API Product 1 - 002**

RESPONSE:


```
{
   "SerialNumber": "API Product 1 - 002",
   "PartNumber": "API Product 1",
   "Line": "API Line 1",
   "ProductQualities": [
      {
         "Route": "API Route 2",
         "Location": "API Route 2 Station 3/6 Outbound Queue",
         "Status": "NotSet",
         "ProductQuality": "NotSet",
         "ChildProductQuality": "Yellow",
         "ChildProductQualities": [
            {
```

```
                "Route": "API Route 2",
                "Location": "API Line 1 Station 6",
                "Status": "Started",
                "ProductQuality": "Green",
                "ChildProductQuality": "Yellow",
                "ChildProductQualities": [
                   {
                      "Route": "API Route 2",
                      "Location": "API Line 1 Station 4",
                      "Status": "Started",
                      "ProductQuality": "Green",
                      "ChildProductQualities": []
                   },
                   {
                      "Route": "API Route 2",
                      "Location": "(Graveyard)",
                      "Status": "Completed",
                      "ProductQuality": "Yellow",
                      "ChildProductQualities": []
                   }
                ]
             },
             {
                "Route": "API Route 2",
                "Location": "API Route 2 Station 1/2 Outbound Queue",
                "Status": "NotSet",
                "ProductQuality": "NotSet",
                "ChildProductQuality": "Yellow",
                "ChildProductQualities": [
                   {
                      "Route": "API Route 2",
                      "Location": "(Graveyard)",
                      "Status": "Completed",
                      "ProductQuality": "Yellow",
                      "ChildProductQualities": []
                   },
                   {
                      "Route": "API Route 2",
                      "Location": "API Line 1 Station 2",
                      "Status": "Started",
                      "ProductQuality": "Green",
                      "ChildProductQualities": []
                   }
                ]
             }
          ]
       }
    ]
}
```

VALIDATION

These broken rules will be present on a response containing a **400 HTTP** error code (bad request). Common validation errors can be found here.

Error	**Reason**
InvalidFormat	The value doesn't meet the configured format rules. An input provided exceeds the allowable length.
Required	A value required to fulfil the request was not provided.

Production/Quality/ProductDetail **GET**

Returns product quality with accompanying child product quality that matches the filters, including full details about relevant quality violations. The parameters and general structure are the same as the Product Quality method, but with more objects/information in the response message body.

QUERY STRING PARAMETERS

- *SerialNumber* (string (128), **required**)
  - The serial number of the top-level part
- *PartNumber* (string (255), **optional**)
  - The part number of the top-level product to filter the result when multiple products of the same serial number are used in a production line
- *Line* (string (255), **optional**)
  - The line to filter where the top-level products are found
- *Route* (string (255), **optional)**
  - If known and provided, this parameter will trump the line to filter where the top-level products are found. This is a more concise filter.

BUSINESS RULES

- Product Quality and Child Product Quality are described as follows:
  - NotSet, the product quality has not yet been calculated.
  - Green, the product is shippable.
  - Yellow, the product has an assembly issue and may still be considered shippable.
  - Red, the product has an assembly issue and should not be considered shippable.
- Product assembly issues that determine product quality include:
  - Rejects
  - Bypasses
  - Defects
  - Incomplete Stations
- If any of the above issues exist, then the details will be populated in corresponding arrays.
- The severity of an assembly issue and how it impacts the product quality calculation may be configured on the hardware view in SmartBuild.
- Product Quality and/or Child Product Quality are calculated based on a worst-case scenario; red trumps yellow, yellow trumps green, green trumps grey (which is considered NotSet).

RESPONSE MODEL HIERARCHY

A response will echo the requested parameters and will include the top-level product quality as well as accompanying child product quality. The top-level product qualities may recursively include additional child product qualities depending on how the route/line is configured. For a linear line, only a single level of product qualities will be included in the response. For a composite line, multiple levels will be included in the response, and the number of levels will equal the number of geometric composite branches.

ProductQualityRequest

ProductQualities *(ProductQuality[])*

QualityIssues *(QualityIssue[])*

ChildProductQualities *(ProductQuality[])*

QualityIssues *(QualityIssue[])*

...recursive ChildProductQualities...

RESPONSE MODEL HIERARCHY IN DETAIL

PRODUCT QUALITY REQUEST

- *SerialNumber* (echoed from the request)
- *PartNumber* (echoed from the request)
- *Line* (echoed from the request)
- *Route* (echoed from the request)
- *ProductQualities* (**ProductQuality[]**)

PRODUCT QUALITY

- *Route*
  - The route PINpoint has found the serial number to be present on
- *Location*
  - The current location name of where the serial number is
  - This may be a **queue**, a **station**, or a **graveyard** location (a graveyard meaning the product has been completed and either is available to be merged into the next main assembly or is awaiting to me merged.
- *Status*
  - The status of the serial number
- *ProductQuality* (string)
  - The main product quality of a linear line/route, or the main product quality of a composite branch
- *QualityIssues* (**Defect[]**)
  - A flat list of any Defect (Adhoc, Inspection Operation, Verification Operation, Containment, Shortage), Station Bypass, Step Bypass, Step Reset, Incomplete Step and Reject, including resolution information and location/operator details where applicable.
- *ChildProductQualities* (**ProductQuality[]**)
  - The product quality of this product's child product, this will only be present for products which have children

DEFECT

- *DefectId* (int)
- *DefectType* (string)
- *BuildItemId* (GUID)
- *SerialNumber* (string)
- *RouteName* (string)
- *stepName* (string)
- *StepDescription* (string)
- *CurrentStatusType* (string)
- *ProductionAdvisoryId* (int)
- *ProductionAdvisory* (**ProductionAdvisory**)
- *DefectStatuses* (**DefectStatus[]**)
- *DefectNotes* (**DefectNote[]**)

PRODUCTION ADVISORY

- *ProductionAdvisoryId* (int)
- *ProductionAdvisoryType* (string)
- *ExternalState* (string)
- *UserName* (string)
- *PartNumber* (string)
  - the external PartNumber
- *Quantity* (string (255))

DEFECT STATUS

- *DefectId* (int)
- *DefectStatusType* (string)
- *UserName* (string)
- *Date* (DateTime)

- *StationId* (int)
- *StationName* (string)
- *Reason* (string)
- *ReasonGroup* (string)

DEFECT NOTE

- *DefectId* (int)
- *UserName* (string)
- *Date* (DateTime)
- *Note* (string)

SAMPLE PRODUCT DETAIL RESPONSE


```
{
   "SerialNumber": "API Product 1 - 001",
   "PartNumber": "API Product 1",
   "Line": "API Line 1",
   "ProductQualities": [
      {
         "Route": "API Route 1",
         "Location": "API Line 1 Station 2",
         "Status": "Started",
         "ProductQuality": "Yellow",
         "QualityIssues": [
            {
            "DefectId": 3537,
            "DefectType": "Adhoc",
            "BuildItemId": "4a6329b5-b741-45c3-e7a8-08dd50f71637",
            "SerialNumber": "0000060",
            "RouteName": "Simple Route",
            "StationName": "Stn3",
            "StepName": "Defect Step",
            "StepDescription": "Defect Step",
            "CurrentStatusType": "Resolved",
            "CreatedUser": "m, r",
            "CreatedDate": "2025-02-19T10:10:32.323-05:00",
            "ClosedDate": "2025-02-19T10:10:42.209-05:00",
            "DefectStatuses": [
               {
               "DefectId": 3537,
               "DefectStatusType": "Initialized",
               "UserName": "m, r",
               "Date": "2025-02-19T10:10:32.323-05:00",
               "StationId": 1150,
               "StationName": "Stn3",
               "Reason": "Sticker Misalign",
               "ReasonGroup": "Premium Model\\Fender Reasons",
               "ReasonDescription": "replace sticker"
               },
               {
               "DefectId": 3537,
               "DefectStatusType": "Resolved",
               "UserName": "m, r",
               "Date": "2025-02-19T10:10:42.209-05:00",
               "StationId": 1150,
               "StationName": "Stn3",
               "Reason": "Reason 1",
               "ReasonGroup": "Acknowledge Reasons",
               "ReasonDescription": "replaced"
               }
            ],
            "DefectNotes": []
            },
            {
            "DefectId": 3536,
            "DefectType": "Adhoc",
            "BuildItemId": "4a6329b5-b741-45c3-e7a8-08dd50f71637",
            "SerialNumber": "0000060",
            "RouteName": "Simple Route",
            "StationName": "Stn3",
```

```
"StepName": "Defect Step",
"StepDescription": "Defect Step",
"CurrentStatusType": "Initialized",
"CreatedUser": "m, r",
"CreatedDate": "2025-02-19T10:10:02.267-05:00",
"DefectStatuses": [
   {
   "DefectId": 3536,
   "DefectStatusType": "Initialized",
   "UserName": "m, r",
   "Date": "2025-02-19T10:10:02.267-05:00",
   "StationId": 1150,
   "StationName": "Stn3",
   "Reason": "Fastener Missing",
   "ReasonGroup": "Premium Model\\Fender Reasons",
   "ReasonDescription": ""
   }
],
"DefectNotes": []
},
{
"DefectId": 3569,
"DefectType": "Reject",
"BuildItemId": "4a6329b5-b741-45c3-e7a8-08dd50f71637",
"SerialNumber": "0000060",
"RouteName": "Simple Route",
"StationName": "Stn3",
"CurrentStatusType": "Initialized",
"CreatedUser": "PpAdmin, PpAdmin",
"CreatedDate": "2025-02-19T10:11:33.87-05:00",
"DefectStatuses": [
   {
   "DefectId": 3569,
   "DefectStatusType": "Initialized",
   "UserName": "PpAdmin, PpAdmin",
   "Date": "2025-02-19T10:11:33.87-05:00",
   "StationId": 1150,
   "StationName": "Stn3",
   "Reason": "Reject Reason 1"
   }
],
"DefectNotes": []
},
{
"DefectId": 3538,
"DefectType": "StepBypass",
"BuildItemId": "4a6329b5-b741-45c3-e7a8-08dd50f71637",
"SerialNumber": "0000060",
"RouteName": "Simple Route",
"StationName": "Stn3",
"StepName": "Defect Step",
"StepDescription": "Defect Step",
"CurrentStatusType": "Acknowledged",
"CreatedUser": "PpAdmin, PpAdmin",
"CreatedDate": "2025-02-19T10:11:11.96-05:00",
"DefectStatuses": [
   {
   "DefectId": 3538,
   "DefectStatusType": "Initialized",
   "UserName": "PpAdmin, PpAdmin",
   "Date": "2025-02-19T10:11:11.96-05:00",
   "StationId": 1150,
   "StationName": "Stn3",
   "Reason": "Tool Issue",
   "ReasonGroup": "Bypass Reasons",
   "ReasonDescription": "broken tool"
   },
   {
   "DefectId": 3538,
   "DefectStatusType": "Acknowledged",
   "UserName": "PpAdmin, PpAdmin",
   "Date": "2025-02-19T10:12:48.396-05:00",
   "StationId": 1150,
   "StationName": "Stn3",
```

```
                "Reason": "Reason 1",
                "ReasonGroup": "Acknowledge Reasons",
                "ReasonDescription": "acked"
                }
            ],
            "DefectNotes": []
            },
            {
            "DefectType": "IncompleteStep",
            "BuildItemId": "4a6329b5-b741-45c3-e7a8-08dd50f71637",
            "SerialNumber": "0000060",
            "RouteName": "Simple Route",
            "StationName": "Stn1",
            "StepName": "Step 1",
            "CurrentStatusType": "Acknowledged",
            "ClosedDate": "2025-02-19T10:12:14.889-05:00",
            "DefectStatuses": [
                {
                "DefectStatusType": "Acknowledged",
                "UserName": "PpAdmin PpAdmin",
                "Date": "2025-02-19T10:12:14.889-05:00",
                "StepName": "Step 1",
                "Reason": "Manually Repaired",
                "ReasonGroup": "Process Step ACK Reasons",
                "ReasonDescription": ""
                }
            ],
            "DefectNotes": []
            },
            {
            "DefectType": "IncompleteStep",
            "BuildItemId": "4a6329b5-b741-45c3-e7a8-08dd50f71637",
            "SerialNumber": "0000060",
            "RouteName": "Simple Route",
            "StationName": "Stn1",
            "StepName": "Custom Step",
            "CurrentStatusType": "Initialized",
            "DefectStatuses": [],
            "DefectNotes": []
            },
        ],
        "ChildProductQualities": []
    }
  ]
}
```


VALIDATION

These broken rules will be present on a response containing a **400 HTTP** error code (bad request). Common validation errors can be found here.

Error	**Reason**
InvalidFormat	The value doesn't meet the configured format rules. An input provided exceeds the allowable length.
Required	A value required to fulfil the request was not provided.

Production/Quality/ProductionAdvisory

Production/Quality/ProductionAdvisory **GET**

Returns a list of all Production Advisories that match the filters. This will not return defects under the advisory.

QUERY STRING PARAMETERS

- *Type* (string, **optional**)
  - the production advisory type, must have match with an existing Production Advisory Type
- *State* (string, **optional**)
  - the production advisory state
- *ExternalId* (string, **optional**)
  - the production advisory external ID

SAMPLE

Request:

http(s)://YourServerName/PinPoint.WebService/External/Production/Quality/ProductionAdvisory?Type=Shortage&State=Install&ExternalId=123

Response:

```
{
   “Type”: “Shortage”,
   “State”: “Install”,
   “ExternalId”: “123”,
   “ProductionAdvisories”: [
      {
         “ProductionAdvisoryId”: 5,
         “ProductionAdvisoryType”: “Shortage”,
         “ExternalState”: “Install”,
         “ExternalId”: “123”,
         “UserName”: “admin”,
         “PartNumber”: “PartOne”,
         “Quantity”: 51,
         “DetailsJson”: “{‘AreaResponsible’: ‘Area 51’,  ‘ExternalUrl’: ‘https://topsecret-location.com’}”
      },
      {
         “ProductionAdvisoryId”: 6,
         “ProductionAdvisoryType”: “Shortage”,
         “ExternalState”: “Install”,
         “ExternalId”: “123”,
         “UserName”: “admin”,
         “PartNumber”: “PartOne”,
         “Quantity”: 2,
         “DetailsJson”: “{‘AreaResponsible’: ‘Area 51’,  ‘ExternalUrl’: ‘https://topsecret-location.com’}”
      },
   ]
}
```

Production/Quality/ProductionAdvisory/{productionAdvisoryId} **GET**

Returns a production advisory by its id.

QUERY STRING PARAMETERS

None. The Production Advisory ID is a path segment (part of the URL), not a query parameter.

SAMPLE

Request:

http(s)://YourServerName/PinPoint.WebService/External/Production/Quality/ProductionAdvisory/5

Response:


```
{
   "ProductionAdvisoryId": 5,
   "ProductionAdvisory": {
      "ProductionAdvisoryId": 5,
      "ProductionAdvisoryType": "Shortage",
      "ExternalState": "Install",
      "ExternalId": "123",
      "UserName": "admin",
      "PartNumber": "PartOne",
      "Quantity": 51,
      "DetailsJson": "{'AreaResponsible': 'Area 51',  'ExternalUrl': 'https://topsecret-location.com'}",
      "QualityIssues": [
         {
            "DefectId": 51,
            "DefectType": "Shortage",
            "BuildItemId": "4599721d-7f8c-4918-8a32-08dd21382e7d",
            "SerialNumber": "00000375",
            "RouteName": "route 1",
            "CurrentStatusType": "Resolved",
            "ProductionAdvisoryId": 5,
            "DefectStatuses": [
               {
                  "DefectId": 51,
                  "DefectStatusType": "Initialized",
                  "Date": "2025-04-29T13:02:03.44-04:00",
                  "StationId": 1049,
                  "StationName": "station 1"
               },
               {
                  "DefectId": 51,
                  "DefectStatusType": "Resolved",
                  "Date": "2025-04-29T14:22:11.858-04:00",
                  "StationId": 1049,
                  "StationName": "station 1",
                  "Reason": "C2"
               }
            ],
            "DefectNotes": []
         }
      ]
   }
}
```

Production/Quality/ProductionAdvisory **POST**

Creates a new production advisory.

MODEL

- *ProductionAdvisoryType* (string, **required**)
- *ExternalId* (string (100), the 3rd party external ID, **required** for *Containment*)
- *ExternalState* (string (50), the 3rd party external state, **required** for *Containment*)
- *PartNumber* (string (100), the part that is short, **required** for *Shortage*)
- *Quantity* (int, the quantity that is short, **required** for *Shortage*)
- *DetailsJson* (string (4000), a JSON string to capture any additional data, **optional**)

SAMPLE

Request:

```
{
  "ProductionAdvisoryType": "Shortage",
  "ExternalState": "OPEN",
  "ExternalId": 4,
  "PartNumber" : "Shortage_Test",
  "Quantity": 4
}
```

Response:

```
{
    "ProductionAdvisoryId": 49,
    "ProductionAdvisoryType": "Shortage",
    "ExternalState": "OPEN",
    "ExternalId": "4",
    "PartNumber": "Shortage_Test",
    "Quantity": 4
}
```

Production/Quality/ProductionAdvisory **PATCH**

Updates a production advisory.

MODEL

- *ProductionAdvisoryId* (string, **required**)
- *ExternalId* (string (100), **optional**)
- *ExternalState* (string (50), **optional**)
- *PartNumber* (string (100), **optional**)
- *Quantity* (int, **optional**)
- *DetailsJson* (string (4000), **optional**)

SAMPLE

Request:

```
{
  "ProductionAdvisoryId": 48,
  "ExternalState": "CLOSED",
  "DetailsJSON": "{'AreaResponsible': 'Area 51',  'ExternalUrl': 'https://topsecret-location.com'}"
}
```

Response:

```
{
    "ProductionAdvisoryId": 48,
    "ProductionAdvisoryType": "Containment",
    "ExternalState": "CLOSED",
    "ExternalId": "A",
    "DetailsJson": "{'AreaResponsible': 'Area 51',  'ExternalUrl': 'https://topsecret-location.com'}"
}
```

Production/Quality/Issues

Production/Quality/Issues **GET**

Returns a list of all issues (Adhoc, Verification, Inspection, Shortage, Containment Exception, Bypass, Reject, Incomplete Station/Steps) that match the filters.

QUERY STRING PARAMETERS

- *SerialNumber* (string (128), **required**)
- *PartNumber* (string (100), **required** if the serial number is ambiguous)
- *Line* (string (255), **required** if the serial number is ambiguous)
- *Route* (string (255), **required** if the serial number is ambiguous)

SAMPLE

Request:


```
{
  "SerialNumber": "0000001",
}
```


Response:


```
{
  "SerialNumber": "0000010",
  "QualityIssues": [
    {
      "DefectId": 3649,
      "DefectType": "Adhoc",
      "BuildItemId": "6cb50c23-eda6-4352-7e19-08dd7761d879",
      "SerialNumber": "0000010",
      "RouteName": "Simple Route",
      "StationName": "Stn1",
      "CurrentStatusType": "Initialized",
      "CreatedUser": "User, External",
      "CreatedDate": "2025-04-09T15:17:01.766-04:00",
      "DefectStatuses": [
        {
          "DefectId": 3649,
          "DefectStatusType": "Initialized",
          "UserName": "User, External",
          "Date": "2025-04-09T15:17:01.766-04:00",
          "StationId": 1107,
          "StationName": "Stn1",
          "Reason": "Fastener Missing",
          "ReasonGroup": "Fender Reasons"
        }
      ],
      "DefectNotes": []
    },
    ...
}
```

Production/Quality/Issues/{defectId} **GET**

Returns a single Defect by Defect ID.

QUERY STRING PARAMETERS

None. The defect ID is a path segment (part of the URL), not a query parameter.

SAMPLE

Request:

http(s)://YourServerName/PinPoint.WebService/External/Production/Quality/Issues/89

Response:


```
{
   "DefectId": 89,
   "DefectType": "Adhoc",
   "BuildItemId": "f3dc1ec8-f930-46de-6aba-08dd49e3c69b",
   "SerialNumber": "00000397",
   "RouteName": "route 1",
   "StepName": "step 1 name",
   "StepDescription": "step 1 desc",
   "CurrentStatusType": "Approved",
   "ClosedDate": "2025-05-08T11:32:35.525-04:00",
   "DefectStatuses": [
      {
         "DefectId": 89,
         "DefectStatusType": "Initialized",
         "Date": "2025-05-08T11:31:58.029-04:00",
         "StationId": 1049,
         "StationName": "station 1",
         "Reason": "A1",
         "ReasonGroup": "Reason Group A"
      },
      {
         "DefectId": 89,
         "DefectStatusType": "Resolved",
         "Date": "2025-05-08T11:32:23.303-04:00",
         "StationId": 1049,
         "StationName": "station 1",
         "Reason": "C2"
      },
      {
         "DefectId": 89,
         "DefectStatusType": "Approved",
         "Date": "2025-05-08T11:32:35.525-04:00",
         "StationId": 1049,
         "StationName": "station 1"
      }
   ],
   "DefectNotes": []
}
```

Production/Quality/Issues **POST**

Creates a new quality issue.

CALL DETAILS

- The type of exception is denoted in the JSON payload as **DefectType**.
- Optionally, a **ProductionAdvisory** can be included with the Defect to create a Containment Exception or Shortage in tandem.

MODEL

- *SerialNumber* (string, **required**)
- *PartNumber* (string, **required** if the Serial Number is ambiguous)
- *LineName* (string, **required** if the Serial Number is ambiguous)
- *RouteName* (string, **required** if the Serial Number is ambiguous)
- *DefectType* (string, **required**)
- *DefectStatus* (*ExternalDefectStatus*[], **required**)
- *ProductionAdvisory* (ExternalProductionAdvisory, **required** for Containment and Shortage defects)

EXTERNAL DEFECT STATUS

- *StationId* (int, **required**)
- *UserName* (string, **optional**)
- *Reason* (string, **required** for Adhoc, Verification, Inspection, Containment, Bypass and Reject)
- *ReasonGroup* (string, **required** for Adhoc, Verification, Inspection, Containment, Bypass and Reject)

EXTERNAL PRODUCTION ADVISORY

- *ProductionAdvisoryType* (string, **required**)
- *ExternalId* (string (100), the 3rd party external ID, **required** for *Containment*)
- *ExternalState* (string (50), the 3rd party external state, **required** for *Containment*)
- *PartNumber* (string (100), the part that is short, **required** for *Shortage*)
- *Quantity* (int, the quantity that is short, **required** for *Shortage*)
- *DetailsJson* (string (4000), a JSON string to capture any additional data, **optional**)

SAMPLE

Request:


```
{
  "SerialNumber": "0000010",
  "DefectType": "Adhoc",
  "DefectStatuses": [
    {
      "StationId": 1107,
      "UserName": "ExternalUser",
      "Reason": "Fastener Missing",
      "ReasonGroup": "Fender Reasons"
    }
  ]
}
```


Response:


```
{
  "DefectId": 3650,
  "BuildItemId": "6cb50c23-eda6-4352-7e19-08dd7761d879",
  "SerialNumber": "0000010",
  "PartNumber": "Simple Part",
  "RouteName": "Simple Route",
  "CurrentStatusType": "Initialized",
  "CreatedUser": "External User",
  "CreatedDate": "2025-04-10T04:29:42.8876494-04:00",
  "DefectStatuses": [
    {
```

```
      "DefectId": 3650,
      "DefectStatusType": "Initialized",
      "UserName": "External User",
      "Date": "2025-04-10T04:29:42.8876494-04:00",
      "StationId": 1107,
      "Reason": "Fastener Missing",
      "ReasonGroup": "Fender Reasons"
    }
  ],
  "DefectNotes": []
}
```

Production/Quality/Issues **PATCH**

Updates an existing defect. Used to convert a defect to Containment or back.

BUSINESS RULES

- It's allowed to convert a defect from type **Adhoc, Containment, Inspection Operation, Verification Operation** to **Containment**, or vice versa.
- Converting a **containment exception** to **containment exception** is allowed for the purpose of allowing update to production advisory.

MODEL

- *DefectId* (int, **required**)
- *DefectType* (string (50), change the defect type (e.g., Adhoc -> Containment), **optional**)
- *ProductionAdvisoryId* (int, associate the defect to a production advisory, **optional**)

SAMPLE

Request:


```
{
   "DefectId": 120,
   "DefectType": "Containment",
   "ProductionAdvisoryId": 24
}
```


Response:


```
{
   "DefectId": 120,
   "DefectType": "Containment",
   "BuildItemId": "c083b868-6900-4ca1-c135-08dd4d17ee3e",
   "SerialNumber": "00000402",
   "RouteName": "route 1",
   "StepName": "step 1 name",
   "StepDescription": "step 1 desc",
   "CurrentStatusType": "Initialized",
   "ProductionAdvisoryId": 24,
   "ProductionAdvisory": {
      "ProductionAdvisoryId": 24,
      "ProductionAdvisoryType": "Containment",
      "ExternalState": "Install",
      "ExternalId": "123",
      "UserName": "admin",
      "PartNumber": "PartTwo",
      "Quantity": 3,
      "DetailsJson": "{'AreaResponsible': 'Area 51',  'ExternalUrl': 'https://topsecret-location.com'}"
   },
   "DefectStatuses": [
      {
         "DefectId": 120,
         "DefectStatusType": "Initialized",
         "Date": "2025-05-09T10:35:56.856-04:00",
         "StationId": 1049,
         "StationName": "station 1",
         "Reason": "B4",
         "ReasonGroup": "Reason Group B"
      }
   ],
   "DefectNotes": [
      {
         "DefectId": 120,
         "Date": "2025-05-09T13:48:06.973-04:00",
         "Note": "sd"
      }
   ]
}
```

Production/Quality/Issues/{defectId}/Copy **POST**

Creates a new defect-based issue as a copy of an existing one.

BUSINESS RULES

- Will only copy the first Initialized status.
- Will replace any provided property.

MODEL

- *SerialNumber* (string, **required**)
- *PartNumber* (string, **required** if the serial number is ambiguous)
- *LineName* (string, **required** if the serial number is ambiguous)
- *RouteName* (string, **required** if the serial number is ambiguous)
- *DefectStatuses* (*ExternalDefectStatus*[], used to change the user, **optional**)
- *ProductionAdvisory* (*ExternalProductionAdvisory*, used to change shortage quantity, **optional**)

EXTERNAL DEFECT STATUS

- *UserName* (string, change user for the copied defect, **optional**)

EXTERNAL PRODUCTION ADVISORY

- *Quantity* (int, change shortage quantity for this copy, **optional**)

SAMPLE

Request: POST {{HostAddress}}/api/External/Production/Quality/Issues/3575/Copy


```
{
  "SerialNumber": "0000010",
  "RouteName": "Route 1",
  "DefectStatuses": [
    {
        "StationId": 1107,
        "UserName": "ExternalUser",
    }
  ]
}
```


Response:


```
{
  "DefectId": 3651,
  "BuildItemId": "6cb50c23-eda6-4352-7e19-08dd7761d879",
  "SerialNumber": "0000010",
  "PartNumber": "Simple Part",
  "RouteName": "Simple Route",
  "StationName": "Stn1",
  "CurrentStatusType": "Initialized",
  "CreatedUser": "External User",
  "CreatedDate": "2025-04-10T04:38:24.8906819-04:00",
  "DefectStatuses": [
    {
      "DefectId": 3651,
      "DefectStatusType": "Initialized",
      "UserName": "External User",
      "Date": "2025-04-10T04:38:24.8906819-04:00",
      "StationId": 1107,
      "StationName": "Stn1",
      "Reason": "Sticker Misalign",
      "ReasonGroup": "Fender Reasons"
    }
  ],
  "DefectNotes": []
}
```

Production/Quality/Issues/Status **POST**

Changes the status of a quality issue.

MODEL

- *DefectId* (int, **required**)
- *DefectStatusType* (string, **required**)
- *StationId* (int, **required**)
- *UserName* (string, **optional**)
- *Reason* (string (255), **optional**)
- *ReasonGroup* (string (255), **optional**)
- *ReasonDescription* (string (255), **optional**)

SAMPLE

Request:


```
{
    "DefectId": 3651,
    "DefectStatusType": "Resolved",
    "StationId": 1107,
    "Reason": "Reason 1",
    "ReasonGroup": "Acknowledge Reasons"
}
```


Response:


```
{
  "DefectId": 3651,
  "DefectStatusType": "Resolved",
  "Date": "2025-04-11T03:48:46.0849443-04:00",
  "StationId": 1107,
  "Reason": "Reason 1",
  "ReasonGroup": "Acknowledge Reasons"
}
```

Production/Quality/Issues/Note **POST**

Adds a note to a quality issue.

MODEL

- *DefectId* (int, **required**)
- *Note* (string (2000), **required**)
- *UserName* (string (255), **optional**)

SAMPLE

Request:

```
{
    "DefectId": 3651,
    "UserName": "ExternalUser",
    "Note": "Note 1"
}
```

Response:

```
{
  "DefectId": 3651,
  "UserName": "ExternalUser",
  "Note": "Note 1"
}
```

Superseded Endpoints

Superseded Endpoints are endpoints which still exist in the API for backwards compatibility. They should not be used to develop new integrations.

Configuration/ValidateProducts **POST** (Superseded)

Submissions sent to this endpoint are handled by **Configuration/ValidateParts**

The Products field in the JSON body will be interpreted as Parts.

Configuration/ValidateStationProducts **POST** (Superseded)

Submissions sent to this endpoint are handled by **Configuration/ValidateStationParts**

The Products field in the JSON body will be interpreted as Parts. The StationProducts field in the JSON body will be interpreted as StationParts.

Production/EnableProduct **POST** (Superseded)

Submissions sent to this endpoint are handled by **Production/EnablePart**

The Products field in the JSON body will be interpreted as Parts.

Production/DisableProduct **POST** (Superseded)

Submissions sent to this endpoint are handled by **Production/DisablePart**

The Products field in the JSON body will be interpreted as Parts.

Production/DeleteWorkOrder **POST** (Superseded)

This endpoint is superseded by **Production/WorkOrders**. This endpoint uses a Post command, while the endpoint which supersedes it uses the Delete Verb and does not require JSON Contents.

This API call is used to delete a work order by using a work order number.

CALL DETAILS

- Request body content passed in contains work order number as a string.
- A successful call deletes the work order.
- An unsuccessful call will yield a boolean value indicating a successful or unsuccessful delete and an HTTP error code. Additional information as to what is returned is described in a later section.

BUSINESS RULES

- Only work orders created by PINpoint API or the CSV work order import process (available on the PINpoint Tracking Web application) may be deleted by PINpoint API.
- Work orders may only be deleted if they have not been started.

SAMPLE

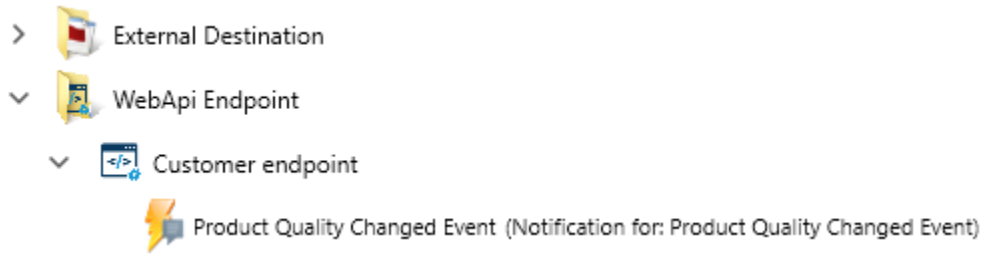
```
"AC912001"
```

Integration Endpoints

PINpoint can be configured to make various outbound HTTP requests to a customer's service.

EVENT NOTIFY

When Andon Events are configured for outbound web API notification, PINpoint will call the endpoint and send information about the associated events using the following request details.



CALL DETAILS

- HTTP POST.
- Version 1.1
- A **Content-Type** of **application/json** (utf-8)
- An **Accept** of **application/json**
- **Accept-Encoding** of **gzip**

METHOD SIGNATURE

- The consumer must implement an HTTP POST method which accepts a **NameValuePair** collection.

```
Example
[HttpPost]
[ResponseType(typeof(bool))]
public async Task<IHttpActionResult> State(FormDataCollection payload)
```

PAYLOAD

FormDataCollection must be a collection of **KeyValuePair**<**string**,**string**>

- The Key is the Andon event property name, and Value is the respective value.
- All Andon events have the same list of properties.
- Types must be inferred/cast, but can safely assume to be standard types (Guid, String, Int, DateTime)
- DateTimes are in UTC.

EXAMPLE COLLECTION:

```
{[BuildItemId, 0439d9ad-5032-4a00-ae3b-47443a60f126]}
{[Colour, #FF000000]}
{[Description, PpAdmin PpAdmin]}
{[EventId, 1108]}
{[Count, 2450]}
{[CreatedDate, 2020-08-18T18:06:38]}
{[OriginalState, False]}
{[State, True]}
{[EventName, User Login Event]}
{[IsTransitory, True]}
{[LastUpdated, 2022-02-13T03:54:41]}
{[LineId, null]}
{[ReasonItemId, null]}
{[ReasonItemRelationId, null]}
{[SerialNumber, null]}
{[StateChangeDate, 2022-02-13T03:54:41]}
{[StationId, 1134]}
```

{[StationName, Station1]}
{[UserId, null]}
{[OperatorDisplayName,]}

Support

For any questions contact: ***techsupport@pinpointinfo.com***