Simulation of HTTP responses using HTTP Filter Plugin on Symbian

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();
 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.

(C) Copyright Nokia 2014. All rights reserved.

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

Example Application

File:Filter.zip

Version Hint

Windows Phone: [[Category:Windows Phone]]
[[Category:Windows Phone 7.5]]
[[Category:Windows Phone 8]]

Nokia Asha: [[Category:Nokia Asha]]
[[Category:Nokia Asha Platform 1.0]]

Series 40: [[Category:Series 40]]
[[Category:Series 40 1st Edition]] [[Category:Series 40 2nd Edition]]
[[Category:Series 40 3rd Edition (initial release)]] [[Category:Series 40 3rd Edition FP1]] [[Category:Series 40 3rd Edition FP2]]
[[Category:Series 40 5th Edition (initial release)]] [[Category:Series 40 5th Edition FP1]]
[[Category:Series 40 6th Edition (initial release)]] [[Category:Series 40 6th Edition FP1]] [[Category:Series 40 Developer Platform 1.0]] [[Category:Series 40 Developer Platform 1.1]] [[Category:Series 40 Developer Platform 2.0]]

Symbian: [[Category:Symbian]]
[[Category:S60 1st Edition]] [[Category:S60 2nd Edition (initial release)]] [[Category:S60 2nd Edition FP1]] [[Category:S60 2nd Edition FP2]] [[Category:S60 2nd Edition FP3]]
[[Category:S60 3rd Edition (initial release)]] [[Category:S60 3rd Edition FP1]] [[Category:S60 3rd Edition FP2]]
[[Category:S60 5th Edition]]
[[Category:Symbian^3]] [[Category:Symbian Anna]] [[Category:Nokia Belle]]

(C) Copyright Nokia 2014. All rights reserved.