

How to validate FIX messages within the user code in FIX Antenna HFT

- [Overview](#)
- [How to validate FIX messages within the user code in FIX Antenna HFT](#)
- [FIX Antenna HFT Parser exceptions](#)

Overview

This article describes how to validate FIX messages in FIX Antenna HFT in case of FIX protocol violation.

How to validate FIX messages within the user code in FIX Antenna HFT

When the incoming FIX message violates FIX protocol there is a need to validate such message. If the message contains validation errors, an exception should be generated and a Reject message should be sent to the counterparty.

The `Utils::Exception` is a base class so the more specific exception from a parser is needed here.

Instead of `Utils::Exception` the `ParserException` is needed to obtain information for the Reject message.

The header file `\include\fixantennaengine\src\parser\src\ParserException.h` for the `ParserException` class contains all the needed information, including the reject reason.

The following piece of code shows how to validate messages:

```
void validateLiteMsg( Engine::MsgPipeElem* poolElem )
{
    try
    {
        int size;
        const char* ptr = elem->msg.toRaw(&size);
        std::auto_ptr<Engine::FIXMessage> nonLiteMsg( Engine::FIXMsgProcessor::singleton()->parse
(ptr, size, NULL, false, false ) ); //parse() can throw
        // validate the message values
        nonLiteMsg->checkFields();
        //validate the message structure
        Parser::Options opts; // set the opts as required
        opts.verifyTagsValues = false; //already done
        nonLiteMsg->checkRequired(opts);
    }
    catch (Parser::ParserException const& e)
    {
        sendReject(elem->msg, e.refSeqNum(), e.refTagID(), e.refMsgType(), e.getRejReason(), e.what()
);
    }
    catch (std::exception const &e)
    {
        logError(e.what());
    }
}
```

FIX Antenna HFT Parser exceptions

Here is the list of Parser exceptions thrown by all public methods of Fix Antenna HFT:

Exception	Description
<code>fixantenna/engine/src/parser/src/ParserException.h: class ParserException :</code> <code>public FixProtocolException</code>	Appears when something is wrong with the message format, e.g. the message format does not correspond to the protocol defines.
<code>fixantenna/engine/src/parser/src/ParserException.h: class</code> <code>InvalidTagNumberException : public ParserException</code>	The tag number is invalid.

fixantenna/engine/src/parser/src/ParserException.h: class RequiredTagMissingException : public ParserException	The required tag is missing.
fixantenna/engine/src/parser/src/ParserException.h: class TagNotDefinedForThisMessageTypeException : public ParserException	The tag is not defined for this message type.
fixantenna/engine/src/parser/src/ParserException.h: class TagSpecifiedWithoutValueException : public ParserException	The tag is specified without a value.
fixantenna/engine/src/parser/src/ParserException.h: class InvalidMsgTypeException : public ParserException	The MsgType (35) is invalid.
fixantenna/engine/src/parser/src/ParserException.h: class TagAppearsMoreThanOnceException : public ParserException	The tag is duplicated.
fixantenna/engine/src/parser/src/ParserException.h: class RepeatingGroupFieldsOutOfOrderException : public ParserException	Repeating group fields are out of order.
fixantenna/engine/src/parser/src/ParserException.h: class IncorrectNumInGroupCountForRepeatingGroupException : public ParserException	The NumInGroup count for repeating group is incorrect.
fixantenna/engine/src/parser/src/ParserException.h: class IncorrectChecksumException : public ParserException	The CheckSum is incorrect.
fixantenna/engine/src/parser/src/ParserException.h: class IncorrectMessageLengthException : public ParserException	The message length is invalid.
fixantenna/engine/src/parser/src/ParserException.h: class FieldLengthExceededException : public ParserException	The field length exceeds the maximum value.
fixantenna/engine/src/parser/src/ParserException.h: class TagParsingException : public ParserException	The parsing of the field from the raw message is failed.
fixantenna/engine/src/parser/src/ParserException.h: class TagOutOfOrderException: public Parser::ParserException	The tag is specified out of the required order.
fixantenna/engine/src/parser/src/ParserException.h: class RepeatingGroupIndexOutOfRangeException : public ParserException	The repeating group index is out of the range.
fixantenna/engine/src/parser/src/ParserException.h: class RequiredBlockWithoutFields : public ParserException	The required block doesn't contain any fields.
fixantenna/engine/src/parser/src/ParserException.h: class SenderTargetValidationError : public ParserException	The Sender/Target field differs from the expected in the received message.