

LiteSpeed eService API Documentation

V1.1

If you would like to automate the purchasing of LiteSpeed licenses, you can integrate the LiteSpeed eService API with your own ordering system. It is very simple to implement — just a single url request — and you will still get the same email notifications you would by manually ordering.

For security reasons, Lite Speed Technologies, Inc. reserves the right to enable or disable this feature without notifying the other party.

(See appendix at end for the change log.)

Table of Contents

Procedure.....	3
Basic Input fields (GET or POST).....	3
Basic Output Fields.....	4
Actions.....	4
1. Ping: Test API Connection.....	4
2. New Order.....	5
3. Cancel Leased License.....	8
4. Release Registered License.....	10
5. Suspend/Unsuspend Leased License.....	12
6. Upgrade or Downgrade a License.....	15
7. Query: Check License Related Information.....	19
Appendix: Update History.....	22

Procedure

1. Contact LiteSpeed sales to get your account approved for API integration. With the API, you can use same login and password you already use for the online store. You can also request to limit your API access to only certain IPs.
2. Follow this documentation. Write a GET or POST method with all the input fields supplied. The result will be returned as an xml string.
3. We are happy to refund any licenses you purchase while testing the API. Just contact sales.

URL:

https://store.litespeedtech.com/reseller/LiteSpeed_eService.php

Basic Input fields (GET or POST)

Every request needs to contain the following 4 fields:

litespeed_store_login – LiteSpeed online store account login email.

litespeed_store_pass – Account login password.

eService_version – API protocol version. 1.1 is currently supported.

eService_action – Available values:

“Ping”: For testing connection and basic login check.

“Order”: New order.

“Cancel”: Cancel a leased license.

“ReleaseLicense”: Release a serial number from a registered server so it can be used on a new server.

“Suspend”: Temporarily suspend a license to force user to take action on issues such as nonpayment, policy violation, etc. License will stop working while suspended. The license, however, will still appear on invoices.

“Unsuspend”: Unsuspend a suspended license.

“Upgrade”: Upgrade or downgrade a license without changing lease period. (Available for monthly and yearly leases, not for owned licenses.)

“Query”: Check license-related info.

We will gradually add more functions based on demand.

Basic Output Fields

Output will be a single string in xml format.

All the values are wrapped within <LiteSpeed_eService>. 3 tags will always be present: <action>, <result>, and <message>.

```
<LiteSpeed_eService>
  <action>Ping</action>
  <result>error</result>
  <message>Invalid login</message>
</LiteSpeed_eService>
```

Actions

1. Ping: Test API Connection

Input fields (GET or POST):

litespeed_store_login – LiteSpeed online store account login email.

litespeed_store_pass – Account login password.

eService_version – 1.1

eService_action – [Ping](#)

Ping Output:

1. If any errors occur during this process, <result> will be “error” and <message> will contain a detailed description:

```
<LiteSpeed_eService>
  <action>Ping</action>
  <result>error</result>
  <message>Invalid password!</message>
```

```
</LiteSpeed_eService>
```

2. If your login ID and password are correct and the account is authorized to use eService API, you'll receive a success message. <api_version> is the latest version of eService API. We'll update the version number if there is a protocol change.

```
<LiteSpeed_eService>

  <action>Ping</action>

  <api_version>1.1</api_version>

  <result>success</result>

  <message>ping reply</message>

</LiteSpeed_eService>
```

2. New Order

Even with the eService API, you will receive a new order email and a serial number email, the same as if you ordered manually. The eService API only allows ordering one license at a time.

Input fields (GET or POST):

litespeed_store_login – LiteSpeed online store account login email.

litespeed_store_pass – Account login password.

eService_version – 1.1

eService_action – **Order**

order_product – Product type. Available values: "LSWS" or "LSLB".

order_cpu – What kind of license. Available values:

"1": 1-CPU license

"2": 2-CPU license

"4": 4-CPU license

"8": 8-CPU license

“V”: VPS license

“U”: Ultra-VPS license *(Available for LSWS 4.2.2 and up.)*

If <order_product> is “LSLB”, <order_cpu> is not required.

order_modules – Additional features added for an extra cost. *(Available for LSWS 4.2.8 and up.)* Use a comma-delimited string. Available values:

“cache”: Cache module can only be added to VPS, Ultra-VPS and 1-CPU licenses. If ordered for other types of license, this input will be ignored. *(Available for LSWS 4.2.8 and up.)*

order_period – Billing cycle. Available values: “monthly”, “yearly”, “owned”.

order_payment – Payment method. Available values:

“credit”: Use account credit. User can utilize “Add funds” function to pre-deposit money, which will show up as account credit.

“creditcard”: Use credit card to pay. The credit card is pre-defined in the account. If there is available credit in the account, credit will be applied first, even when the payment method is set to “creditcard”.

order_cvv – **(optional)** Credit card security code. Try not to set this field. Only if your bank requires this (meaning that the transaction will fail without it) should you then supply this field. CVV code is not stored in the system, so if you need to set it, you have to set this field every time. Other information from your credit card will be taken from your user account.

order_promocode – **(optional)** Promotional code. If you have a pre-assigned promotional code registered to your account, then you can set it here. Promotional codes are exclusive to each client. Almost all resellers, though, have automatic discounts set up at the invoice level, and thus do not need a promotional code.

New Order Output:

1. If any errors occur during this process, <result> will contain “error” and <message> will contain a detailed description:

```
<LiteSpeed_eService>
  <action>Order</action>
  <result>error</result>
  <message>V1.o.1: Invalid field order_cpu - 9</message>
</LiteSpeed_eService>
```

2. If the transaction cannot be completed, <result> will be "incomplete". For example, if payment method is "credit", but there is not enough credit in your account, or if payment method is "creditcard", but the charge cannot go through, then the transaction will not be completed and <result> will display "incomplete". <license_id> and <invoice_id> will be provided. You will need to login to LiteSpeed's online store and pay the invoice to finish the order.

If payment method is "credit", but not enough credit is available to your account:

```
<LiteSpeed_eService>
  <action>Order</action>
  <license_id>6066</license_id>
  <license_type>WS_L_V</license_type>
  <modules>cache</modules> (Optional field. Will show if module is ordered.)
  <invoice_id>12466</invoice_id>
  <result>incomplete</result>
  <message>Invoice 12466 not paid.</message>
</LiteSpeed_eService>
```

If payment method is "creditcard", but the attempted credit card payment failed:

```
<LiteSpeed_eService>
  <action>Order</action>
  <license_id>9329</license_id>
  <license_type>WS_L_V</license_type>
  <modules>cache</modules> (Optional field. Will show if module is ordered.)
  <invoice_id>20568</invoice_id>
  <result>incomplete</result>
  <message>Invoice 20568 not paid (credit card capture failed).</message>
</LiteSpeed_eService>
```

3. If the transaction is successful, which should happen for the majority of cases, you will get a serial number back. You can parse the message to get the serial number and create your own script for installation. You will still receive the same confirmation email and serial number emails as if you ordered online. There will be no <invoice_id> if the charge was paid with credit.

```
<LiteSpeed_eService>

  <action>Order</action>

  < license_id>6067</ license_id>

  <license_type>WS_L_V</license_type>

  <modules>cache</modules> (Optional field. Will show if module is ordered.)

  <invoice_id>12466</invoice_id>

  < serial>gv06-kXsU-SHBr-pL4N</serial>

  <result>success</result>

  <message>new order accepted</message>

</LiteSpeed_eService>
```

3. Cancel Leased License

Just as if you logged into the LiteSpeed store and requested cancellation, you can choose to cancel immediately or at the end of the billing cycle, and you will get the same email for cancellation confirmation.

Input fields (GET or POST):

litespeed_store_login – LitesSeed online store account login email.

litespeed_store_pass – Account login password.

eService_action – **Cancel**

license_serial – License serial number. (Optional if server_ip is provided.)

server_ip – Server IP. (Optional if license_serial is provided.)

Input must contain <license_serial> or <server_ip>. If both fields are supplied, there will be extra validation to ensure they match each other. <license_serial> is preferred, as <server_ip> may map to multiple licenses. <server_ip> was added for convenience for

IPs that uniquely identify one active license. You can verify that there is only one license attached to a particular IP by using the "LicenseDetail_IP" function in a "query" action. If there are multiple licenses associated with one IP, a "cancel" action using only <server_ip> will be rejected.

cancel_now – When to cancel. The following values are available:

"Y": Immediately.

"N": End of Billing Cycle.

A nightly batch job will process this request, so even "immediate" cancellations will not be processed until the end of the day. If you made a mistake in your request, you will have time to send an email notifying us.

cancel_reason – **(optional)** We appreciate any feedback you can get from your users as to the reasons for cancellation.

Cancel Output:

1. If any errors occurred during the process, <result> will contain "error", and <message> will contain a detailed description:

```
<LiteSpeed_eService>
  <action>Cancel</action>
  <result>error</result>
  <message>procCancel.il.5: Cannot find this serial number</message>
</LiteSpeed_eService>
```

2. If the transaction completes successfully, in addition to the following output, you will receive the same confirmation email as you would have had you canceled manually:

```
<LiteSpeed_eService>
  <action>Cancel</action>
  <license_id>6067</license_id>
  <license_type>WS_L_V</license_type>
  <license_serial>gv06-kXsU-SHBr-pL4N</license_serial>
  <result>success</result>
```

```
<message>cancellation accepted – End of Billing Period</message>
</LiteSpeed_eService>
```

```
<LiteSpeed_eService>
  <action>Cancel</action>
  <license_id>9321</license_id>
  <license_type>WS_L_V</license_type>
  <license_serial>kiLX-KS/y-ZZ9U-183e</license_serial>
  <result>success</result>
  <message>cancellation accepted - Immediate</message>
</LiteSpeed_eService>
```

3. If the license has already been cancelled previously, you will receive a rejection message. Because all cancellations are processed via a nightly batch job, a license may still be listed as “Active” even after you send out a cancellation request:

```
<LiteSpeed_eService>
  <action>Cancel</action>
  <result>reject</result>
  <message>procCancel.2: This license has a cancellation request already -
2014-03-31 14:24:46 Immediate</message>
</LiteSpeed_eService>
```

4. Release Registered License

This is same as logging into the LiteSpeed store and clicking the “Release License” button on the product details page. It will free the serial number and allow it to be registered again.

Be careful: This will immediately remove access for the previously registered server. If the server is still running on LSWS, it will shut down. (For regular server migration, please follow the migration steps outlined in the [server documentation](#). Migration allows you to run the license on two servers in parallel for 3 days.)

After the serial number has been released, you can install the serial number on a new server and use the command `/usr/local/lsws/bin/lshhttpd -r` to register the server. This will generate a new license.key for this server with this serial number.

Input fields (GET or POST):

litespeed_store_login – LiteSpeed online store account login email.

litespeed_store_pass – Account login password.

eService_action – [ReleaseLicense](#)

license_serial – License serial number.

server_ip – The server IP that you want to be released from.

(The serial number has to be currently registered with this `<server_ip>` in order to pass validation and be released.)

License Release Output:

1. If there is any error during the process, `<result>` will contain “error” and `<message>` will contain a detailed description:

```
<LiteSpeed_eService>

  <action>ReleaseLicense</action>

  <result>error</result>

  <message>Cannot find this serial number</message>

</LiteSpeed_eService>
```

```
<LiteSpeed_eService>

  <action> ReleaseLicense </action>

  <result>error</result>
```

```
<message>procRelease.il.12: The IP you submitted xxx.xxx.xxx.xxx does not  
match the license record</message>
```

```
</LiteSpeed_eService>
```

2. If the transaction is successful, in addition to the following output, you will receive the same release confirmation email as if you had released the license manually through the store:

```
<LiteSpeed_eService>  
  
  <action>ReleaseLicense</action>  
  
  <license_id>6067</license_id>  
  
  <license_type>WS_L_V</license_type>  
  
  < license_serial>gv06-kXsU-SHBr-pL4N</license_serial>  
  
  <result>success</result>  
  
  <message>released from server xxx.xxx.xxx.xxx</message>  
  
</LiteSpeed_eService>
```

3. If the license has already been released, you will receive an rejection message. No action is required in this case.

```
<LiteSpeed_eService>  
  
  <action>ReleaseLicense</action>  
  
  <result>reject</result>  
  
  <message>procRelease.4: This license has already been  
released</message>  
  
</LiteSpeed_eService>
```

5. Suspend/Unsuspend Leased License

This is a tool to temporarily suspend a particular user's license in special cases, like nonpayment or policy violation. The web server checks in with the license server at least once every 24 hours. It will shut down when it sees the license has been suspended. As a consequence, your client's website(s) will go down. Please note, though, that this license will continue to appear on your invoices. Once the issue is resolved, you can use an "unsuspend" action to reactivate the

license, or you can request cancellation to permanently cancel it. Only requesting cancellation will take the license off your future invoices.

Input fields (GET or POST):

litespeed_store_login – LiteSpeed online store account login email.

litespeed_store_pass – Account login password.

eService_action – **Suspend** or **Unsuspend**

license_serial – License serial number. (Optional if server_ip is provided.)

server_ip – Server IP. (Optional if license_serial is provided.)

Input must contain <license_serial> or <server_ip>. If both fields are supplied, there will be extra validation to ensure they match each other. <license_serial> is preferred, as <server_ip> may map to multiple licenses. <server_ip> was added for convenience for IPs that uniquely identify one active license. You can verify this by using the “LicenseDetail_IP” function in a query action. If there are multiple licenses associated with one IP, a suspend/unsuspend action with only <server_ip> will be rejected.

reason – (optional) You can put an explanation here if you like.

Suspend/Unsuspend Output:

1. If any errors occurred during the process, <result> will contain “error” or “reject”, and <message> will contain a detailed description:

```
<LiteSpeed_eService>

  <action>Unsuspend</action>

  <result>error</result>

  <message>procUnsuspend.il.7: Cannot find this serial under your
  account</message>

</LiteSpeed_eService>
```

```
<LiteSpeed_eService>

  <action>Suspend</action>

  <result>reject</result>

  <message>procUnsuspend.il.8: This serial has been cancelled</message>
```

```
</LiteSpeed_eService>
```

2. If the transaction is successful, in addition to the following output, you will receive the same confirmation email you would have had you done it manually:

```
<LiteSpeed_eService>

  <action>Suspend</action>

  <license_id>6067</license_id>

  <license_type>WS_L_V</license_type>

  < license_serial>gv06-kXsU-SHBr-pL4N</ license_serial>

  <result>success</result>

  <message>License is suspended successfully</message>

</LiteSpeed_eService>
```

```
<LiteSpeed_eService>

  <action>Unsuspend</action>

  <license_id>6067</license_id>

  <license_type>WS_L_V</license_type>

  <license_serial>gv06-kXsU-SHBr-pL4N</license_serial>

  <result>success</result>

  <message>License is unsuspended successfully</message>

</LiteSpeed_eService>
```

3. If you try to suspend a license that is already suspended or try to unsuspend a license that is not suspended, you will receive a rejection message:

```
<LiteSpeed_eService>

  <action>Suspend</action>

  <result>reject</result>

  <message>procSuspend.1: This license has been suspended
already</message>

</LiteSpeed_eService>
```

```
<LiteSpeed_eService>

  <action>Unsuspend</action>

  <result>reject</result>

  <message>procUnsuspend.1: This license is not in suspension</message>

</LiteSpeed_eService>
```

6. Upgrade or Downgrade a License

This is same as if you logged into the LiteSpeed store to upgrade or downgrade your license. The license change will maintain the same leasing period (either monthly or yearly). For example, if you have a monthly leased 1-CPU LSWS license, you can change to a monthly leased 2-CPU license or a monthly leased VPS license. Owned licenses cannot be upgraded or downgrade through the eService API, nor can a lease period be changed. These changes must be made manually.

Upgrading a license will generate an invoice for the prorated amount due for the new license. Downgrading will generate a prorated amount of credit for the difference in license prices. You will receive email notifications for invoice payment and upgrade/downgrade confirmation.

Input fields (GET or POST):

litespeed_store_login – LiteSpeed online store account login email.

litespeed_store_pass – Account login password.

eService_version – 1.1

eService_action – **Upgrade**

license_serial – License serial number. (Optional if server_ip is provided.)

server_ip – Server IP. (Optional if license_serial is provided.)

Input must contain <license_serial> or <server_ip>. If both fields are supplied, there will be extra validation to ensure they match each other. <license_serial> is preferred, as <server_ip> may map to multiple licenses. <server_ip> was added for convenience for

IPs that uniquely identify one active license. You can verify this by using the "LicenseDetail_IP" function in a query action. If there are multiple licenses associated with one IP, an upgrade/downgrade action with only <server_ip> will be rejected.

upgrade_cpu – (Optional) The kind of license you want to upgrade/downgrade to. Available values:

"1": 1-CPU license

"2": 2-CPU license

"4": 4-CPU license

"8": 8-CPU license

"V": VPS license

"U": Ultra-VPS license *(Available for LSWS 4.2.2 and up.)*

add_modules – (Optional) *(Available for LSWS 4.2.8 and up.)* Comma-delimited list of modules to be added for this license. Available values:

"cache" *(Available for LSWS 4.2.8 and up. Applicable to VPS, Ultra-VPS and 1-CPU licenses.)*

remove_modules – (Optional) *(Available for LSWS 4.2.8 and up.)* Comma-delimited list of modules to be removed from this license. Available values:

"cache" *(Available for LSWS 4.2.8 and up. Applicable to VPS, Ultra-VPS and 1-CPU licenses.)*

At least one of the <upgrade_cpu>, <add_modules>, or <remove_modules> values must be included and be different from the current values, otherwise the upgrade request will be rejected. Multiple values can be included.

order_payment – Payment method. (Not necessary for downgrade requests. For downgrade requests, prorated credit will be added to the account.) Available values:

"credit": Use account credit. User can utilize "Add funds" function to pre-deposit money, which will show up as account credit.

"creditcard": Use credit card to pay. The credit card is pre-defined in the account. If there is available credit in the account, credit will be applied first even when the payment method is set to "creditcard".

order_cvv – (optional) Credit card security code. Try not to set this field. Only if your bank requires this (meaning that the transaction will fail without it) should you then supply this field. CVV code is not stored in the system, so if you need to set it, you have

to set this field every time. Other information from your credit card will be taken from your user account.

Upgrade/Downgrade Output:

1. If any errors occur during this process, <result> will contain "error" and <message> will contain a detailed description:

```
<LiteSpeed_eService>

  <action>Upgrade</action>

  <result>error</result>

  <message>Invalid cpu!</message>

</LiteSpeed_eService>
```

2. If the transaction cannot be completed, <result> will be "incomplete". For example, if payment method is "credit", but there is not enough credit in your account, or if payment method is "creditcard", but the charge cannot go through, then the transaction will not be completed and <result> will display "incomplete". <license_id> and <invoice_id> will be provided. Your package will be upgraded even if the invoice is not paid immediately. You will need to login to LiteSpeed's online store to pay the invoice manually. <old_modules>, <modules>, <module_invoice_id>, and <module_invoice_amt> will only appear if new modules have been added or removed.
If payment method is "credit", but not enough credit is available in your account:

```
<LiteSpeed_eService>

  <action>Upgrade</action>

  < license_serial>gv06-kXsU-SHBr-pL4N</license_serial>

  <license_id>6066</license_id>

  <license_type>WS_L_1</license_type>

  <old_license_type>WS_L_V</old_license_type>

  <invoice_id>12466</invoice_id>

  <invoice_amt>10.99</invoice_amt>

  <modules>cache</modules> (Current modules.)

  <old_modules></old_modules> (Modules before upgrade.)
```

```
<module_invoice_id>12467</module_invoice_id>

<module_invoice_amt>7.54</module_invoice_amt>

<result>incomplete</result>

<message>Upgrade from WS_L_V to WS_L_1 accepted. Invoice 12466 not
paid. Module cache added. Module upgrade invoice 12467 not
paid.</message>

</LiteSpeed_eService>
```

If payment method is “creditcard”, but the attempted credit card payment failed:

```
<LiteSpeed_eService>

<action>Upgrade</action>

<license_serial>gv06-kXsU-SHBr-pL4N</license_serial>

<license_id>9329</license_id>

<license_type>WS_L_1</license_type>

<old_license_type>WS_L_V</old_license_type>

<invoice_id>20568</invoice_id>

<invoice_amt>10.99</invoice_amt>

<result>incomplete</result>

<message>Upgrade from WS_L_V to WS_L_1 accepted. Invoice 20568 not
paid(credit card capture failed).</message>

</LiteSpeed_eService>
```

4. If the transaction finishes successfully, <result> will be “success”. <old_modules>, <modules>, <module_invoice_id>, and <module_invoice_amt> will only appear if new modules have been added or removed.
Downgrade sample return:

```
<LiteSpeed_eService>

<action>Upgrade</action>

<license_serial>gv06-kXsU-SHBr-pL4N</license_serial>

<license_id>6067</license_id>
```

```
<license_type>WS_L_V</license_type> (New license type.)  
<old_license_type>WS_L_1</old_license_type>  
<credit_amt>10.99</credit_amt> (For downgrades, credit is generated and  
shown in this field.)  
<result>success</result>  
<message>Downgrade from WS_L_1 to WS_L_V accepted.</message>  
</LiteSpeed_eService>
```

Upgrade sample return:

```
<LiteSpeed_eService>  
  <action>Upgrade</action>  
  <license_serial>gv06-kXsU-SHBr-pL4N</license_serial>  
  <license_id>6067</license_id>  
  <license_type>WS_L_1</license_type> (New license type.)  
  <old_license_type>WS_L_V</old_license_type>  
  <invoice_id>12466</invoice_id> (Upgrades generate a new invoice for the  
prorated difference in license prices. A new <invoice_id> will be provided.)  
  <invoice_amt>10.99</invoice_amt> (Prorated invoice balance.)  
  <result>success</result>  
  <message>Upgrade from WS_L_V to WS_L_1 accepted.</message>  
</LiteSpeed_eService>
```

Upgrade module sample return:

```
<LiteSpeed_eService>  
  <action>Upgrade</action>  
  <license_serial>gv06-kXsU-SHBr-pL4N</license_serial>  
  <license_id>6067</license_id>  
  <license_type>WS_L_V</license_type>  
  <modules>cache</modules> (Current modules.)
```

```
<old_modules></old_modules> (Modules before upgrade.)  
  
<module_invoice_id>12467</module_invoice_id> (New modules generate a  
new invoice for the price of the module prorated to the rest of the billing  
period.)  
  
<module_invoice_amt>7.54</module_invoice_amt>  
  
<result>success</result>  
  
<message>Module cache added.</message>  
  
</LiteSpeed_eService>
```

7. Query: Check License Related Information

Input fields (GET or POST):

litespeed_store_login – LiteSpeed online store account login email.

litespeed_store_pass – Account login password.

eService_version – 1.1

eService_action – **Query**

query_field – Currently supported values:

“**AllActiveLicenses**”

“**AllSuspendedLicenses**”

“**LicenseDetail_IP:IP Address**”

LicenseDetail_IP:xx.xxx.xxx.xxx (Please replace “IP Address” above with the IP address you would like to look up. No space between tag, colon and IP address.)

If there is more than one active license associated with the query IP, “error” will be returned.

“**LicenseDetail_Serial:Serial Number**” (Since Dec 16, 2011)

LicenseDetail_Serial:cccccccccccc (Please replace “Serial Number” above with the serial number you would like to look up. No space between tag, colon and serial number.)

If there is no active license with the query serial number (including licenses that have been canceled or terminated), "error" will be returned.

(If you have a specific function you'd like to see us implement, let us know. We'll do our best to make the system more useful.)

Query Output:

1. If any errors occur during this process, <result> will contain "error" or "reject" and <message> will contain a detailed description.

```
<LiteSpeed_eService>

  <action>Query</action>

  <result>error</result>

  <message>Invalid login</message>

</LiteSpeed_eService>
```

```
<LiteSpeed_eService>

  <action>Query</action>

  <result>reject</result>

  <message>LicenseDetail_Serial.il.8: This serial has already been
cancelled</message>

</LiteSpeed_eService>
```

2. If <query_field> is "AllActiveLicenses" or "AllSuspendedLicenses", <message> will contain a comma-delimited list of license types, ID, IP, and serial. License type lists product type, whether it is leased or owned, and VPS or number of CPUs. For example, WS_L_4 means leased 4-CPU LSWS license. WS_O_V means owned VPS LSWS license. WS_L_U means leased Ultra VPS LSWS license. <result> will be the total count of active licenses. If there is an immediate cancellation request against a license, it will not show up in the active list. Suspended licenses also do not show up in the active list.

```
<LiteSpeed_eService>

  <action>Query</action>

  <query_field>AllActiveLicenses</query_field>
```

```
<result>4</result>

<message>WS_L_1: id (xxx) ip (xxx.xxx.xxx.xxx) serial (aaaaaaaaaaaaa),
WS_L_2: id (xxx) ip (xxx.xxx.xxx.xxx) serial (bbbbbbbbbbbb),
WS_O_1: id (xxx) ip (xxx.xxx.xxx.xxx) serial (ccccccccccccc),
LB_L: id (xxx) ip (xxx.xxx.xxx.xxx) serial (ddddddddddd)

</message>

</LiteSpeed_eService>
```

3. If <query_field> is "LicenseDetail_IP:ip address" many details about the license associated with that IP will be returned:

Notes:

- Licenses are uniquely identified by serial numbers. More than one license may be used on the same IP. If an IP address is queried and has more than one license associated with it, the query output will be blank. For this reason, it may be better to query by serial number.
- The <license_expire_date> is generally seven days after the <next_due_date>. This is a seven day grace period. It gives users time to resolve any billing issues without facing downtime.
- <last_access_date> gives the last date that this license communicated with LiteSpeed's licensing server. This allows you to check whether the license is actually in use. This method is only meant to be part of verification. There are situations (like a misconfigured firewall) that could cause an in-use server to not contact LiteSpeed's licensing server. Just because <last_access_date> is old does not mean the license is definitely unused.
- <migrated_ip> will only contain a value if the license is in the process of being migrated, i.e. migration has started but the license has not been registered on a new server yet. In this case, <server_ip> will be blank and <migrated_ip> will show the IP of the server that the license is being migrated from.

```
<LiteSpeed_eService>

<action>Query</action>

<query_field>LicenseDetail_IP:xx.xxx.xxx.xxx</query_field>

<license_id>xxxx</license_id>

<license_type>LS_L_1</license_type>

<modules>cache</modules> (Only shows if modules exist for that license.)

<license_serial>ccccccccccccc</license_serial>
```

```
<license_expire_date>2014-02-08</license_expire_date>
<license_update_expire_date>2014-02-08</license_update_expire_date>
<server_ip>xx.xxx.xxx.xxx</server_ip>
<order_date>2011-12-01</order_date>
<billing_cycle>Monthly</billing_cycle>
<next_due_date>2014-02-01</next_due_date>
<status>Active</status>
<last_access_date>2014-01-07</last_access_date>
<migrated_ip></migrated_ip>
<result>success</result>
<message>LicenseDetail_IP</message>
</LiteSpeed_eService>
```

4. If <query_field> contains "LicenseDetail_Serial", output is same as query by IP, except <query_field> and <message> contain different values:

```
<LiteSpeed_eService>
  <action>Query</action>
  <query_field>LicenseDetail_Serial:cccccccccccc</query_field>
  <license_id>xxxx</license_id>
  <license_type>LS_L_1</license_type>
  <modules>cache</modules> (Only shows if modules exist for that license.)
  <license_serial>cccccccccccc</license_serial>
  <license_expire_date>2014-02-08</license_expire_date>
  <license_update_expire_date>2014-02-08</license_update_expire_date>
  <server_ip>xx.xxx.xxx.xxx</server_ip>
  <order_date>2011-12-01</order_date>
  <billing_cycle>Monthly</billing_cycle>
```

```
<next_due_date>2014-02-01</next_due_date>

<status>Active</status>

<last_access_date>2014-01-07</last_access_date>

<migrated_ip></migrated_ip>

<result>success</result>

<message>LicenseDetail_Serial</message>

</LiteSpeed_eService>
```

Appendix: Error Messages

Actions should not be taken based on the <message> field. This field is simply to provide some explanation and we cannot guarantee that these results will not change. The <result> field is a more stable variable that can be reliably used to determine actions.

Below are some <message> field values associated with different <result> values:

Rejection messages: <result>reject</result>

```
This license has been suspended already
This license has a cancellation request already
No registration entry found for serial xxx.xxx.xxx.xxx
This license has already been released
This license has been suspended already
This license is not suspended
Cannot find an active license for IP xxx.xxx.xxx.xxx
Cannot find a matching serial number for IP xxx.xxx.xxx.xxx
This serial has already been cancelled
This license is not active
The IP you submitted (xxx.xxx.xxx.xxx) does not match the license record
```

Error messages: <result>error</result>

```
eService Access denied!
eService IP Access denied!
Missing login email!
Missing login password!
```


Invalid login email!
Invalid password!
Invalid request!
Invalid action!
Missing eService_version
API version not supported – x.x
Missing required field “xxxxx”
Invalid field “xxxxx” - “xxxxx”
Missing field order_cpu
Missing field license_serial or server_ip
Missing field upgrade_cpu or add_modules or remove_modules
A module cannot be in both add_modules and remove_modules
Failed to create order
Failed to generate serial
Upgrade API only supports monthly or yearly leased licenses
Server IP does not match the registration records
Error in process
Invalid query field “xxxxx”
Cannot identify unique active license for IP xxx.xxx.xxx.xxx
Cannot find this serial number
Cannot find this serial under your account
Cannot find this license ID in DB
Internal DB error
Nothing found

Appendix: Update History

February 23, 2011: First release of API.

March 7, 2011: Added “releaselicense” action.

November 22, 2011: Added “suspend” and “unsuspend” actions.

December 16, 2011: Added look up by server IP option to “cancel”, “suspend”, and “unsuspend” actions. Added look up server IP and serial number functions to “query” action.

September 29, 2012: Added “query” action.

December 4, 2012: Added “update” action.

March 7, 2013: Added Ultra VPS license support.

January 15, 2014: Added “AllSuspendedLicenses” query.

February 3, 2014: Added <last_access_date> and <migrated_ip> to "LicenseDetail" queries. Added notes explaining "LicenseDetail" output.

March 20, 2014: Added support for modules and the cache module. For downgrades, credits will show up in "credit_amt" or "module_credit_amt" instead of as a negative value in the "invoice_amt" field. Added more details in the output "message" field.

We will continue to add more functions. Please send in your requests.