

registrar Module

Jan Janak
FhG FOKUS

Edited by
Jan Janak

registrar Module

Edited by and Jan Janakand Jan Janak

Copyright © 2003 FhG FOKUS

Revision History

Revision \$Revision: 1.1 \$ \$Date: 2003/07/24 14:10:55 \$

Table of Contents

1. User's Guide	1
1.1. Overview	1
1.2. Dependencies	1
1.2.1. SER Modules	1
1.2.2. External Libraries or Applications.....	1
1.3. Exported Parameters.....	1
1.3.1. default_expires (integer)	1
1.3.2. min_expires (integer).....	1
1.3.3. max_expires (integer).....	2
1.3.4. default_q (integer)	2
1.3.5. append_branches (integer)	2
1.3.6. use_domain (integer).....	3
1.3.7. case_sensitive (integer)	3
1.3.8. desc_time_order (integer).....	3
1.4. Exported Functions	4
1.4.1. save(domain)	4
1.4.2. save_noreply(domain)	4
1.4.3. lookup(domain)	4
1.4.4. registered(domain)	5
2. Developer's Guide	6
3. Frequently Asked Questions	7

List of Examples

1-1. Set default_expires parameter.....	1
1-2. Set min_expires parameter	2
1-3. Set max_expires parameter	2
1-4. Set default_q parameter.....	2
1-5. Set append_branches parameter.....	3
1-6. Set use_domain parameter	3
1-7. Set case_sensitive parameter.....	3
1-8. Set desc_time_order parameter.....	3
1-9. save usage	4
1-10. save_noreply usage	4
1-11. lookup usage	5
1-12. registered usage	5

Chapter 1. User's Guide

1.1. Overview

The module contains REGISTER processing logic.

1.2. Dependencies

1.2.1. SER Modules

The following modules must be loaded before this module:

- *usrloc* - *User Location Module*.
- *sl* - *Stateless Replies*.

1.2.2. External Libraries or Applications

The following libraries or applications must be installed before running SER with this module loaded:

- *None*.

1.3. Exported Parameters

1.3.1. `default_expires` (integer)

If the processed message contains neither Expires HFs nor expires contact parameters, this value will be used for newly created usrloc records. The parameter contains number of second to expire (for example use 3600 for one hour).

Default value is 3600.

Example 1-1. Set `default_expires` parameter

```
...  
modparam("registrar", "default_expires", 1800)  
...
```

1.3.2. min_expires (integer)

The minimum expires value of a Contact, values lower than this minimum will be automatically set to the minimum. Value 0 disables the checking.

Default value is 60.

Example 1-2. Set min_expires parameter

```
...
modparam("registrar", "min_expires", 60)
...
```

1.3.3. max_expires (integer)

The maximum expires value of a Contact, values higher than this maximum will be automatically set to the maximum. Value 0 disables the checking.

Default value is 0.

Example 1-3. Set max_expires parameter

```
...
modparam("registrar", "max_expires", 120)
...
```

1.3.4. default_q (integer)

The parameter represents default q value for new contacts. Because ser doesn't support float parameter types, the value in the parameter is divided by 100 and stored as float. For example, if you want default_q to be 0.38, use value 38 here.

Default value is 0.

Example 1-4. Set default_q parameter

```
...
modparam("registrar", "default_q", 100)
...
```

1.3.5. append_branches (integer)

The parameter controls how lookup function processes multiple contacts. If there are multiple contacts for the given username in usrloc and this parameter is set to 1, Request-URI will be overwritten with the highest-q rated

contact and the rest will be appended to sip_msg structure and can be later used by tm for forking. If the parameter is set to 0, only Request-URI will be overwritten with the highest-q rated contact and the rest will be left unprocessed.

Default value is 1.

Example 1-5. Set append_branches parameter

```
...
modparam("registrar", "append_branches", 0)
...
```

1.3.6. use_domain (integer)

If set to 1 then the registrar will use username@domain as address of record. If the variable is set to 0 then only username will be used as the address of record.

Default value is 0.

Example 1-6. Set use_domain parameter

```
...
modparam("registrar", "use_domain", 1)
...
```

1.3.7. case_sensitive (integer)

If set to 1 then AOR comparison will be case sensitive, if set to 0 then AOR comparison will be case insensitive--This is recommended.

Default value is 0.

Example 1-7. Set case_sensitive parameter

```
...
modparam("registrar", "case_sensitive", 1)
...
```

1.3.8. desc_time_order (integer)

If set to 1 then all contacts will be ordered in descending modification time order. In this case the most recently updated/created contact will be used.

Default value is 0.

Example 1-8. Set desc_time_order parameter

```
...
modparam("registrar", "desc_time_order", 1)
...
```

1.4. Exported Functions

1.4.1. save(domain)

The function processes a REGISTER message. It can add, remove or modify usrloc records depending on Contact and Expires HFs in the REGISTER message. On success, 200 OK will be returned listing all contacts that are currently in usrloc. On an error, error message will be send with a short description in reason phrase.

Meaning of the parameters is as follows:

- *domain* - Logical domain within registrar. If dababase is used then this must be name of the table which stores the contacts.

Example 1-9. save usage

```
...
save("location");
...
```

1.4.2. save_noreply(domain)

Same as `save()` but it doesn't send a reply.

Meaning of the parameters is as follows:

- *domain* - Logical domain within registrar. If database is used then this must be name of the table which stores the contacts.

Example 1-10. save_noreply usage

```
...
save_noreply("location");
...
```


1.4.3. lookup(domain)

The function extracts username from Request-URI and tries to find all contacts for the username in `usrloc`. If there are no such contacts, -1 will be returned. If there are such contacts, Request-URI will be overwritten with the contact that has the highest q value and optionally the rest will be appended to the message (depending on `append_branches` parameter value).

Meaning of the parameters is as follows:

- *domain* - Name of table that should be used for the lookup.

Example 1-11. lookup usage

```
...
lookup("location");
...
```

1.4.4. registered(domain)

The function returns true if the AOR in the Request-URI is registered, false otherwise. The function does not modify the message being processed, it neither rewrites the Request-URI if a contact is found nor appends branches.

Meaning of the parameters is as follows:

- *domain* - Name of table that should be used for the lookup.

Example 1-12. registered usage

```
...
if (registered("location")) {
    sl_send_reply("100", "Trying");
    ...
};
...
```

Chapter 2. Developer's Guide

The module does not provide any sort of API to use in other SER modules.

Chapter 3. Frequently Asked Questions

1. Where can I find more about SER?

Take a look at <http://iptel.org/ser>.

2. Where can I post a question about this module?

First at all check if your question was already answered on one of our mailing lists:

- <http://mail.iptel.org/mailman/listinfo/serusers>
- <http://mail.iptel.org/mailman/listinfo/serdev>

E-mails regarding any stable version should be sent to [<serusers@iptel.org>](mailto:serusers@iptel.org) and e-mail regarding development versions or CVS snapshots should be send to [<serdev@iptel.org>](mailto:serdev@iptel.org).

If you want to keep the mail private, send it to [<serhelp@iptel.org>](mailto:serhelp@iptel.org).

3. How can I report a bug?

Please follow the guidelines provided at: <http://iptel.org/ser/bugs>