Use the Database Visibility API to Configure Collectors

On this page:

Related pages:

- Add Database Collectors
- Configure the Database Agent to Monitor Server Hardware
- AppDynamics APIs

The Database Visibility API allows you to get, create, update, and delete Database Visibility Collectors.

Include the following headers for all Database Visibility API requests:

```
Accept: application/json; Content-type: application/json
```

JSON is currently the only supported format.

Supported API Calls

Get all Collectors

```
GET /controller/rest/databases/collectors
```

Get a Specific Collector

```
GET /controller/rest/databases/collectors/{configurationId}
```

Create a Collector

```
POST /controller/rest/databases/collectors/create
```

The JSON you send must contain the relevant Collector information. The required fields describing the Collector vary based on the type of database. Below is a sample JSON request body:

```
{ "type": "MYSQL", "name": "cart", "hostname": "localhost", "port": "3306", "username": "root",
"password": "singcontroller", "agentName": "Default Database Agent", "enabled": "true" }
```

For more information, see the table in the "UI Collector versus JSON Collector Configuration Field Names" section that follows.

Update a Collector

1. Make a GET request for the collector that you want to update.
2. Copy the JSON response body that is returned by the GET request to a text editor, and modify the fields that you want to update.
3. Make a POST request for the collector that you want to update, and include the updated JSON.

```
POST /controller/rest/databases/collectors/update
```

Below is a sample JSON request body:

```
{ "type": "MYSQL", "name": "cart", "hostname": "localhost", "port": "3306", "username":
"root", "password": "singcontroller", "agentName": "Default Database Agent", "enabled":
"true" }
```

The JSON you send must contain all the details of the existing collector with only the fields that you want to modify changed. To ensure you have all the fields, use the Get a Specific Collector call.

Delete a Specific Collector

```
DELETE /controller/rest/databases/collectors/{configurationId}
```

Below is an example of a delete request.

```
DELETE /controller/rest/databases/collectors/{1}
```

Batch Delete Multiple Collectors

```
POST /controller/rest/databases/collectors/batchDelete
```

Send an array of the configuration Ids of the Collectors.

Below is an example of a batch delete command.

```
curl --user user1@customer1:password -H "Accept: application/json" -H "Content-type: application
/json" -X POST -d “[1,2,3]” http://demo.appdynamics.com/controller/rest/databases/collectors
/batchDelete
```

Example Request and Response

Example Request

```
curl --user user1@customer1:password http://demo.appdynamics.com/controller/rest/databases
/collectors
```

Example Response

```
[
 {
   "performanceState": null,
   "collectorStatus": "COLLECTING_DATA",
   "eventSummary": null,
   "config": {
     "id": 1,
     "version": 0,
     "name": "test",
     "nameUnique": true,
     "builtIn": false,
     "createdBy": "user1",
     "createdOn": 1453317194781,
     "modifiedBy": "user1",
     "modifiedOn": 1453317194781,
     "type": "MYSQL",
     "hostname": "localhost",
     "useWindowsAuth": false,
     "username": "root",
     "password": "appdynamics_redacted_password",
     "port": 8080,
     "loggingEnabled": false,
     "databaseName": null,
     "failoverPartner": null,
     "connectAsSysdba": false,
     "useServiceName": false,
     "sid": null,
     "customConnectionString": null,
     "enterpriseDB": false,
     "useSSL": false,
     "enableOSMonitor": false,
     "hostOS": null,
     "useLocalWMI": false,
     "hostDomain": null,
     "hostUsername": null,
     "hostPassword": "",
     "certificateAuth": false,
     "removeLiterals": true,
     "sshPort": 0,
     "agentName": "Default Database Agent"
   }
 }
]
```


UI Collector versus JSON Collector Configuration Field Names

Use the table below to ensure you use the correct field names for your API calls. The Collector configuration field names are described in Configure the Database Agent to Monitor Server Hardware and Add Database Collectors.

Section	UI Collector Configuration Field Name	JSON Collector Configuration Field Name
		id (AppDynamics assigns this ID to the Collector when you configure the Collector. You need this ID when doing a batch delete.)
	Database Type	type
	Database Agent	agentName

	Database	name
Connection Details	Hostname/IP Address	hostname
	EnterpriseDB	enterpriseDB
	Failover Partner	failoverPartner
	Listener Port	port
	Custom JDBC Connection String	customConnectionString
	Use Service Name	useServiceName
	SID or SERVICE_NAME	sid
	Connect as a sysdba	connectAsSysdba
	Username	username
	Password	password
	Logging Enabled	
Hardware Monitoring	Monitor Operating System	enableOSMonitor
	Operating System	hostOS
	Use Local WMI	useLocalWMI
	Domain	hostDomain
	SSH Port	sshPort
	Use certificate	certificateAuth
	Username	hostUsername
	Password	hostPassword

SSL field

In addition to JSON Configuration Fields listed above, there is also the ssl field. SSL is a configurable property for the Database Agent. If the Database Agent has been configured to use SSL, then you must also provide the ssl field and its value in your Database Visibility API calls.