

ID	TSS001388	Creation date	June 12, 2009
Platform	S60 3rd Edition FP1, FP2	Devices	Tested on Nokia N95, N78
Category	Symbian C++	Subcategory	Browsing

Keywords (APIs, classes, methods, functions): RHTTPSession, RHTTPFilterCollection

Description

The HTTP Filter Plugins sit between the HTTP Framework and the client and are loaded whenever there is a HTTP Session on the device. The filter receives all the transaction events that take place between the client and the HTTP framework and can be used to manipulate them. Thus it's also possible to even stop a transaction and send a dummy HTTP response to the client.

Solution

A HTTP Transaction encapsulates a HTTP request and a response. The code snippet given below shows how to cancel the HTTP request received from the client.

```
iTransaction = aTransaction;    //take a copy of the transaction
    aTransaction.Cancel(iFilterHandle); //cancel the transaction
```

The transaction's response object can then be set with our own headers and body and the function `RHTTPTransaction::SendEventL()` is used to send the response events. The code snippet given below illustrates this:

```
RHTTPResponse resp = iTransaction.Response();
//Sending response headers
RHTTPHeaders hdrs = resp.GetHeaderCollection();
RStringF valStr = iSession.StringPool().OpenFStringL( KContentType );
THTTPHdrVal val( valStr );
hdrs.SetFieldL( iSession.StringPool().StringF( HTTP::EContentType,
RHTTPSession::GetTable(), val );
RStringF okStr = iSession.StringPool().OpenFStringL(KOk);
iTransaction.Response().SetStatusCode(200);
iTransaction.Response().SetStatusText(okStr);
iTransaction.SendEventL(THTTPEvent::EGotResponseHeaders, THTTPEvent::EIncoming, iFilterHandle);
valStr.Close();
```

TSS001388_-_Simulation_of_HTTP_responses_using_HTTP_Filter_Plugin

```
okStr.Close();
```

Sending response body

```
resp.SetBody(*this);  
iTransaction.SendEventL(THTTPEvent::EGotResponseBodyData, THTTPEvent::EIncoming, iFilterHandle);
```

Sending response complete event

```
iTransaction.SendEventL(THTTPEvent::EResponseComplete, THTTPEvent::EIncoming, iFilterHandle);
```

Sending transaction succeed event

```
iTransaction.SendEventL(THTTPEvent::ESucceeded, THTTPEvent::EIncoming, iFilterHandle);
```

Please note that the events must be sent in the above sequence for the transaction to succeed.

NOTE: HTTP filters require an extensive set of capabilities, including `NetworkControl`. See the below example application for details.

Example Application

<http://wiki.forum.nokia.com/index.php/Image:Filter.zip>