

# [Reverse API](#)

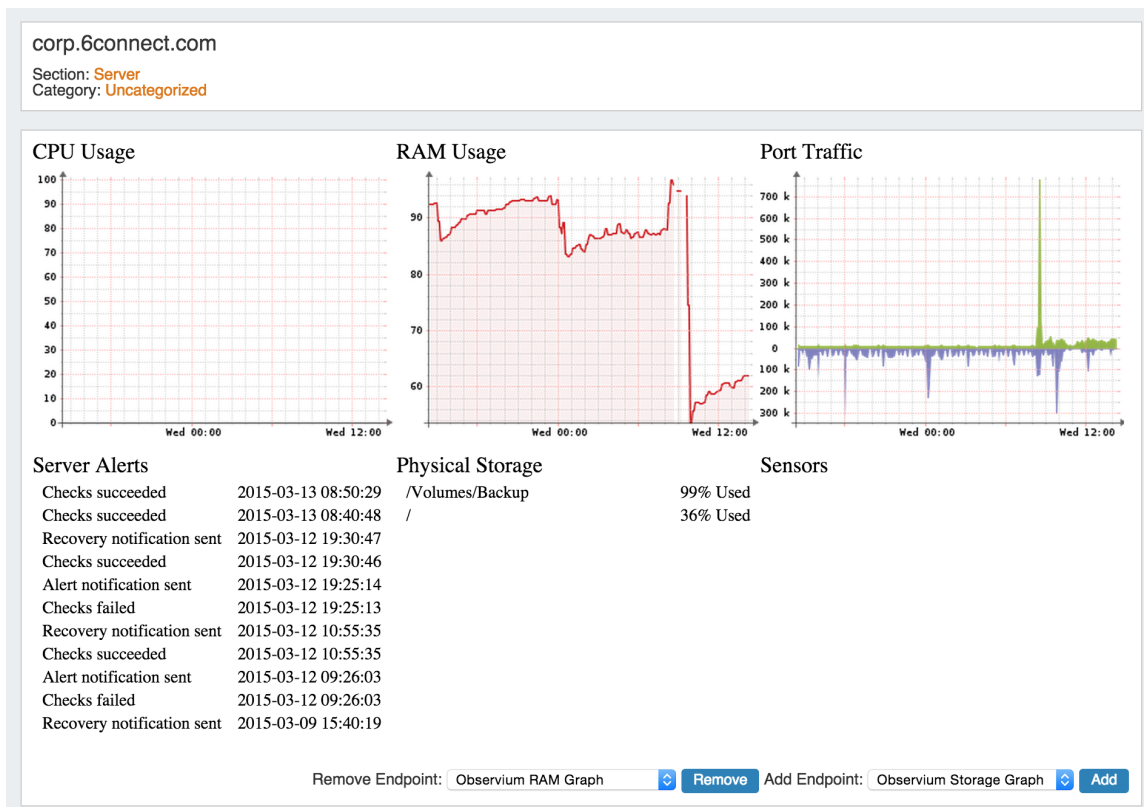
## [Reverse API](#)

- Reverse API
  - Overview
  - API Call Formatting
  - Reverse API Detail:

## [Overview](#)

### Reverse API Tools - Beta

ProVision's Reverse API calls and UI elements allow for integration with outside APIs to improve workflow and create custom display content. In the ProVision user interface, the [Reverse API](#) page allows for endpoints to be built and provides a text editor to create presentation JavaScript commands. This JavaScript presentation code is then displayed in the [Reverse API Gadget](#).



Using the ProVision Reverse API (rAPI), you can perform these same actions and customize to meet your specific needs outside of the ProVision UI.

## [API Call Formatting](#)

Reverse API (rAPI) calls are made to hit a user-defined URL or command line program. If the rAPI type is set to 'external,' the system makes a HTTP request. If the rAPI type is set to 'local,' it executes the call on the command line of the local machine. In both cases the call is first customized with the attributes of the resource supplied to the execute function. This allows a single rAPI endpoint to serve a wide array of individual resources, fetching only information relevant to that particular resource without having to store anything locally.

rAPI calls are formatted thusly:

```
http://observium.tcp0.com/graph.php?
height=200&width=265&type=device_bits&legend=no&username=api&password=password&device={observium-id}
```

The interesting part about this URL is the bit in curly-braces: {observium-id}. When this call is made, the system decodes the URL by searching for everything within curly-braces and replaces it with data pulled from a resource. A rAPI call may have as many or as a few curly-brace replacement targets as is needed. Each curly-brace target will be replaced with exactly one resource attribute. All targets must be successfully replace for the command to succeed.

For example, when this rAPI call is decoded with information from the 1-dev resource, the decoded call is as follows:

```
http://observium.tcp0.com/graph.php?
height=200&width=265&type=device_bits&legend=no&username=api&password=password&device=21
```

The rAPI service then pulls the data from that URL, pairs it with its presentation code, and returns it via the execute endpoint. If a user attempted to execute the above call on a resource which did not have the "observium-id" property, the execute endpoint would return the following error: "Reverse API Call references token 'observium-id', which does not exist in resource."



All curly-brace calls reference a resource attribute by their unique resource slugs. The list of available resource attributes can be customized using the existing resource system endpoints.



Curly-brace targets may take the form {parent.observium-id}. The "parent." prefix indicates that in decoding this particular brace, the attributes of the resource's parent are to be used.

## [Reverse API Detail:](#)

For detail on the Reverse API calls and parameters, proceed to [Reverse API - Detail](#).