KEMP PS wrapper

All the cmdlets inside the KEMP PS wrapper, except the `Test-LmServerConnection`[1] cmdlet, return a PS object with the following structure:

- ReturnCode (integer)
- Response (string)
- Data (PS object, if any)

The ReturnCode field possible values are:
- 200: the command has been completed with success
- 4xx/500: the command ended with an error. The error code depends on the error type.

The Response field possible values are:
- Command successfully executed.
- Description of the error when the command fails. Example: `Unknown parameter value lmversion`.

The field Data contains the response if any. The structure of this field depends from the command. The elements of the this field can be accessed using the "***dot***" notation. This field is empty if the commands fails.

Example 1: Get the version of the firmware installed into the LoadMaster.

```
$lma = Get-LmParameter -Param version -LoadBalancer 172.21.59.85 -SubjectCN user1
$lma | Format-List

ReturnCode : 200
Response   : Command successfully executed.
Data       : @{version=7.2.39.0}

$lma.Data.version
7.2.39.0
```

Example 2: Get the available licenses for a specific order id.

```
$licDet = Get-LicenseType -KempId gigio@gigetto.com -Password supersecretpasswd `
                          -LoadBalancer 172.21.59.85 -Credential $lm85BalCreds -OrderId gigio20170517
$licDet | Format-List

ReturnCode : 200
Response   : Command successfully executed.
Data       : @{License=System.Object[]}

$licDet.Data.License

id            : 0632b88b577c71591798268bcd4e01132f082309
name          : VLM-5000 ESP GEO with Basic 2 Years
available     : 1
description   : VLM-5000 ESP GEO with Basic 2 Years
tethered      : False
LicenseStatus : Permanent License
BuyMoreAt     : https://www.kemptechnologies.com/buy-me-now?KEMPID=gigio@gigetto.com

id            : fc488d991cffb7a5958625427d6bfb0b3edc008e
name          : VLM-5000 WAF GEO with Basic 3 Years
available     : 1
description   : VLM-5000 WAF GEO with Basic 3 Years
tethered      : False
LicenseStatus : Permanent License
BuyMoreAt     : https://www.kemptechnologies.com/buy-me-now?KEMPID=gigio@gigetto.com

name          : VLM-5000 with Enterprise Plus subscription
available     : 1
tethered      : 0
id            : 3eb92178611573946b422cf8d0df69d04c07fede
LicenseStatus : Temp License
description   : VLM-5000 with Enterprise Plus subscription
BuyMoreAt     : https://www.kemptechnologies.com/buy-me-now?KEMPID=gigio@gigetto.com
```

---

[1] The cmdlet `Test–LmServerConnection` returns True if the LoadMaster is reachable by the API, False if not.

In the above example the $licDet.Data.License is an array and each single element of the array can be access using the "[]" notation. For example to access the field *name* of the second element of the previous array we have to use the following well known notation: $licDet.Data.License[1].name. The index of the array starts from 0. The NULL object is returned if we try to access a not existing element.

The benefits of the this structure of the cmdlets answer are:
- Easy check for success/error (ReturnCode)
- Short description (Response)
- Data field (PS object on success/null on error)

ERRORS

If the error is a functional one, like wrong credentials, wrong parameter value, wrong load master ip address, the cmdlet returns a PS object as described with a ReturnCode containing the code of the error and with Response containing the description of the error that has been occurred.

Example: Try to get the firmware version installed into the LoadMaster using an invalid certificate (installed in the Windows machine but not belonging to any user inside the LoadMaster).

```
$lma = Get-LmParameter -Param version -LoadBalancer 172.21.59.85 -SubjectCN user1
$lma

ReturnCode Response                                                      Data
---------- --------                                                      ----
       401 The remote server returned an error: (401) Unauthorized.
```

If the error is due to a wrong/missing mandatory input, the cmdlet throws an exception.

Example: Get the firmware version installed into the LoadMaster using a certificate not installed in the Windows machine.

```
$lma = Get-LmParameter -Param version -LoadBalancer 172.21.59.85 -SubjectCN invalidcertificate

ERROR: Can't find a certificate with "invalidcertificate" as CN in the default
Cert:\CurrentUser\My store.
At
C:\Users\user1\Work\Powershell\Kemp.LoadBalancer.Powershell\Kemp.LoadBalancer.Powershell.psm1:27
3 char:5
+                 Throw $errStr
+                 ~~~~~~~~~~~~~~
    + CategoryInfo          : OperationStopped: (ERROR: Can't fi...tUser\My store.:String) [],
RuntimeException
    + FullyQualifiedErrorId : ERROR: Can't find a certificate with "invalidcertificate" as CN in
the default Cert:\CurrentUser\My store.
```

Example: Get the firmware version installed into the LoadMaster without credentials/certificate.

```
$lma = Get-LmParameter -Param version -LoadBalancer 172.21.59.85

ERROR: login method param is empty. Credentials or SubjectCN must be specified.
At
C:\Users\user1\Work\Powershell\Kemp.LoadBalancer.Powershell\Kemp.LoadBalancer.Powershell.psm1:24
4 char:3
+         Throw $errStr
+         ~~~~~~~~~~~~~~
    + CategoryInfo          : OperationStopped: (ERROR: login me...t be specified.:String) [],
RuntimeException
    + FullyQualifiedErrorId : ERROR: login method param is empty. Credentials or SubjectCN must
be specified.
```