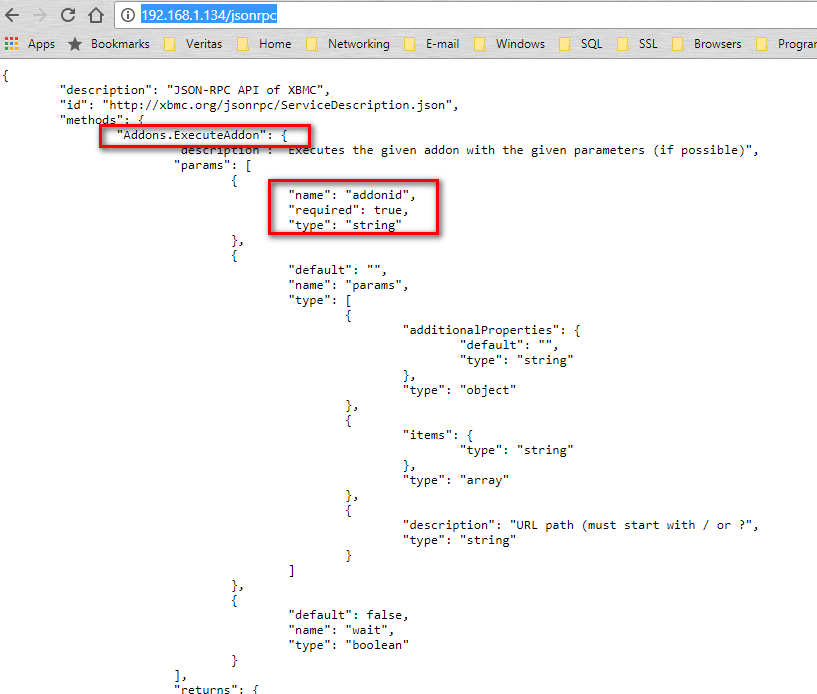
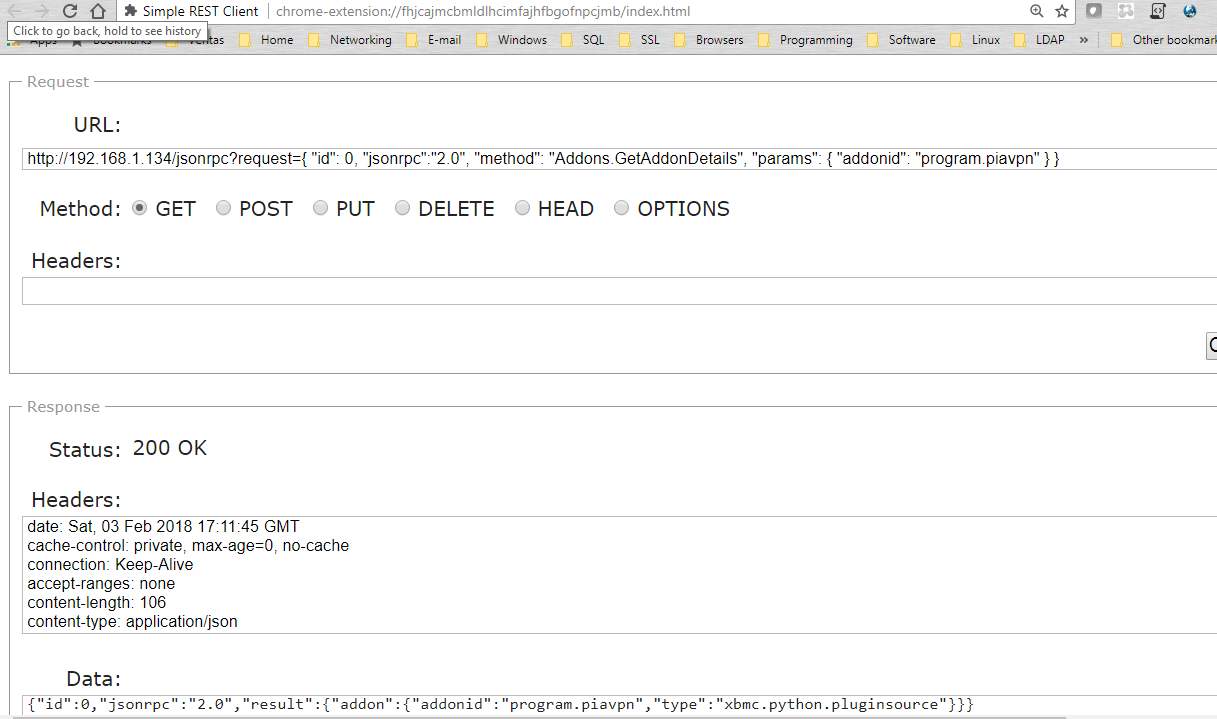
**Kodi JSON-RPC**

Kodi has a useful built-in web server, see: <https://kodi.wiki/view/Webserver>

Go to [**http://192.168.1.134/jsonrpc**](http://192.168.1.134/jsonrpc) and you will see the functions in the Kodi JSON-RPC API:  
  
  
See also:  
<https://kodi.wiki/view/JSON-RPC_API/v6>  
<https://forum.kodi.tv/forumdisplay.php?fid=174>

**Note:** I had initially thought that I could, for example, use the “Addons.ExecuteAddon” API function to generate an XML file that I could then point the Kodi RSS Manager at. However, it turns out that this API cannot return any text, so I had to find another way to achieve this; I used a cron job to call a Python script to generate an RSS XML file on the filesystem, which I then pointed Kodi to using <file:///localhost/storage/MyGeneratedRSSFeed.xml>.  
  
Anyway, back to JSON-RPC. Here’s an example calling “Addons.GetAddonDetails”. We only need to pass an “addonid” parameter, which has to match the value in the addon’s addon.xml file. To test this, we could simply enter the GET request in the browser, but here we are using the “Simple REST Client”, as mentioned in the “Debugging” section of <https://kodi.wiki/view/JSON-RPC_API>:  
  
http://192.168.1.134/jsonrpc?request={ "id": 0, "jsonrpc":"2.0", "method": "Addons.GetAddonDetails", "params": { "addonid": "program.piavpn" } }  
  
Note: The “id” value comes back in the reply, so it is a way to verify that the reply was for this request.

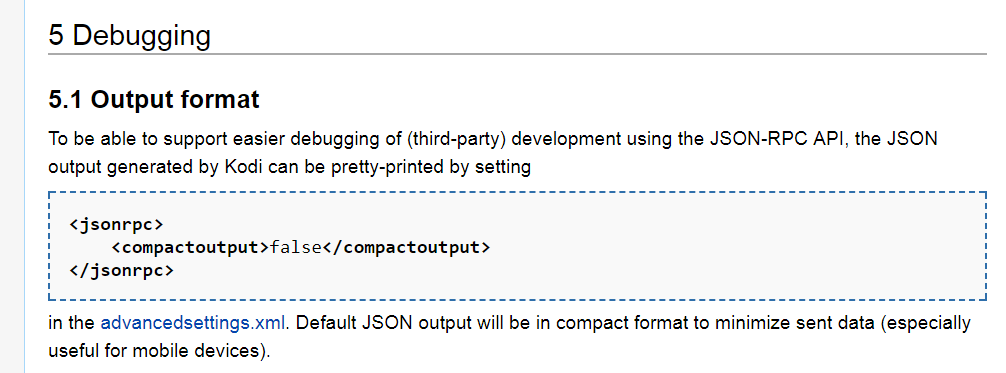


More JSON-RPC Examples:   
<https://kodi.wiki/view/JSON-RPC_API/Examples>

<https://forum.kodi.tv/showthread.php?tid=157996>

And using curl: <https://github.com/trakt/script.trakt/issues/192>

**Note:** From <https://kodi.wiki/view/JSON-RPC_API>:



**Using “Addons.ExecuteAddon” to invoke my own addon**

As mentioned above, I had hoped to use the API function to return RSS XML for the Kodi ticker. However, this appears to only ever return JSON output, plus I could not get it to return any text at all (see <https://forum.kodi.tv/showthread.php?tid=225540>).

Here’s an example calling a script addon that I created:  
  
http://192.168.1.134/jsonrpc?request={ "id": 0, "jsonrpc":"2.0", "method": "Addons.ExecuteAddon", "params": { "addonid": "script.myscripts" } }  
  
  
In addition to using this to return RSS XML, I had hoped that I could pass arguments in my GET request to invoke specific custom functions I had implemented in this addon. I can see that this should work, so long as you don’t need to return anything back to the caller!

I didn’t really get far with the script addon, but it \*was\* working. Here is the addon.xml file I used:  
  
/storage/.kodi/addons/script.myscripts/addon.xml:

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<addon id="script.myscripts" name="My Scripts" version="1.0.0" provider-name="JeremyC">

<requires>

<import addon="xbmc.python" version="2.1.0"/>

</requires>

<extension point="xbmc.python.script" library="default.py" />

<extension point="xbmc.addon.metadata">

<summary lang="en\_GB">Execute my own scripts and functions</summary>

<description lang="en\_GB">

The purpose for this addon is to provide a way to call my own custom scripts and functions via the Kodi JSON-RPC API function Addons. ExecuteAddon

</description>

<language></language>

<platform>all</platform>

</extension>

</addon>

I can still see potential uses for this addon as a way to deliver scripts and Python modules to be used by other addons, or to be invoked remotely using the Kodi built-in web server, via wget or curl.   
  
*JeremyC 03-02-2018***END**