

Benutzerhandbuch



Neukirchen
August 2015

Kontakt:

Kai Hübl
Lambertsberg 17
D-34626 Neukirchen
kai.huebl@asneg.de

Contents

1	Einleitung.....	5
1.1	Inhalt	5
1.2	OpcUaWebServer	5
1.3	Web Panel	6
2	Installation	7
3	Konfiguration.....	7
3.1	OpcUaServer.xml	7
3.1.1	OpcUaWebServer	7
3.1.2	HttpServer	8
3.1.3	WebSocketServer.....	8
3.1.4	OpcUaClient	8
3.2	OpcUaClient.xml	9
3.2.1	OpcUaClient	9
3.2.2	Endpoint.....	9
3.2.3	NamespaceUri.....	9
3.2.4	NodeList.....	10
4	Websocket Kommunikation	10
4.1	Überblick	10
4.2	Notation	10
4.3	Message	11
4.4	Header	11
4.5	Management.....	12
4.5.1	ValueListRequest.....	12
4.5.2	ValueListResponse	12
4.5.3	ValueInfoRequest	12
4.5.4	ValueInfoResponse.....	13
4.6	Lesen	14
4.6.1	ReadRequest	14
4.6.2	ReadResponse	14
4.7	Lesen von historischen Daten	14
4.7.1	HistoricalReadRequest.....	14
4.7.2	HistoricalReadResponse	15
4.8	Schreiben	16
4.8.1	WriteRequest	16
4.8.2	WriteResponse	16
4.9	Monitoring	17
4.9.1	MonitorStartRequest.....	17
4.9.2	MonitorSartResponse.....	17

4.9.3	MonitorStopRequest	17
4.9.4	MonitorStopResponse.....	18
4.9.5	MonitorUpdateMessage.....	18

1 Einleitung

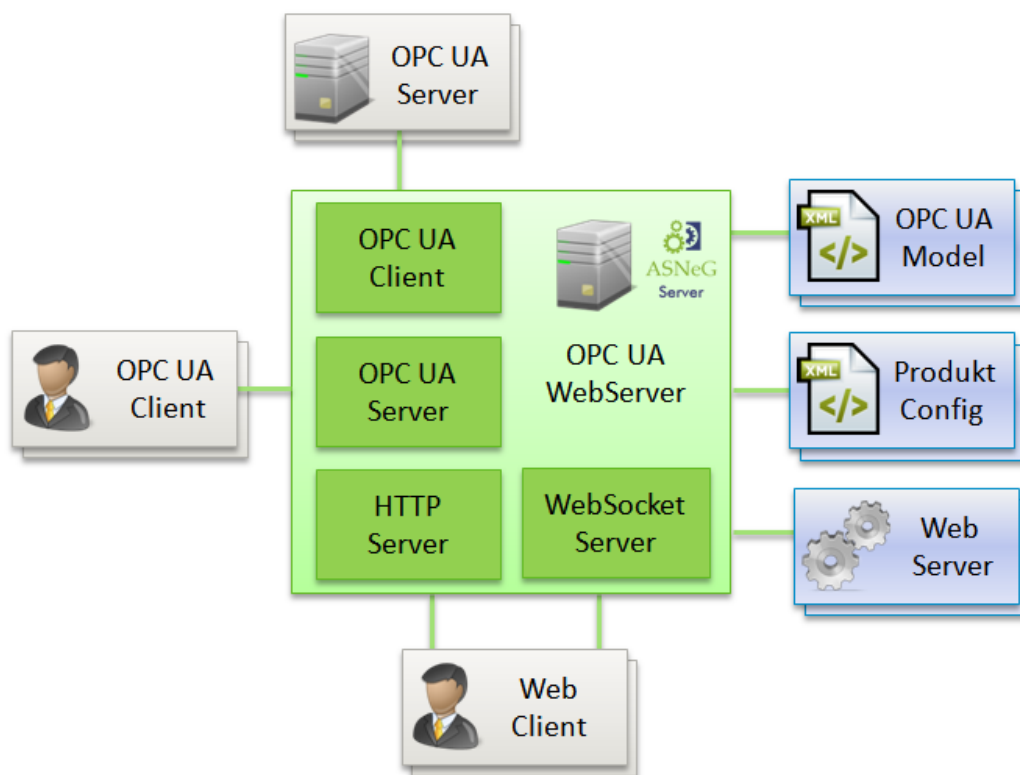
1.1 Inhalt

Das Installations-Paket OpcUaWebServer enthält mehrere Libraries und Anwendungen, die für verschiedene Zwecke eingesetzt werden können. Die folgende Liste gibt einen Überblick über den Inhalt des Installations-Paketes.

- OpcUaWebServer
- Web Panel

1.2 OpcUaWebServer

Der OPC UA WebServer verbindet eine Webanwendung mit einem oder mehreren OPC UA Servern und kann somit für beliebige Visualisierungslösungen eingesetzt werden.



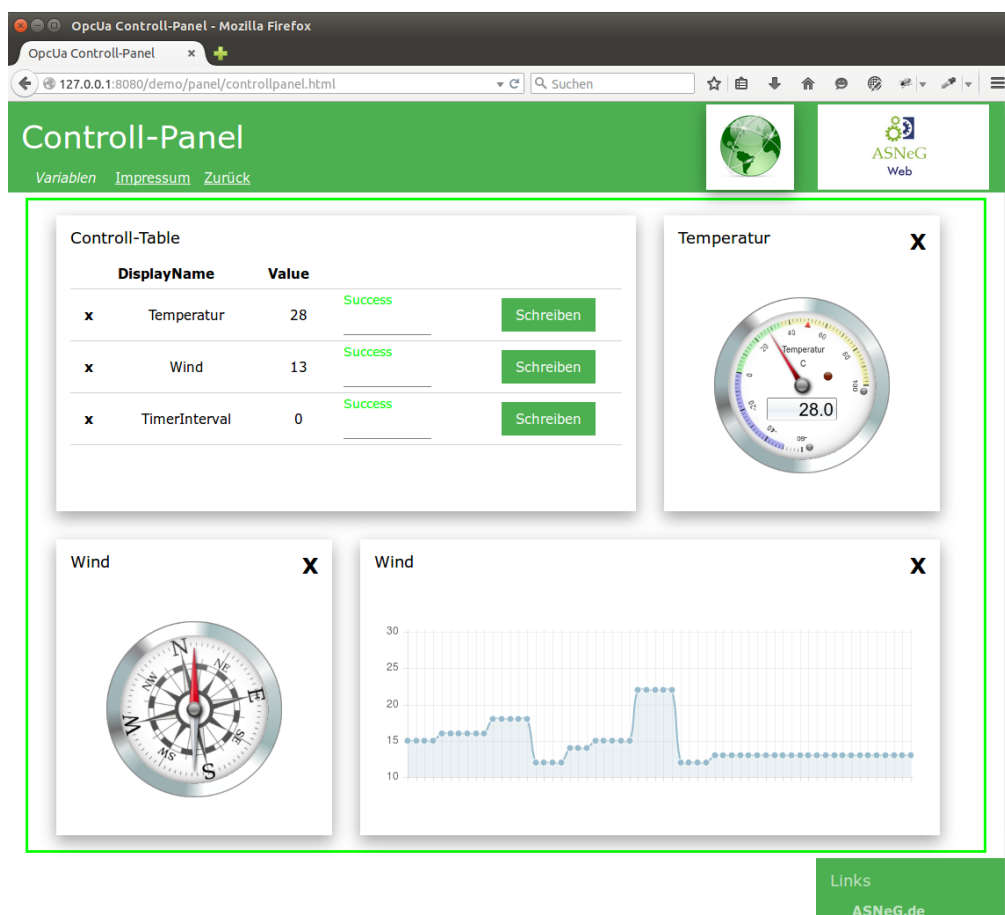
Für die Anbindung eines oder mehrerer OPC UA Server besitzt der OPC UA WebServer eine OPC UA Client Komponente. Die OPC UA Server Komponente wird für Management Aufgaben verwendet.

Die Anbindung an die Webanwendung erfolgt über zwei Schnittstellen. Die HTTP Schnittstelle (Port 8080) wird für den Zugriff auf die Webseite durch die Web Clients verwendet. Die WebSocket Schnittstelle (Port 8081) wird vom Web Client für den Abruf von Prozess Daten verwendet, welche über die OPC UA Client Komponente bereitgestellt werden. Die Kommunikation zwischen der Webanwendung und dem OpcUaServer über den WebSocket erfolgt bidirektional und ereignisgesteuert. Die Prozessdaten werden ereignisgesteuert vom OpcUaWebServer an die Webanwendung übertragen. Polling der Prozessdaten durch die Webanwendung wird somit nicht benötigt.

Für den Zugriff auf die Prozessdaten durch den Web Client steht eine JavaScript Bibliothek zur Verfügung.

1.3 Web Panel

Das Web Panel wird für die Visualisierung von Prozessdaten verwendet. Die Webseite des Web Panels wird vom OPC UA WebServer über eine HTTP Schnittstelle abgerufen. Der Abruf der Prozessdaten erfolgt ebenfalls über den OPC UA WebServer über eine WebSocket Schnittstelle.



2 Installation

Die Installation des Pakets ist identisch mit der Installation für den OpcUaServer. Bitte entnehmen Sie die Installationsanleitung aus diesem Handbuch.

2.1.1 Systemvoraussetzungen

Für die Installation des OpcUaWebServer müssen die folgenden Voraussetzungen vor der Installation erfüllt sein.

- Installation des OpcUaStack
(<http://asneg.de/download.html>)
- Installation der OpenSSL in der Version 1.0.2d (32Bit)
(<http://heise.de/download/win32-openssl.html>)

3 Konfiguration

3.1 OpcUaServer.xml

Der Inhalt der Konfigurationsdatei besteht aus einem serverspezifischen Teil für den OpcUaServer und einem anwendungsspezifischem Teil für den OpcUaWebServer. Der serverspezifische Teil für den OpcUaServer ist im Handbuch für den OpcUaServer beschrieben. Bitte entnehmen Sie die Anleitung aus diesem Handbuch. Die Beschreibung der Konfiguration für den OpcUaWebServer sehen Sie im folgenden Abschnitt.

3.1.1 OpcUaWebServer

OpcUaWebServer	Die gesamte Konfiguration für den OpcUaWebServer wird mit diesem Tag umschlossen. Der Tag befindet sich innerhalb des Tag OpcUaServer der OpcUaServer Konfigurationsdatei.
HttpServer	In diesem Bereich wird die Konfiguration für den HTTP Server abgelegt.
WebSocketServer	In diesem Bereich wird die Konfiguration für den WebSocket Server abgelegt.
OpcUaClient	In diesem Bereich wird die Konfiguration für den OPC UA Client abgelegt.

3.1.2 HttpServer

HttpServer	In diesem Bereich wird die Konfiguration für den HTTP Server abgelegt.
Address	IP Adresse des HTTP Servers.
Port	Port des HTTP Servers.
WebDirectory	Root Verzeichnis in dem sich die Webseiten des HTTP Servers befinden.
RequestTimeout	Sendet ein Webclient nach dem Öffnen der Verbindung zum HTTP Server innerhalb einer definierten Zeitspanne keine Anfrage, so wird die Verbindung zum Webclient vom HTTP Server geschlossen. Dieser Parameter definiert diese Zeitspanne in Millisekunden.
IPLogger	In diesem Bereich wird die Konfiguration für den IP logger abgelegt. Der IP Logger protokolliert alle IP Adressen der Clients die auf den HTTP Server zugreifen.
LogFileName	Verzeichnis und Name der Logdatei.
MaxLogFileNumber	Maximale Anzahl der Logdateien
MaxLogFileSize	Maximale Größe der Logdatei.
MaxIPAge	Greift ein Client mehrfach auf den HTTP Server zu, so wird die IP Adresse innerhalb einer definierten Zeitspanne nur einmal protokolliert. Dieser Parameter definiert diese Zeitspanne in Millisekunden.

3.1.3 WebSocketServer

OpcUaServer	In diesem Bereich wird die Konfiguration für den WebSocket Server abgelegt.
Address	IP Adresse des WebSocket Server
Port	Port des WebSocket Server
RequestTimeout	Sendet ein Webclient nach dem Öffnen der Verbindung zum WebSocket Server innerhalb einer definierten Zeitspanne keine Anfrage, so wird die Verbindung zum Webclient vom WebSocket Server geschlossen. Dieser Parameter definiert diese Zeitspanne in Millisekunden.
IdleTimeout	Erfolgt innerhalb einer definierten Zeitspanne keine Kommunikation zwischen dem Webclient und dem WebSocket Server, so wird die Verbindung zum Webclient durch den WebSocket Server geschlossen. Dieser Parameter definiert diese Zeitspanne in Millisekunden.

3.1.4 OpcUaClient

OpcUaClient	In diesem Bereich wird die Konfiguration für den OPC UA
--------------------	---

	Client abgelegt.
ClientConfigFile*	Verzeichnis und Name der Konfigurationsdatei für den OPC UA Client

3.2 OpcUaClient.xml

Diese Konfigurationsdatei wird für die Konfiguration der OPC UA Client Schnittstelle verwendet. In der Konfigurationsdatei OpcUaServer.xml (Tag ClientConfigFile) muss ein Verweis auf diese Konfigurationsdatei existieren. Sollen mehrere OPC UA Server vom OPC UA Client angesprochen werden so muss für jeden OPC UA Server eine eigene Client Konfigurationsdatei angelegt werden.

3.2.1 OpcUaClient

OpcUaClient	Die gesamte Konfiguration für den OPC UA Client wird mit diesem Tag umschlossen.
Endpoint	In diesem Bereich wird der Endpunkt des OPC UA Servers abgelegt.
NamespaceUri	In diesem Bereich wird die Liste der Namespace Uris abgelegt.
NodeList	In diesem Bereich wird die Liste der OPC UA Server Nodes abgelegt.

3.2.2 Endpoint

Endpoint	In diesem Bereich wird der Endpunkt des OPC UA Servers abgelegt.
ServerUri	Server Uri des OPC UA Servers mit dem sich der OPC UA Client verbinden soll.

3.2.3 NamespaceUri

NamespaceUri	In diesem Bereich wird die Liste der Namespace Uris abgelegt.
Uri+	Liste aller Namespace Uri für die in der Konfiguration verwendeten Nodes. Der ersten Uri in der Liste wird der Namespace Index 0, der zweiten Uri in der Liste der Namespace Index 1 usw. zugeordnet.

3.2.4 NodeList

NodeList	In diesem Bereich wird die Liste der OPC UA Server Nodes abgelegt, auf die der OPC UA Client Zugriff erhalten soll.
Node+	Enthält die Parameter der Node des Servers auf die der OPC UA Client Zugriff erhalten soll.
Attr.ValueName	Jeder OPC UA Server Node wird ein eindeutiger Name zugeordnet. Dieser Parameter definiert diesen Namen. Der ValueName wird in der Webanwendung für die eindeutige Identifizierung der Variable verwendet.
Attr.NodeId	Node Id der OPC UA Variable. Der OPC UA Client verwendet die Node Id für den Zugriff auf die Variable im OPC UA Server. Der in der Node Id verwendete Namespace Index muss im Bereich NamespaceUri definiert sein.
Attr.NodeType	Type der OPC UA Variable.
Attr.Array	Gibt an ob es sich bei der OPC UA Variable um ein Array handelt.
MetaData	Dieser Tag wird für die Definition beliebiger Metadaten Objekte verwendet. Metadaten Objekte können über den ValueName von der Webanwendung abgerufen und für beliebige Zwecke verwendet werden.

4 Websocket Kommunikation

4.1 Überblick

Die Kommunikation zwischen einer Webanwendung und dem OpcUaServer erfolgt über ein Websocket (Port 8081) unter Verwendung des Websocket-Protokoll (RFC 6455) auf Protokollebene. Websockets haben den Vorteil, dass Daten nicht über Polling, sondern ereignisbasiert, vom OpcUaWebServer an die Webanwendung übertragen werden können.

Zum Aufbau der Kommunikationsverbindung öffnet die Webanwendung in der Rolle des Clients die Verbindung zum Websocket des OpcUaWebServer.

Die Kommunikation auf Anwendungsebene zwischen der Webanwendung und dem OpcUaServer erfolgt über Nachrichten. Als Format für die Nachrichten wird das JSON Format verwendet.

4.2 Notation

In diesem Abschnitt wird die folgende Notation verwendet.

- [**<Tag>**] Der Tag ist in der Konfiguration keinmal oder genau einmal enthalten
- **<Tag>** Der Tag ist in der Konfiguration genau einmal enthalten
- **<Tag>*** Der Tag ist in der Konfiguration keinmal oder mehrfach enthalten
- **<Tag>+** Der Tag ist in der Konfiguration einmal oder mehrfach enthalten
- **@<Tag>** Der Tag enthält ein Array

4.3 Message

Jede JSON Nachricht hat den folgenden Aufbau.

JSON Nachricht		
	Header	In diesem Bereich wird der Header der Nachricht abgelegt.
	Body	In diesem Bereich wird der Body der Nachricht abgelegt.

4.4 Header

Jede JSON Nachricht besitzt einen Request Header mit dem folgenden Aufbau.

Header	In diesem Bereich werden die Header Informationen für einen Request bzw. für eine Response abgelegt.
ClientHandle	Dieser Parameter wird vom Client für die Zuordnung eines Request zur Response verwendet. Der ClientHandle wird vom Client in einem Request an den Server übergeben und vom Server in der Response an den Client zurückgegeben.
MessageType	Dieser Parameter definiert den Type der Nachricht. Folgende Typen werden unterstützt. READ_REQUEST READ_RESPONSE WRITE_REQUEST WRITE_RESPONSE VALUELIST_REQUEST VALUELIST_RESPONSE VALUEINFO_REQUEST VALUEINFO_RESPONSE MONITORSTART_REQUEST MONITORSTART_RESPONSE MONITORSTOP_REQUEST MONITORSTOP_RESPONSE

	MONITORUPDATE_MESSAGE HISTORICALREAD_REQUEST HISTORICALREAD_RESPONSE
[StatusCode]	Dieser Parameter wird für die Übergabe einer Fehlerbeschreibung vom Server zum Client verwendet, wenn auf der Kommunikationsebene ein Fehler aufgetreten ist. Der Parameter wird nur in einer Response Nachricht verwendet. Ist kein Fehler aufgetreten so ist der Parameter nicht vorhanden.

4.5 Management

4.5.1 ValueListRequest

In der Clientkonfiguration des OpcUaWebServers werden alle OPC UA Variablen übergeordneter Server, auf die zugegriffen werden soll, in einer Nodeliste definiert. In der Nodeliste wird für jede Node ein eindeutiger Name durch das Attribute ValueName festgelegt. Der Name einer Variablen wird in der Webanwendung für den Zugriff auf den Wert dieser Variablen verwendet. Die Webanwendung ruft mit dem ValueListRequest eine Liste der Variablen vom OpcUaWebServer ab. Der ValueListRequest wird vom OpcUaServer mit einer ValueListResponse beantwortet.

Header	In diesem Bereich werden die Header abgelegt.
---------------	---

4.5.2 ValueListResponse

Header	In diesem Bereich werden die Header abgelegt.
Body	In diesem Bereich wird die Nachricht abgelegt.
[StatusCode]	Dieser Parameter wird für die Übergabe einer Fehlerbeschreibung vom Server zum Client verwendet, wenn auf der Anwendungsebene ein Fehler aufgetreten ist. Ist kein Fehler aufgetreten so ist der Parameter nicht vorhanden.
@Variables	Dieser Parameter enthält ein Array von Variablenamen.

4.5.3 ValueInfoRequest

In der Clientkonfiguration des OpcUaWebServers werden alle OPC UA Variablen übergeordneter Server, auf die zugegriffen werden soll, in einer Nodeliste definiert. Die in

der Nodeliste definierten Variablen enthalten zusätzliche Informationen, wie zum Beispiel der Datentype der Variable. Zusätzlich können für jede Variable innerhalb eines Meta Tags beliebige Metainformationen für die Variable definiert werden. Die Webanwendung ruft mit dem ValueInfoRequest Metadaten vom OpcUaServer ab. Der ValueInfoRequest wird vom OpcUaWebServer mit einer ValueInfoResponse beantwortet.

Header	In diesem Bereich werden die Header Informationen abgelegt.
Body	In diesem Bereich wird die Nachricht abgelegt.
@Variables	Dieser Parameter enthält ein Array von Variablennamen für die Informationen vom OpcUaWebServer abgerufen werden sollen.

4.5.4 ValueInfoResponse

Header	In diesem Bereich werden die Header Informationen abgelegt.
Body	In diesem Bereich wird die Nachricht abgelegt.
[StatusCode]	Dieser Parameter wird für die Übergabe einer Fehlerbeschreibung vom Server zum Client verwendet, wenn auf der Anwendungsebene ein Fehler aufgetreten ist. Ist kein Fehler aufgetreten so ist der Parameter nicht vorhanden.
@Variables	Dieser Parameter enthält ein Array von Ergebnisobjekten.
[StatusCode]	Dieser Parameter wird für die Übergabe einer Fehlerbeschreibung vom Server zum Client verwendet, wenn die angeforderte Variable nicht ermittelt werden konnte. Ist kein Fehler aufgetreten so ist der Parameter nicht vorhanden.
Variable	Enthält den Namen der Variable.
Type	Dieser Parameter enthält den Type der Variable.
IsArray	Dieser Parameter gib an ob es sich bei der Variable um eine Array Variable handelt.
MetaData	Der Benutzer kann für eine Variable beliebige Metadaten definieren. Dieser Parameter enthält diese Metadaten.

4.6 Lesen

4.6.1 ReadRequest

Der Wert einer Variable wird von der Webanwendung mit dem ReadRequest vom OpcUaServer angefordert. Der ReadRequest wird vom OpcUaServer mit einer ReadResponse beantwortet.

Header	In diesem Bereich werden die Header Informationen abgelegt.
Body	In diesem Bereich wird die Nachricht abgelegt.
Variable	Dieser Parameter enthält den Namen der Variable für die der Wert abgefragt werden soll.

4.6.2 ReadResponse

Header	In diesem Bereich werden die Header abgelegt.
Body	In diesem Bereich wird die Nachricht abgelegt.
[StatusCode]	Dieser Parameter wird für die Übergabe einer Fehlerbeschreibung vom Server zum Client verwendet, wenn auf der Anwendungsebene ein Fehler aufgetreten ist. Ist kein Fehler aufgetreten so ist der Parameter nicht vorhanden.
Variable	Dieser Parameter enthält den Wert der Variable. Das Format ist Abhängig vom Datentype der gelieferten Variable.
SourceTimestamp	Dieser Parameter enthält den SourceTimestamp der Variable. Als Format für die Zeitangabe wird ISO 8601 verwendet. Beispiel: "2015-09-06T09:03:21.237123"
ServerTimestamp	Dieser Parameter enthält den ServerTimestamp der Variable. Als Format für die Zeitangabe wird ISO 8601 verwendet. Beispiel: "2015-09-06T09:03:21.237123"

4.7 Lesen von historischen Daten

4.7.1 HistoricalReadRequest

Historische Daten einer Variablen werden von der Webanwendung mit dem HistoricalReadRequest vom OpcUaServer angefordert. Der HistoricalReadRequest wird vom OpcUaServer mit einer HistoricalReadResponse beantwortet.

Header	In diesem Bereich werden die Header Informationen abgelegt.
Body	In diesem Bereich wird die Nachricht abgelegt.
Variable	Dieser Parameter enthält den Namen der Variable für die historische Daten abgefragt werden sollen.
StartTime	Dieser Parameter enthält den Startzeitpunkt der historischen Datenreihe. Als Format für die Zeitangabe wird ISO 8601 verwendet. Beispiel: "2015-09-06T09:03:21.237123"
EndTime	Dieser Parameter enthält den Endzeitpunkt der historischen Datenreihe. . Als Format für die Zeitangabe wird ISO 8601 verwendet. Beispiel: "2015-09-06T09:03:21.237123"

4.7.2 HistoricalReadResponse

Header	In diesem Bereich werden die Header abgelegt.
Body	In diesem Bereich wird die Nachricht abgelegt.
@DataValues	In diesem Bereich ist ein Array von Objekten enthalten. Jedes Objekt enthält einen Datenwert der Zeitreihe.
[StatusCode]	Dieser Parameter wird für die Übergabe einer Fehlerbeschreibung vom Server zum Client verwendet, wenn auf der Anwendungsebene ein Fehler aufgetreten ist. Ist kein Fehler aufgetreten so ist der Parameter nicht vorhanden.
Value	Dieser Parameter enthält den Wert der Variable. Das Format ist Abhängig vom Datentype der gelieferten Variable.
SourceTimestamp	Dieser Parameter enthält den SourceTimestamp der Variable. Als Format für die Zeitangabe wird ISO 8601 verwendet. Beispiel: "2015-09-06T09:03:21.237123"
ServerTimestamp	Dieser Parameter enthält den ServerTimestamp der Variable. Als Format für die Zeitangabe wird ISO 8601 verwendet. Beispiel: "2015-09-06T09:03:21.237123"

4.8 Schreiben

4.8.1 WriteRequest

Der Wert einer Variable wird von der Webanwendung mit dem WriteRequest vom OpcUaServer in eine OPC UA Variable geschrieben. Der WriteRequest wird vom OpcUaServer mit einer WriteResponse beantwortet.

Header	In diesem Bereich werden die Header abgelegt.
Body	In diesem Bereich wird die Nachricht abgelegt.
Variable	Dieser Parameter enthält den Namen der Variable, für die ein Wert in die OPC UA Variable geschrieben werden soll.
Value	Dieser Parameter enthält den Wert der Variable. Das Format ist Abhängig vom Datentype der gelieferten Variable.
SourceTimestamp	Dieser Parameter enthält den SourceTimestamp der Variable. Als Format für die Zeitangabe wird ISO 8601 verwendet. Beispiel: "2015-09-06T09:03:21.237123"
ServerTimestamp	Dieser Parameter enthält den ServerTimestamp der Variable. Als Format für die Zeitangabe wird ISO 8601 verwendet. Beispiel: "2015-09-06T09:03:21.237123"

4.8.2 WriteResponse

Header	In diesem Bereich werden die Header abgelegt.
Body	In diesem Bereich wird die Nachricht abgelegt.
[StatusCode]	Dieser Parameter wird für die Übergabe einer Fehlerbeschreibung vom Server zum Client verwendet, wenn auf der Anwendungsebene ein Fehler aufgetreten ist. Ist kein Fehler aufgetreten so ist der Parameter nicht vorhanden.

4.9 Monitoring

4.9.1 MonitorStartRequest

Der OpcUaWebServer überwacht OPC UA Variablen. Ändert sich der Wert oder der Status einer überwachten OPC UA Variable, so wird automatisch eine Update Nachricht vom OpcUaWebServer an die Webanwendung mit dem neuen Wert bzw. Status übertragen. Zum Start einer Überwachung sendet die Webanwendung ein MonitorStartRequest an den OpcUaWebServer. Der MonitorStartRequest wird vom OpcUaWebServer mit einer MonitorStartResponse beantwortet.

Header	In diesem Bereich werden die Header Informationen abgelegt.
Body	In diesem Bereich wird die Nachricht abgelegt.
Variable	Dieser Parameter enthält den Namen der Variable, für die ein Überwachungsauftrag erstellt werden soll.

4.9.2 MonitorSartResponse

Header	In diesem Bereich werden die Header abgelegt.
Body	In diesem Bereich wird die Nachricht abgelegt.
[StatusCode]	Dieser Parameter wird für die Übergabe einer Fehlerbeschreibung vom Server zum Client verwendet, wenn auf der Anwendungsebene ein Fehler aufgetreten ist. Ist kein Fehler aufgetreten so ist der Parameter nicht vorhanden.

4.9.3 MonitorStopRequest

Zum Stop einer Überwachung sendet die Webanwendung ein MonitorStopRequest an den OpcUaWebServer. Der MonitorStopRequest wird vom OpcUaWebServer mit einer MonitorStopResponse beantwortet.

Header	In diesem Bereich werden die Header Informationen abgelegt.
Body	In diesem Bereich wird die Nachricht abgelegt.
Variable	Dieser Parameter enthält den Namen der Variable, für die ein Überwachungsauftrag gelöscht werden soll.

4.9.4 MonitorStopResponse

Header	In diesem Bereich werden die Header abgelegt.
Body	In diesem Bereich wird die Nachricht abgelegt.
[StatusCode]	Dieser Parameter wird für die Übergabe einer Fehlerbeschreibung vom Server zum Client verwendet, wenn auf der Anwendungsebene ein Fehler aufgetreten ist. Ist kein Fehler aufgetreten so ist der Parameter nicht vorhanden.

4.9.5 MonitorUpdateMessage

Wird eine OPC UA Variable durch den OpcUaWebServer überwacht und ändert sich der Wert dieser OPC UA Variable, so sendet der OpcUaWebServer eine MonitorUpdate-Message an die Webanwendung.

Header	In diesem Bereich werden die Header abgelegt.
Body	In diesem Bereich wird die Nachricht abgelegt.
Variable	Dieser Parameter enthält den Namen dessen Wert bzw. Status sich geändert hat.
Value	Dieser Parameter enthält den Wert der Variable. Das Format ist Abhängig vom Datentyp der gelieferten Variable.
SourceTimestamp	Dieser Parameter enthält den SourceTimestamp der Variable. Als Format für die Zeitangabe wird ISO 8601 verwendet. Beispiel: "2015-09-06T09:03:21.237123"
ServerTimestamp	Dieser Parameter enthält den ServerTimestamp der Variable. Als Format für die Zeitangabe wird ISO 8601 verwendet. Beispiel: "2015-09-06T09:03:21.237123"