

Creating a new ProtocolHandler

The protocol handler receives data from a transport handler and prepares it for the data handler or receives data from the data handler and prepares it for the transport handler. A simple protocol is a comma separated list (CSV).

To create a new protocol handler AbstractProtocolHandler must be extended (or IProtocolHandler implemented). Two constructors are needed. A default constructor for OSGi and a constructor with the arguments in the codeblock below. The super constructor needs to be called with these arguments, they are needed for the connection of the transport-, protocol- and data-handler.

```
public YourProtocolHandler() {
    super();
}

public YourProtocolHandler(ITransportDirection direction, IAccessPattern access, OptionMap options,
    IStreamObjectDataHandler<T> dataHandler) {
    super(direction, access, dataHandler, options);
}

/**
 * Creates a new protocol handler
 * @param direction is this handler used in a source (IN) or in a sink (OUT)
 * @param access which kind of access pattern is supported ( PUSH, PULL, ROBUST_PUSH, ROBUST_PULL)
 * @param options set of options as key value pairs
 * @param dataHandler the data handler thats connected to the protocol handler
 * @return
 */
public IProtocolHandler<T> createInstance(ITransportDirection direction, IAccessPattern access,
    Map<String, String> options, IDataHandler<T> dataHandler) {
    return new YourProtocolHandler<>(direction, access, options, dataHandler);
}
```

The following method could be overwritten:

- open(): This method is called, when the query is started.
Important: When overwriting this method, getTransportHandler().open() must be called, too.
- close(): This method is called, when the query is stopped.
Important: When overwriting this method, getTransportHandler().close() must be called, too.

Pullbased

- boolean hasNext(): must be overwritten, to state if a new element is available for processing
- T getNext(): must be overwritten to deliver the next element
- boolean isDone(): can be overwritten to state if a source will not deliver anymore elements

Pushbased

For pushbased access the methods from the Interface ITransportHandlerListener need to be overwritten:

```

/**
 * Is called when a new connection with the transport handler is established
 * @param caller
 */
void onConnect(ITransportHandler caller);

/**
 * Is called when an existing connection to the transport handler is interrupted
 * @param caller
 */
void onDisconnect(ITransportHandler caller);

/**
 * Implement this method to process the message
 * @param message as ByteBuffer
 */
void process(ByteBuffer message);

/**
 * Implement this method to process the message
 * @param message as String Array
 */
void process(String[] message);

/**
 * Implement this method to process the message
 * @param message as T
 */
void process(T m);

```

Typically, these methods are called from the underlying transport handler