

WMS WebControl pro API

Dokumentation



Der SonnenLichtManager



Allgemeine Hinweise

Mit Herausgabe dieser Unterlage verlieren alle früheren entsprechenden Unterlagen ihre Gültigkeit. Änderungen, die der technischen Verbesserung dienen, sind vorbehalten. Text- und Grafikeil dieser Unterlage wurden mit besonderer Sorgfalt erstellt. Für möglicherweise trotzdem vorhandene Fehler und deren Auswirkungen kann keine Haftung übernommen werden.

Rechtliche Hinweise

- ▶ Bedienungsanleitungen, Handbücher und Software sind urheberrechtlich geschützt.
- ▶ Die Rechte an der Software sind in den mitgelieferten Lizenzbestimmungen festgelegt.
- ▶ WAREMA und das WAREMA Logo sind Warenzeichen der WAREMA Renkhoff SE.
- ▶ Alle anderen aufgeführten Marken- oder Produktnamen sind Warenzeichen oder eingetragene Warenzeichen der jeweiligen Inhaber.

Kontakt

Kundencenter Steuerungssysteme

Verkauf, Auftragsannahme und Anwendungstechnik

Tel. +49 9391 20-3750 • Fax -3759
info.steueringssysteme@warema.de

International

Tel. +49 9391 20-3740 • Fax -3749
steuerungssysteme.international@warema.de

Helpline Steuerungssysteme

Tel. +49 9391 20-9317 • Fax -6769
service@warema.de

Vetrieb Gebäudesystemtechnik

Dillberg 33, 97828 Marktheidenfeld
Tel. +49 9391 20-3720 • Fax -3719

Tel. +49 9391 20-3720 • Fax -3719

© 2024, WAREMA Renkhoff SE

Inhalts- verzeichnis

1	Sicherheitshinweise.....	5
1.1	Symbol- und Piktogrammerklärung.....	5
1.2	Bestimmungsgemäße Verwendung.....	6
1.3	Leser-Zielgruppe.....	6
1.4	Weitere Dokumente.....	6
1.5	Empfohlenes Betriebssystem.....	6
2	Allgemeines.....	7
2.1	Allgemeine Beschreibung der WMS WebControl pro API.....	7
3	Befehle der WMS WebControl pro API.....	8
3.1	getConfiguration.....	8
3.2	ping.....	11
3.3	action.....	11
3.4	sceneActions.....	12
3.5	getStatus.....	13
4	ActionType - Parameterauswahl.....	15
4.1	Percentage.....	15
4.2	Percentage Delta.....	15
4.3	Rotation.....	15
4.4	Rotation Delta.....	15
4.5	Switch.....	16
4.6	Toggle.....	16
4.7	Stop.....	16
4.8	Impulse.....	16
4.9	Identify.....	16
4.10	Enumeration.....	17
5	Parameterliste.....	18
6	Testen der WMS WebControl pro API.....	19
7	Aktuelle Protokollversion.....	19

1 Sicherheitshinweise

Wir haben die WMS Produkte und diese Software unter Beachtung der grundlegenden Sicherheitsanforderungen entwickelt und geprüft.

Trotzdem bestehen Restrisiken!

- Lesen Sie deshalb dieses Handbuch, bevor Sie die Steuerungen in Betrieb nehmen und bedienen!
- **Beachten Sie unbedingt die hier aufgeführten Sicherheitshinweise und die Warnhinweise in diesem Handbuch! Ansonsten erlischt jeglicher Gewährleistungsanspruch seitens des Herstellers!**
- Bewahren Sie dieses Handbuch für künftige Verwendung auf!

1.1 Symbol- und Piktogrammerklärung

Die Sicherheitshinweise in dieser Anleitung sind mit Warnsymbolen gekennzeichnet.



VORSICHT

mahnt zu **vorsichtigem Handeln**.

Die möglichen Folgen einer Nichtbeachtung können **Sachschäden** sein.



Das i-Symbol kennzeichnet wichtige **Hinweise** und hilfreiche **Tipps**.

- Das **Quadrat** kennzeichnet eine **Anweisung** oder eine **Handlungsaufforderung**. Führen Sie diesen Handlungsschritt aus!
- ▶ Das **Dreieck** kennzeichnet ein **Ereignis** oder ein **Resultat** einer vorangegangenen Handlung.
- ▶ Das **schwarze Dreieck** ist das **Aufzählungszeichen** für Listen oder Auswahlen.

1.2 Bestimmungsgemäße Verwendung

Die WMS Produkte ermöglichen den Anschluss unterschiedlicher Geräte wie Sonnenschutzprodukte und Beleuchtungseinrichtungen sowie von Fensterantrieben und Messwertgebern.



VORSICHT

Das WMS WebControl pro ist für den Innenbereich vorgesehen. Es ist nicht wasserdicht.



Die Reichweite von Funksteuerungen wird durch die gesetzlichen Bestimmungen für Funkanlagen und durch bauliche Gegebenheiten begrenzt. Bei der Projektierung muss auf einen ausreichenden Funkempfang geachtet werden. Dies gilt insbesondere dann, wenn das Funksignal durch Wände und Decken dringen muss. Die Steuerung sollte nicht in direkter Nähe metallischer Bauteile (Stahlträger, Stahlbeton, Brandschutztür) installiert werden.

- Prüfen Sie deshalb vor der endgültigen Montage/Aufstellung die Funktion der Anlage. Sowohl das WMS WebControl pro als auch die anzusteuernenden WMS Komponenten müssen einwandfreien Funkempfang haben.

1.3 Leser-Zielgruppe

Diese Anleitung wendet sich an Personen, welche Ihre Sonnenschutzprodukte über eine App oder einen Browser bedienen möchten.

1.4 Weitere Dokumente

Diese Anleitung enthält alle Informationen zur Konfiguration des WMS WebControl pro. Zusätzlich zu dieser Anleitung steht Ihnen folgendes Dokument zur Verfügung:

Dokument	Art.-Nr. / Link
Postman Collection	www.warema.de/webcontrol
WMS WebControl pro - Handbuch	2022743
WMS WebControl pro - Quick Start Guide	2022747
WMS studio pro - Software	www.warema.com/funksysteme-support

1.5 Empfohlenes Betriebssystem

Microsoft Betriebssystem:

Ab Windows7

2 Allgemeines

Die WMS WebControl pro API ist eine frei zugängliche Schnittstelle, die im lokalen Netzwerk via http nutzbar ist. Diese kann verwendet werden, um unterschiedliche Befehle an das WMS WebControl pro zu übermitteln. Es kann sich hierbei um Steuerbefehle, Konfigurationsabfragen oder auch Identifikationsabfragen handeln. Im nachfolgenden wird zuerst der allgemeine Aufbau eines solchen Befehls erläutert. Anschließend werden die unterschiedlichen Befehle detailliert erklärt und jeweils mit einem Anwendungsbeispiel untermauert. Voraussetzung für die Nutzung der Schnittstelle ist ein WMS WebControl pro, welches zuvor mit Hilfe der Software WMS studio pro parametrisiert und mit entsprechenden Kanälen, Szenen und Geräten versehen wurde.

2.1 Allgemeine Beschreibung der WMS WebControl pro API

Standardmäßig besteht der Request aus der Angabe des *command*, der *source* sowie der *protocolVersion*. Darüber hinaus wird der jeweilige Request noch um optionale Befehlsparameter erweitert (siehe Kapitel 3 auf Seite 8)

Der allgemeine Aufbau eines solchen Befehls sieht wie folgt aus:

```
{
  "protocolVersion": "<APIVersion>",
  "command": "<Name des Befehls>",
  "source" : <Die ID der Befehlsquelle>,
  <optionale Befehlsparameter>
}
```

Parameter	Beschreibung	Zusätzliche Informationen
<i>protocolVersion</i>	Aktuelle Version des verwendeten Protokolls	die Version ist Kapitel 7 auf Seite 19 zu entnehmen
<i>command</i>	Name des Befehls	Siehe Auflistung unterhalb
<i>source</i>	Die ID der Quelle, die die Befehle absetzt	Bei Nutzung der WMS WebControl pro API muss die sourceID immer zwingend mit 2 angegeben werden

Folgende Befehle können über die WMS WebControl pro API abgebildet werden:

- ▶ ping
- ▶ action
- ▶ sceneActions
- ▶ getConfiguration
- ▶ getStatus

3 Befehle der WMS WebControl pro API

Ein erfolgreicher Request wird vom WMS WebControl pro immer mit dem Header-Status `200 OK` bestätigt. Lediglich der Body unterscheidet sich abhängig von den unterschiedlichen Befehlen, die nachfolgend, unter der Verwendung von Beispielen, erläutert werden.

3.1 getConfiguration

Mithilfe dieses Befehls kann die Geräteliste des WMS WebControl pro ausgelesen werden. Es ergibt sich somit eine Auflistung des WMS WebControl pro-Systems, die die steuerbaren Kanäle und Szenen aufzeigt. Darüber hinaus können die jeweiligen Aktionen eines Kanals zurückgelesen werden sowie die zugehörigen Minimal- und Maximalwerte (*minValue & maxValue*). *GetConfiguration* wird unter anderem von den Befehlen *actions*, *sceneActions* sowie *getStatus* benötigt, um die jeweiligen IDs zu bestimmen. Diese Befehle werden in den folgenden Abschnitten noch genauer erläutert.

Die Anfrage der WMS WebControl pro-Geräteliste sieht wie folgt aus:

```
{
  "protocolVersion": "1.0",
  "command": "getConfiguration",
  "source" : 2
}
```

Darauf antwortet das WMS WebControl pro zum Beispiel mit (dieses Beispiel wird als Wegweiser für die Verwendung der jeweiligen IDs in den Befehlen: *actions*, *sceneActions* und *getStatus* verwendet):

```
{
  "command": "getConfiguration",
  "protocolVersion": "1.0.0",
  "destinations":
  [
    {
      "id": 17776,
      "animationType": 0,
      "names":
      [
        "Küche",
        "",
        "",
        ""
      ],
      "actions":
      [
        {
          "id": 0,
          "actionType": 0,
          "actionDescription": 2,
          "minValue": 0,
          "maxValue": 100
        },
        {
          "id": 6,
          "actionType": 2,
          "actionDescription": 3,
          "minValue": -127,
          "maxValue": 127
        },
        {
          "id": 16,
          "actionType": 6,
          "actionDescription": 12
        },
        {
          "id": 22,
          "actionType": 8,
          "actionDescription": 13
        },
        {
          "id": 23,
          "actionType": 7,
          "actionDescription": 12
        }
      ]
    },
    {
      "id": 200951,
      "animationType": 999,
      "names":
      [
        "Aktor Potentialfrei",
        "",
        "",
        ""
      ],
      "actions":
      [
        {
          "id": 22,
          "actionType": 8,
          "actionDescription": 13
        },

```

```

    {
      "id": 26,
      "actionType": 9,
      "actionDescription": 999,
      "minValue": 0,
      "maxValue": 16
    }
  ],
  "rooms":
  [
    {
      "id": 42581,
      "name": "Raum 0",
      "destinations":
      [
        17776,
        116682,
        194367,
        200951
      ],
      "scenes":
      [
        688966
      ]
    }
  ],
  "scenes":
  [
    {
      "id": 688966,
      "names":
      [
        "Gute Nacht",
        "",
        "",
        ""
      ]
    }
  ]
}

```

Beispiel *action*

Möchte man z.B. eine Aktion am Kanal *Küche* mit der *destinationID* 17776 durchführen, so kann die zugehörige *actionID* und *actionType* aus der Ausgabe der Konfiguration entnommen werden. In diesem Beispiel hat der Gerätekanal *Küche* fünf *actionIDs* (0, 6, 16, 22 und 23). Um auf dem Aktor Aktion 0 auszuführen, muss noch der zugehörige *actionType* herausgesucht werden. In diesem Fall handelt es sich um *actionType* 0. Aus der Parameterliste kann zurückgelesen werden, dass es sich um den *actionType Percentage* handelt. Anhand dieses *actionTypes* können die Parameter für die jeweilige Aktion festgelegt werden (siehe Kapitel 4 auf Seite 15).

Beispiel *sceneAction*

Ähnlich sieht das Vorgehen aus, wenn eine Szene ausgeführt werden soll. Hier muss auch die WMS WebControl pro-Konfiguration ausgelesen werden. In diesem Beispiel soll die Szene *Gute Nacht* ausgeführt werden. Hierfür benötigen wir die Szenen ID. Diese ist in diesem Fall 688966.

Beispiel *getStatus*

Möchte man den Status eines Gerätekanals abfragen, so muss *destinationID* dieses Kanals zurück gelesen werden. Der Kanal *Küche* hat zum Beispiel die *destinationID* 17776.

3.2 ping

Dieser Befehl kann zum Anfragen eines bestimmten WMS WebControl pro verwendet werden.

Der Aufbau des Befehls *ping* sieht wie folgt aus:

```
{
  "protocolVersion": "1.0",
  "command": "ping",
  "source": 2
}
```

In diesem Beispiel ist die aktuelle Protokollversion des WMS WebControl pro die Version 1.0 (siehe Kapitel 7 auf Seite 19). Die Source 2 ist die Standard Source die zu verwenden ist.

3.3 action

Hiermit kann ein Befehl über das WMS WebControl pro an ein oder mehrere Warema Produkte (kanalweise) im lokalen WMS Netz gesendet werden. Dabei kann es sich um einen Einzel- oder auch einen Sammelbefehl handeln. Im Gegensatz zu einem Einzelbefehl, welcher genau einen Aktionsbefehl pro Request enthält, kann ein Sammelbefehl zwei oder mehrere Aktionsbefehle pro Request enthalten. Die Befehle eines Sammelbefehls dürfen nur an einen gemeinsamen Kanal gesendet werden. Der allgemeine Aufbau eines solchen Befehls ist dem folgenden Quellcode zu entnehmen.

```
{
  "protocolVersion": "1.0",
  "command": "action",
  "source": 2,
  "responseType": <responseType>,
  "actions":
  [
    {
      "destinationId": <destinationId>,
      "actionId": <actionID>,
      "parameters":
      {
        <parameters>
      }
    }
  ]
}
```

Wie auch in den vorherigen Beispielen ist die Protokollversion für dieses Beispiel 1.0 sowie die Source 2. Darüber hinaus hat dieser Befehl noch weitere Befehl spezifische Parameter. Diese Parameter sind der Parameterliste in Kapitel 4 auf Seite 15 zu entnehmen.

Um die Verwendung dieses Befehls zu verdeutlichen wird ein kurzes Beispiel aufgezeigt.

Sollen z.B. Produkte in einem Kanal (in diesem Beispiel Kanal *Küche* mit der *destinationID* 17776) auf die Position 20% und die Neigung 70° gefahren sowie lediglich eine Bestätigung als Antwort gesendet werden, so würde der Befehl wie folgt aussehen.

```
{
  "protocolVersion": "1.0",
  "command": "action"
  "source": 2
  "responseType": 0
  "actions":
  [
    {
      "destinationId": 17776,
      "actionId": 0,
      "parameters":
      {
        "percentage": 20
      }
    },
    {
      "destinationId": 17776,
      "actionId": 6,
      "parameters":
      {
        "rotation": 70
      }
    }
  ]
}
```

Um die *destinationID* sowie die *actionID* zu erhalten, muss die Konfiguration des WMS WebControl pro über *getConfiguration* ausgelesen werden. Die Verwendung dieser IDs wird in Kapitel 3.1 auf Seite 8 unter *getConfiguration* erläutert.

3.4 sceneActions

In einer Szene können unterschiedliche Produkttypen gemeinsam bedient werden. Mit Hilfe dieses Befehls können Szenen ausgelöst oder auch neu eingelernt werden. Um eine Szene neu einzulernen, muss diese bereits im WMS WebControl pro existieren. Sie kann mit Hilfe des Einlernens mit neuen Parametern überschrieben werden. Dabei wird jeweils die aktuelle Position / Zustand des Produkts in die Szene übernommen. Es ist nicht möglich eine neue Szene anzulegen oder zu entfernen. Darüber hinaus können auch keine Geräte der Szene hinzugefügt oder gelöscht werden. Dies ist nur mit dem WMS studio pro möglich.

```
{
  "protocolVersion": "1.0",
  "command": "sceneAction",
  "source": 2,
  "responseType": 0,
  "sceneId": 688966,
  "sceneActionType": 1
}
```

Hier wird eine Szene (mit der *sceneID* 688966, die über den Befehl *getConfiguration* ausgelesen werden kann), ausgeführt. Hierbei ist der *sceneActionType* 1. Somit wird dies Szene mit Hilfe dieses Befehls ausgeführt (siehe Parameterliste, Kapitel 5 auf Seite 18).

Die Verwendung dieser IDs wird in Kapitel 3.1 auf Seite 8 unter *getConfiguration* erläutert.

3.5 getStatus

Dieser Befehl kann den aktuellen Status eines Gerätekanals / Zieles abfragen. Es können nicht mehrere Gerätekanäle gleichzeitig angefragt werden. Der Aufbau des Befehls sieht wie folgt aus:

```
{
  "protocolVersion": 1.0,
  "command": "getStatus",
  "source": 2,
  "destinations": [ <destinationID> ]
}
```

Die *destinationID* kann ebenfalls mit `getConfiguration` ausgelesen werden. Die Verwendung dieser IDs wird in Kapitel 3.1 auf Seite 8 unter `getConfiguration` erläutert. Für die ID 17776 würde der Befehl folgendermaßen aussehen.

```
{
  "protocolVersion": 1.0,
  "command": "getStatus",
  "source": 2,
  "destinations": [ 17776 ]
}
```

Darauf antwortet das WMS WebControl pro mit zum Beispiel:

```
{
  "protocolVersion": "1.0.0",
  "command": "getStatus",
  "details":
  [
    {
      "destinationId": 17776,
      "data":
      {
        "drivingCause": 5,
        "heartbeatError": false,
        "blocking": false,
        "productData":
        [
          {
            "actionId": 0,
            "value":
            {
              "percentage": 127.5
            }
          },
          {
            "actionId": 6,
            "value":
            {
              "rotation": -75
            }
          },
          {
            "actionId": 23,
            "value":
            {
              "rotation": -75
            }
          }
        ]
      }
    }
  ]
}
```

Für eine verständliche Interpretation dieser Antwort wird der zurückgelieferte Parameter Driving Cause erläutert.

In diesem Beispiel besitzt der zuletzt ausgeführte Driving Cause (Fahrursache für letzte Positionsänderung) die ID 5. Blickt man nun auf die Parameterliste aus Kapitel 5 auf Seite 18, so kann man diese IDs den jeweiligen Fahrursachen zuordnen. Die ID 5 kann dem Eis Alarm zugeordnet werden.

4 ActionType - Parameterauswahl

Anhand der *ActionTypes* werden die Parameter für die jeweiligen Actions bestimmt.

Die zugehörigen Minimal- und Maximalwerte können über *getConfiguration* ausgelesen werden (siehe Kapitel 3 auf Seite 8).

4.1 Percentage

Hier wird ein absoluter Prozentwert, zum Beispiel als Position, übermittelt.

Minimalwert: 0%, Maximalwert 100%

Beispiel „Fahre Produkt auf absolut 5%“

```
{  
  "percentage": 5  
}
```

4.2 Percentage Delta

Hier wird ein Prozentwert als relativer Wert übermittelt, um z.B. abhängig von der aktuellen Position x% aufzufahren oder zuzufahren.

Minimalwert: -100%, Maximalwert: 100%

Beispiel „Bewege Produkt um 5% auf“

```
{  
  "percentageDelta": -5  
}
```

4.3 Rotation

Dieser Parameter wird verwendet, um die Neigung eines Produktes einzustellen. Es handelt sich hierbei um einen absoluten Wert.

Minimalwert: -360°, Maximalwert: 360°

Beispiel „Stelle die Neigung des Produkts auf -100°“

```
{  
  "rotation": -100  
}
```

4.4 Rotation Delta

Dieser Parameter wird verwendet, um einen relativen Wert, für die Neigung eines Produktes zu übermitteln.

Minimalwert: -720°, Maximalwert: 720°

Beispiel „Wende das Produkt um 50° auf“

```
{  
  "rotationDelta": -50  
}
```

4.5 Switch

Dient zum Ein- bzw. Ausschalten eines Produktes.
Werte: True (Einschalten), False (Ausschalten)
Beispiel „Schalte das Produkt ein“

```
{  
  "onOffState": true  
}
```

4.6 Toggle

Hiermit kann ein schaltbares Gerät umgeschaltet werden. Zum Beispiel um ein Produkt zwischen Zustand „Aus“ und „An“ umzuschalten. Dieser *ActionType* benötigt keine weiteren Parameter.

4.7 Stop

Dies unterbricht einen aktuell ausgeführten Befehl. Auch dieser *ActionType* benötigt keine weiteren Parameter.

4.8 Impulse

Mithilfe dieses Befehls wird ein kurzer Impulsbefehl gesendet.
Werte: 0 (Impulse Up), 1 (Impulse Down)
Beispiel "Bewege das Produkt ein Stück nach oben"

```
{  
  "impulse": 0  
}
```

4.9 Identify

Löst eine Aktion zum Identifizieren des jeweiligen Ziels aus (z.B. „Winken“). Die Art der Identifikation hängt vom Produkttypen ab. Dies kann z.B. kurzes auf- und zufahren bedeuten. Dieser *ActionType* benötigt keine weiteren Parameter.

4.10 Enumeration

Wird verwendet, um die aktuell auszuführende Aktion eines potentialfreien Aktors auszuwählen bzw. mit dem Befehl *action* aufzuführen.
Minimalwert 0, Maximalwert: 16

Es kann zwischen folgenden Values gewählt werden:

Aktion	Wert
Keine Aktion	0
Laufzeit Hoch kurz	1
Laufzeit Tief kurz	2
Laufzeit Hoch lang	3
Laufzeit Tief lang	4
Dauerhoch	5
Dauertief	6
Laufzeit Toggle Hoch	7
Laufzeit Toggle Tief	8
Hoch und Tief aus	9
Laufzeit Toggle Hoch und Tief gleichzeitig	10
Dauerhoch und Dauertief gleichzeitig	11
Hoch aus	12
Tief aus	13
Laufzeit kurz in Gegenrichtung	14
Hoch toggeln	15
Tief toggeln	16

Beispiel „Bewege ein Produkt (über den Aktor Potentialfrei) mit der Aktion Dauerhoch“

```
{  
  "enumeration": 5  
}
```

5 Parameterliste

Dieser Abschnitt erläutert die Parameter, die für die Befehle zur Verfügung stehen oder auch mit Hilfe der Befehle ausgelesen werden können.

Source

2 = Standardsource (immer zu verwenden)

AnimationType

0 = Venetian blind
1 = Awning
2 = Roller shutter / blind
3 = Slat roof
4 = Window
5 = Switch
6 = Dimmer
999 = Unknown

ActionType

0 = Percentage
1 = PercentageDelta
2 = Rotation
3 = RotationDelta
4 = Switch
5 = Toggle
6 = Stop
7 = Impulse
8 = Identify
9 = Enumeration
999 = Unknown

ActionDescription:

0 = Awning drive
1 = Valance drive
2 = Slat drive
3 = Slat rotate
4 = Roller shutter / blind drive
5 = Window drive
6 = Light switch
7 = Load switch
8 = Light dimming
9 = Load dimming
10 = Light toggle
11 = Last toggle
12 = Manual Command
13 = Identify
999 = Unknown

ResponseType

0 = Instant (Antwort senden sobald das WMS WebControl pro den Befehl empfangen hat)

1 = Detailed (Optional; Antwort mit detaillierten Informationen senden, sobald die Informationen zusammengestellt wurden. Unterstützt das Gerät diesen ResponseType nicht, so findet automatisch eine „Instant“ Response statt.)

SceneActionType

0 = Relearn

1 = Execute

DrivingCause

1 = Sun

2 = Dusk/Dawn

3 = Wind

4 = Rain

5 = Ice

6 = Temperature

7 = Switching Time

8 = Scene

9 = Control Mode

10 = Manual

11 = Safety

12 = Contact

13 = Central Command

999 = Unknown

6 Testen der WMS WebControl pro API

Die zuvor vorgestellte API kann z.B. mit dem Tool Postman getestet werden. Bei Postman handelt es sich um ein Softwaretool, welches zum Bauen und Verwenden von API's eingesetzt werden kann und in einer freien Testversion verfügbar ist.

Folgender Link

www.warema.de/webcontrol

beinhaltet unter dem Abschnitt "Dokumente & Downloads" eine Postman Collection. Hier finden Sie Beispiele von Befehlen, die zum Testen und Verstehen der WMS WebControl pro API verwendet werden können. Diese Befehle sind anhand von Beispielen verdeutlicht.



Die Datei "WMS_WCP_API.postman_collection.json" ist gezippt. Das Zip-Archiv "webcontrolpro_api_postman_collection.zip" enthält die zu importierende Postman Collection Datei "webcontrolpro_api.postman_collection.json". Bitte verwenden Sie ein entsprechendes Tool zum Entpacken.

7 Aktuelle Protokollversion

Die WMS WebControl pro Schnittstelle kann ab der Containerversion: „11H“ verwendet werden.

Die aktuelle Protokollversion lautet: 1.0

