How do I add or update a NetScaler resource seamlessly, using a single NITRO API call?

Many administrators use NITRO API scripts to automate configuration execution on their NetScaler appliances. In large deployments, they have thousands of load balancing virtual servers and other NetScaler entities to configure. One step they often have to perform is to check whether the entity currently exists. If it does, they modify it. If it doesn't exist, they create a new entity. These checks are needed to ensure that automated scripts progress smoothly, without errors caused by using the wrong method. While these checks serve a purpose, they also make scripts lengthy and time consuming.

To help simplify management operations and optimize the length of automation scripts, administrators needed a single API call that could be used be for both operations: add and update. Beginning with NetScaler release 11.1, a new API parameter addresses this need.

This new query parameter is called ***idempotent***. When enabled in a POST (create) request, it directs NITRO to perform a POST/create operation if the resource doesn't exist, or an update operation if the resource does exist.

This parameter is set in the URL of the POST request. Valid values are NO (the default) and YES.

Important points to remember:

- Applicable only to ADD (POST) operation.
- Parameters that are not included in the payload of the request are not reset to their default values. For example, if an existing lbvserver named "V1" has "lbmethod" set to "roundrobin," an idempotent request to ADD an lbvserver with name "V1" but without a value for "lbmethod" does not change "lbmethod" to its default value of "leastconnection."
- If this parameter is not included in the URL, no behavior is changed (backward compatibility is maintained).
- If the ADD (POST) request has arguments that are not part of an ADD (POST) command by definition but are part of SET (update) command by definition, then situation this is handled intelligently by internally generating one create request and one update request.

Example: Creating an lbvserver

- URL: http://10.102.216.119/nitro/v1/config/lbvserver?idempotent=yes
- HTTP Method: POST
- Request Headers
  - Content-Type: application/json
  - X-NITRO-ONERROR: exit
- Payload:
  {"lbvserver":{"name":"lbv1","servicetype":"http","lbmethod":"roundrobin"}}

Example: Updating an lbvserver can also happen using POST request itself. Internally NITRO does an update (PUT) operation as the entity already exists in this case.

- URL: http://10.102.216.119/nitro/v1/config/lbvserver?idempotent=yes
- HTTP Method: POST
- Request Headers
  - Content-Type: application/json
  - X-NITRO-ONERROR: exit
- Payload:
  {"lbvserver":{"name":"lbv1","servicetype":"http","lbmethod":"leastconnection"}}