



ENTERPRISE FEEDBACK SUITE

The Service Layer

Contents

1	The Service Layer	3
2	Configuring the Services	4
3	Accessing the Service Layer.....	5
3.1	Access Modes.....	5
3.2	Format Handler	5
3.3	Authentication	6
3.4	Calling the Service Description	6
3.5	Important Parameters	7
3.6	Examples	7

1 The Service Layer

Since EFS 8.1, the new Service Layer allows to access EFS key functionalities from outside EFS. Thus, applications relying on EFS data and functionalities can be created outside EFS with quite different technologies.

QuestBack customers profit from the new infrastructure in many respects:

- The spectrum of Customizing Solutions offered by QuestBack is extended, regarding both the available functionalities and the technologies that can be used.
- The Service Layer opens up new possibilities for customers who wish to connect to EFS with their own applications or who wish to integrate EFS into their own applications.

The scope of functionality in brief:

- From EFS 9.1, the Service Layer covers the scope of functionality of the old SOAP services. More key functionalities of EFS will be made available step by step in future releases. Future EFS functions which are of interest to customers will be made available as well.
- The services can be called from outside via HTTP or HTTPS.
- Various data exchange formats can be used (e.g. JSON).

2 Configuring the Services

A dedicated configuration menu allows to view and configure the services which are available on your EFS installation.

- The menu is in **Options-> Services->Service configuration**.
- The menu has the same design as the menu for the familiar SOAP services (**Options-> Services-> Web services**).
- The **Service configuration** menu is only available if QuestBack Support has activated usage of services for your installation. To view and access the menu, you need either read rights for the ACL right “webservice_conf” or membership in the root team.

The following steps are necessary to use a specific service:

- QuestBack Support has to add the service.
- The service must be activated. If necessary, the button **Activate all services** allows to activate all services available on the installation en bloc.
- The staff account used to access the service is allocated to a specific staff team. This staff team must have the necessary access rights to the service. Access rights to individual services can be assigned on the tab **Access groups**.
- Furthermore, many services include a check for object rights. To use survey.questionnaire.createPage or survey.questionnaire.deletePage, for example, the staff team needs write rights for the target project.
- All calls are logged on the **Access log** tab. The entries can be searched by IP address, name of the admin account used, service name and date.

3 Accessing the Service Layer

3.1 Access Modes

The service layer offers two access modes:

- **Description:** In this modus, the service layer will describe itself. In SOAP format, for example, a WSDL will be generated and delivered. This modus allows clients to find out which services are provided and which parameters they have
- **Transaction:** In this modus, a service method is called and executed.

How you address these modes depends on the format handler used.

3.2 Format Handler

Currently the service layer supports HTTP/HTTPS and three internal formats:

- **PHP-serialized:** Here, the input and output parameters are transferred as serialized PHP arrays. This is the recommended approach for PHP clients.
 - To activate the handler, use the URL parameter "handler" with the value "php".
 - The name of the invoked method is handed over in the URL parameter "method". The method name is structured as follows: MODULENAME.ACTORNAME.METHODNAME (separator: periods).
 - If the request is an HTTP GET request, the description mode will be triggered. Otherwise, the transaction mode will be used.
- **JSON:** Data are transferred in JSON encoding. See the example below.
 - To activate the handler, use the URL parameter "handler" with the value "json".
 - The name of the invoked method is handed over in the URL parameter "method". The method name is structured as follows: MODULENAME.ACTORNAME.METHODNAME (separator: periods).
 - If the request is an HTTP GET request, the description mode will be triggered. Otherwise, the transaction mode will be used.
- **SOAP:** SOAP is used to transfer data. See the example below.
 - To activate the handler, use the URL parameter "handler" with the value "soap".

- The name of the invoked method is handed over in the URL parameter "method". The method name is structured as follows: MODULENAME_ACTORNAME_METHODNAME (separator: underlines _).
- If the URL parameter "wsdl" is set in the request, the description mode will be triggered and a WSDL will be generated. Otherwise, the transaction mode will be used.

3.3 Authentication

Two authentication methods can be used:

- **Tokens:** QuestBack Support can provide tokens for you and your staff members. These tokens can be used for authentication when invoking a service (parameter name: "token").
- **Account name and password for the EFS admin area:** The easiest authentication method, as any staff member with admin area access already owns an account. But please mind: When calling the services via http, account name and password will be transferred unencrypted, i.e. it is possible to log them e.g. on proxies and use them to access the admin area. Therefore, QuestBack recommends to use tokens or to call the services via SSL.

3.4 Calling the Service Description

Self-description of the PHP handler:

```
http://your-domain.com/service/?handler=php
```

Calling the WSDL:

```
http://your-domain.com/service/?handler=soap&wsdl=1
```

Please note: The new service layer allows to configure access rights individually for each service and each staff team. Therefore, when accessing the service layer, you will now find a dynamic list of services. There is no central WSDL file anymore, instead the WSDL file will be generated dynamically each time someone invokes the services according to that person's access rights. That way, the invoking users will see a WSDL for the very services they are allowed to use.

3.5 Important Parameters

These are the most important parameters:

- handler: name of the format handler (php, json, soap).
- method: name of the called method for PHP and JSON handler.
- version: optionally-used version number of the Service API.
- token: contains the token for authentication.
- wsdl: generates the WSDL if the SOAP handler is used.

3.6 Examples

Example: JSON call with http authentication

```
/**
 * Please adapt
 **/

$baseUrl="http://your.server.name/";
$username="user1";
$password="topsecret";

/* ***** */

/**
 * if necessary please adapt
 */

set_include_path(get_include_path().':'.dirname(__FILE__).'/../wcp/3rd/');

/* ***** */

require 'Zend/Http/Client.php';

$call = array('method'=>'efs.system.getLoad','params'=>array());

$url=$baseUrl."service/index.php?handler=json";
```

```
$client=new Zend_Http_Client();
$client->setUri($url);
$client->setAuth($userName,$passwd);
$client->setMethod(Zend_Http_Client::POST);
$client->
setRawData(json_encode(array('method'=>$call['method'],'jsonrpc'=>'2.0','id'=>1,'params'=>$call['params'])));
$response=$client->request();

/**
 * Get JSON result
 */
$return=$response->getBody();
```

Example: SOAP call with http authentication

```
/**
 * Please adapt
 */
$baseUrl="http://your.server.name/";
$username="user1";
$password="topsecret";
/* ***** */

ini_set("soap.wsdl_cache_enabled",0);
ini_set("default_socket_timeout",10);

/**
 * if necessary please adapt
 */
```



```
set_include_path(get_include_path().'.':dirname(__FILE__).'/../wcp/3rd/');
```

```
/* ***** */
```

```
$call = array('method'=>'efs.system.getLoad','params'=>array());
```

```
$client = new SoapClient($baseUrl."service/index.php?handler=soap&wsdl=1",array(
    "style"=>SOAP_RPC,
    "use"=>SOAP_ENCODED,
    "encoding"=>"UTF-8",
    "login"=>$userName,
    "password"=>$passwd,
    "trace"=>1,
));
```

```
$r=call_user_func_array(array($client,str_replace('.', '_', $call['method'])), $call['params']);
```

```
/**
```

```
 * Get result as XML String
```

```
 **/
```

```
$req = $s->__getLastRequest();
```

© 2013 – QuestBack GmbH

The information contained in this publication is the property of QuestBack GmbH. Passing on and reproduction of this publication or any parts of it is only permitted with the express written approval of QuestBack GmbH.

This publication is issued without any guarantee whatsoever. Information contained herein can be changed without prior notification. In addition QuestBack accepts no guarantee as to the exactness or completeness of the information, text, graphics, links and other items in this publication. This publication is a preliminary version, which is not subject to any valid license agreement or other agreements with QuestBack. The software products offered by QuestBack and/or its distributors may contain software components of other software manufacturers. All products may show country-specific differences.

QuestBack, Enterprise Feedback Suite, EFS, EFS Survey, EFS Panel, EFS Employee and EFS Leadership, other QuestBack products and services mentioned in the text and their corresponding logos are trademarks and can be registered trademarks of QuestBack GmbH in Germany and other countries worldwide. All other names of products and services are trademarks of their respective companies.

Germany

QuestBack GmbH

Gustav-Heinemann-Ufer 72a

50968 Cologne

Norway (Head office)

QuestBack AS

Løvenskiolds gate 26

0260 Oslo