



Turbonomic 8.4.1 REST API Changes

December 14, 2021

This document describes issues that result in changes to the REST API. It is a running list of changes that accumulates these changes until the next release of the API Programming Guide. When we release a new version of the programming guide, it will include these changes. At that time, the list will start with new change to the API.

Because the API is a full reflection of the Turbonomic product, some changes to the product necessarily cause changes in the API. This can include bug fixes and improvements to the product. For example, we might add new commodities that you can get for a given entity type. Or a bug fix might change the data that is returned by an API request. We strive to make these changes backward-compatible whenever possible.

REST API Changes

- **Improvement:**

Fix Version: 8.3.1

For Dynatrace, you can now enable or disable collection of VM metrics via the Dynatrace target.

When managing applications, analysis takes VM metrics into account. In a Dynatrace environment, Turbonomic can discover VM data from the Dynatrace target and also from the underlying VMs. If it collects VM data from both sources, analysis will use the Dynatrace data.

For most environments it is best to use the VM data that Turbonomic collects directly from the VMs themselves. The data can be richer and more accurate, and discovery of this data through Dynatrace can experience unsatisfactory performance.

To give you control over the source of VM metric data that analysis will use, this release introduces the **Collect Virtual Machine Metrics** setting when you configure a Dynatrace target. By default, this setting is **Off**. In that case, Turbonomic does not collect VM metrics via the Dynatrace target, and analysis uses data collected from the VMs themselves.

NOTE:

If you turn this setting **On** for the target, then analysis does not use the percentiles that you configure in the **Aggressiveness** settings for the affected VMs.

When you configure a Dynatrace via the API, you can now use the optional setting:

```
{
  "displayName": "Collect Virtual Machine Metrics",
  "name": "collectVmMetrics",
  "value": "false",
  "defaultValue": "false",
  "isMandatory": false,
  "isSecret": false,
  "isMultiline": false,
  "isTargetDisplayName": false,
  "valueType": "BOOLEAN",
  "description": "Overwrite Hypervisor or Cloud Provider Virtual Machine
metrics with data from the target",
  "verificationRegex": "(true|false)"
}
```

- **Improvement:**

Fix Version: 8.3.1

This release includes the Minimum Wasted File Size setting for Wasted File policies.

When it discovers wasted files, Turbonomic identifies a given storage, plus the list of wasted files on it. In some circumstances, if the volume has many files, the data to describe the volume is too large to transport from the target back to Turbonomic. In that case, analysis can fail to include the affected storage, and the log can post an error that includes the string (where * is a wildcard):

```
BroadcastStage.*Entity.*failed to be broadcast because it's too large
```

To reduce the data payload for that storage, you can direct wasted storage browsing to skip over files of a minimum size. Then the data only includes larger files in its list, which can reduce the overall payload. The default minimum is 1 MB.

If you continue to see this problem, you can increase the size of wasted files to add into the list. That should reduce the number of wasted files on a volume. To adjust the **Minimum Wasted Files Size** setting:

1. Configure a Storage Policy.
 Navigate to **Settings** and choose **Policies**. Then edit or create a new Storage automation policy.
2. Edit the Minimum Wasted Files Size setting.

In the **Storage Settings** section, edit the value for **Minimum Wasted Files Size**.

When editing policies via the API, you can find this setting in the storage settings manager (`...api/v3/settings/specs?manager_uuid=storagesettingsmanager`). The setting is defined as:

```
{
  "uuid": "minWastedFilesSize",
  "displayName": "Minimum Wasted Files Size [KB]",
  "defaultValue": "1000.0",
  "valueType": "NUMERIC",
}
```

```

        "valueObjectType": "String",
        "min": 0,
        "max": 1000000000,
        "entityType": "Storage"
    }

```

- **Fixed Issue:**

Fix Version: 8.3.6

Customer Issue 116983

When you execute a POST stats/uuid call with no groupBy setting, the API sometimes returns grouped stats.

For a POST: `.../stats/{statistic_Uuid}` call to the API, you can specify `groupBy` parameters for the statistics you want to get. If you do not provide any `groupBy` values, the API should return ungrouped stats records. However, under some circumstances it does return grouped records. The API should be consistent.

- **Fixed Issue:**

Fix Version: 8.3.3

Customer Issue 116940

Executing a GET on the businessunits endpoint can return data even if there are no businessunit entities in the given scope.

Executing a GET on the businessunits endpoint can return data even if there are no businessunit entities in the given scope. The data it returns is an array with a single element in it. For example:

```
[{"master": false, "riSupported": false, "hasRelatedTarget": false}]
```

The proper response for this circumstance should be an empty array.

API Deprecation Contract

The following endpoints, requests, DTOs, and/or parameters have been deprecated in the Turbonomic API. The End of Life (EOL) gives the product version in which we will finalize these changes, and will no longer offer the deprecated items.

NOTE:

Instructions for alternative usage are given for deprecated items with a defined EOL.

EOL: 8.4.1

We will make the following deprecation changes in the 8.4.1 release. These deprecated features will be removed from the API at that time.

Endpoints	None
-----------	------

Requests	None
Request Parameters	<ul style="list-style-type: none"> Request: GET /search Parameter Name: regex Instruction: Use the queryType parameter.
DTOs	<ul style="list-style-type: none"> DTO: ScenarioChangeApiDTO Instruction: Use the loadChanges or configChanges parameters in the ScenarioApiDTO. DTO: StatPercentileApiDTO Instruction: Use StatHistUtilizationApiDTO.
DTO Parameters	<ul style="list-style-type: none"> DTO: ActionApiDTO Parameter Name: resizeToValue Instruction: Use the newValue parameter. DTO: ActionScopesApiDTO Parameter Name: limitEntities Instruction: Use the limit parameter. DTO: BillingFamilyApiDTO Parameter Name: uuidToNameMap Instruction: Use the businessUnitApiDTOList parameter. DTO: BusinessUnitApiDTO Parameter Name: hasRelatedTarget Instruction: Use the associatedTargetId parameter. DTO: ScenarioApiDTO Parameter Name: changes Instruction: Use the loadChanges or configChanges parameters. DTO: ScopeUuidsApiInputDTO Parameter Name: topologyContextId Instruction: Use the marketId parameter. DTO: StatApiDTO Parameter Name: percentile Instruction: Use the histUtilizations parameter. DTO: TargetApiDTO Parameter Name: currentBusinessAccount Instruction: Use the associatedTargetId parameter of the BusinessUnitApiDTO.

EOL: 8.5.1

Behavior Changes:

- Pagination of Returned Data

The default pagination will change for the following requests:

- GET /groups
- GET /groups/<groupId>/members
- GET /groups/<groupId>/entities
- GET /targets/<targetId>/entities set
- GET /tags
- GET /tags/{tagKey}/entities

To paginate returned data, the `limit`, `orderBy`, and `cursor` parameters control how the APIs paginate their data. Currently, if you do not specify a `limit` for the listed API requests, the API returns an unlimited set of data.

Starting with version 8.5.1, if you do not specify a `limit` for these requests, the API will paginate the data to a default limit. If your scripts make these requests without setting a limit, you should update them to check for a pagination cursor in the returned data.

We will make the following deprecation changes in the 8.5.1 release. These deprecated features will be removed from the API at that time.

Endpoints	None
Requests	<ul style="list-style-type: none"> • Request: GET /users/onetimepassword <p>Instruction: Deprecated with no replacement. This is not required for any Turbonomic activities.</p>
Request Parameters	None

DTOs	None
<p>DTO Parameters</p>	<ul style="list-style-type: none"> • DTO: ActionApiDTO Parameter Names: <ul style="list-style-type: none"> ◦ <code>currentLocation</code> ◦ <code>newLocation</code> Instruction: An action response no longer returns the following data to these parameters: <ul style="list-style-type: none"> ◦ <code>priceIndex</code> ◦ <code>state</code> ◦ <code>severity</code> ◦ <code>unplacementDetails</code> ◦ <code>costPrice</code> ◦ <code>severityBreakdown</code> ◦ <code>realtimeMarketReference</code> ◦ <code>consumers</code> ◦ <code>providers</code> ◦ <code>template</code> ◦ <code>numRelatedVMs</code> ◦ <code>connectedEntities</code> ◦ <code>aspects</code> ◦ <code>tags</code> ◦ <code>onDemandBeforeRate</code> ◦ <code>onDemandAfterRate</code> If you require these data, you can find the associated fields in <code>ActionApiDTO.currentEntity</code> and <code>ActionApiDTO.newEntity</code>. • DTO: BusinessUnitPriceAdjustmentApiDTO Parameter Name: <code>serviceDiscounts</code> Instruction: Use the <code>servicePriceAdjustments</code> parameter. • DTO: BusinessUnitApiDTO Parameter Name: <code>discount</code> Instruction: Use the <code>priceAdjustment</code> parameter.

	<ul style="list-style-type: none"> • DTO: CloudServicePriceAdjustmentApiDTO Parameter Name: discount Instruction: Use the priceAdjustment parameter. • DTO: CloudServicePricingModelApiDTO Parameter Name: discount Instruction: Use the priceAdjustment parameter. • DTO: TemplatePriceAdjustmentDTO Parameter Name: discount Instruction: Use the priceAdjustment parameter.
--	---

EOL: 8.6.1

Behavior Changes:

- Pagination of Returned Data from /search requests:

For searches, the API paginates the returned data. If you do not specify a pagination limit, then the API returns pages set to the default limit. (This behavior was normalized all searches starting with version 8.5.1). Starting with version 8.6.1, the default and max pagination limits for search data will be 500. You can set pagination to less than 500, but if your search results include more than 500 entries, the results will be paginated. Your scripts should always check for a pagination cursor when processing search results.

We will make the following deprecation changes in the 8.6.1 release. These deprecated features will be removed from the API at that time.

Endpoints	None
Requests	<ul style="list-style-type: none"> • None

Request Parameters	None
DTOs	<ul style="list-style-type: none"> ReportScheduleApiDTO <p>This release will remove ReportScheduleApiDTO.</p>
DTO Parameters	<ul style="list-style-type: none"> <p>DTO: ActionApiDTO</p> <p>Parameter Name: CurrentLocation</p> <p>Instruction: The object in this parameter will only contain the following fields:</p> <ul style="list-style-type: none"> links uuid className displayName discoveredBy environmentType vendorIDs <p>DTO: ActionApiDTO</p> <p>Parameter Name: NewLocation</p> <p>Instruction: The object in this parameter will only contain the following fields:</p> <ul style="list-style-type: none"> links uuid className displayName discoveredBy environmentType vendorIDs <p>DTO: ProductVersionDTO</p> <p>Parameter Name: buildUser</p> <p>Instruction: Deprecated with no replacement. This parameter was not set in the 8.x product family.</p> <p>DTO: GroupApiDTO</p> <p>Parameter Name: remoteId</p> <p>Instruction: Deprecated with no replacement. This parameter was not set in the 8.x product family.</p>

- DTO: UserApiDTO

Parameter Names:

- `roleUuid`
- `roleName`

Instruction: Deprecated. Use the `roles` parameter in UserApiDTO.