Class java.net.URLConnection.Callback

```
java.lang.Object
  |
  +--java.net.URLConnection.Callback
```

public abstract class **URLConnection.Callback**
extends Object

A callback object for a URLConnection. A callback object will be queried at different stages in the connection process: before and after the connection, and when a connection exception occurs. A connection exception may be an IOException or an exceptional protocol condition (such as a HTTP redirect or authentication failure.) A connection exception may occur during the connection itself, or during any other stage of the protcol, and is broadly defined to include any relevant stages of the connection. In an HTTP connection, for example, it may include the getInputStream or getOutputStream as well as other methods.

For backward compatibility reasons, the callback object is not guaranteed to be called by all connection objects. Implementation of protocol handlers are required to support callback objects.

See Also:

URLConnection.setCallback(java.net.URLConnection.Callback), URLConnection.getCallback()

Method Summary

abstract void	**afterConnection**(URLConnection connection) This method should be invoked by a connection object after a connection has been established.
abstract boolean	**afterEvent**(URLConnection connection, IOException exception) This method should be invoked by a connection object after a connection has been attempted and an exceptional condition has been encountered.
abstract void	**beforeConnection**(URLConnection connection) This method should be invoked by a connection object before a connection is established.

Methods inherited from class java.lang.Object
clone , equals , finalize , getClass , hashCode , notify , notifyAll , toString , wait , wait , wait

Method Detail

beforeConnection

```
public abstract void beforeConnection(URLConnection connection)
                            throws IOException
```

This method should be invoked by a connection object before a connection is established. This allows callback object to set pre-connection state, such as user and password information, cookie information, etc. The default does nothing.

If the callback object wishes to abort the connection, it should throw an IOException, and the connection object should let the exception be thrown.

Parameters:

`connection` - the connection on which the callback is being performed.

---

afterConnection

```
public abstract void afterConnection(URLConnection connection)
                              throws IOException
```

This method should be invoked by a connection object after a connection has been established. This allows callback object to consult various post-connection headers, such as content length and content type.

If the callback object decides that the resulting connection is invalid, it should the connection by throwing an exception. Protocol handlers, in turn, should call into the callback object before they set the connected flag of the connection so that if the callback throws an exception, the connection is not left in a connected state.

Parameters:

`connection` - the connection on which the callback is being performed.

Throws:

IOException - if the callback object decides that the resulting connection is invalid.

---

afterEvent

```
public abstract boolean afterEvent(URLConnection connection,
                                   IOException exception)
                            throws IOException
```

This method should be invoked by a connection object after a connection has been attempted and an exceptional condition has been encountered. For example, an exceptional condition could be an IOException such as an UnknownHostException, or exceptional protocol condition such as an HTTP redirect or an HTTP authentication failure. This allows callback object to decide what to do based on this exceptional condition. The callback may want to modify headers (such user and password information) and retry or it may want to abort the connection.

If the callback wants to let the failure stand, and let the connection itself throw the exception, it should return false. If it wants to retry the connection, it should return true. The default implementation throws the exception argument if it is not null and returns false if the exception argument is null.

Parameters:

`connection` - the connection on which the callback is being performed.

`exception` - the exception raised, if any. The exception will be a valid IOException for I/O errors, such as UnknownHostException, and will be null in case of a protocol error, such as a HTTP document not found (404) error or an FTP authentication failure.

Returns:

whether or not the retry the connection.

Throws:

IOException - if the callback object decides that the resulting connection is invalid.

Submit a bug or feature
Submit comments/suggestions about new javadoc look

This documentation was generated with a post-Beta4 version of Javadoc.