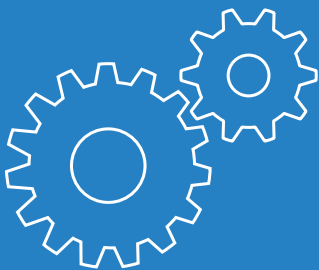


LANCOM Wireless ePaper Server

API-Referenz



Inhalt

1	AccessPointInfoResource.....	3
2	AccessPointResource.....	8
3	ConfigurationResource.....	11
4	LabelInfoResource.....	15
5	LabelResource.....	26
6	Level1ExportResource.....	30
7	Level1ImportResource.....	34
8	LicenseResource.....	39
9	ProblemResource.....	42
10	ServiceStatusResource.....	44
11	TagResource.....	45
12	TaskResource.....	48
13	TemplateResource.....	53
14	TransactionResource.....	55
15	UpdateStatusResource.....	57

1 AccessPointInfoResource

Diese Ressource liefert Statusinformationen zu allen Access Points, die im Server registriert sind. Die Statusinformationen enthalten Details wie Access Point-ID, IP-Adresse, verwendeter Funkkanal und Verbindungsstatus.

Table 1: Übersicht

Methode	URL	Name
GET	/service/accesspointinfo	getAccessPoints
GET	/service/accesspointinfo/channelconflict	getAccessPointsWithChannelConflict
GET	/service/accesspointinfo/status/{connectionStatus}	getAccessPointsWithConnectionStatus
GET	/service/accesspointinfo/nochannel	getAccessPointsWithoutChannel

getAccessPoints()

Gibt Statusinformationen zu allen registrierten Access Points zurück.

Typ	Request	
Methode:	GET	
URL:	/service/accesspointinfo?page={page}&recordsPerPage={recordsPerPage}	
Antwort:	ESLAccessPoint PagedResult	Ergebnisliste von ESLAccessPoint.
	Typ	XML-Typ
	Definition und Beispiel	siehe ESLAccessPointPagedResult
Parameter:	page	Die Seite der Ergebnisliste.
	Stil	QUERY
	Erforderlich	Nein
	Typ	Integer
	recordsPerPage	Die maximale Anzahl von Einträgen pro Seite, welche zurückgegeben werden sollen.
	Stil	QUERY
	Erforderlich	Nein
	Typ	Integer

getAccessPointsWithChannelConflict()

Gibt Statusinformationen zu allen registrierten Access Points mit Kanalkonflikten zurück. Ein Kanalkonflikt liegt vor, wenn ein Kanal auf mehr als einem Access Point ausgewählt ist.

Typ	Request
-----	---------

Methode:	GET	
URL:	/service/accesspointinfo/channelconflict?page={page}&recordsPerPage={recordsPerPage}	
Antwort:	ESLAccessPoint PagedResult	Ergebnisliste von ESLAccessPoint.
	Typ	XML-Typ
	Definition und Beispiel	siehe ESLAccessPointPagedResult
Parameter:	page	Die Seite der Ergebnisliste.
	Stil	QUERY
	Erforderlich	Nein
	Typ	Integer
	recordsPerPage	Die maximale Anzahl von Einträgen pro Seite, welche zurückgegeben werden sollen.
	Stil	QUERY
	Erforderlich	Nein
	Typ	Integer

getAccessPointsWithConnectionStatus()

Gibt nach Verbindungsstatus gefilterte Statusinformationen zu allen registrierten Access Points zurück.

Typ	Request	
Methode:	GET	
URL:	/service/accesspointinfo/status/{connectionStatus}?page={page}&recordsPerPage={recordsPerPage}	
Antwort:	ESLAccessPoint PagedResult	Ergebnisliste von ESLAccessPoint.
	Typ	XML-Typ
	Definition und Beispiel	siehe ESLAccessPointPagedResult
Parameter:	connectionStatus	Der zu filternde Verbindungsstatus.
	Stil	PATH
	Erforderlich	Ja
	Typ	ConnectionStatus
	page	Die Seite der Ergebnisliste.
	Stil	QUERY
	Erforderlich	Nein
	Typ	Integer

recordsPerPage	Die maximale Anzahl von Einträgen pro Seite, welche zurückgegeben werden sollen.
Stil	QUERY
Erforderlich	Nein
Typ	Integer

getAccessPointsWithoutChannel()

Gibt Statusinformationen für alle registrierten Access Points zurück, bei denen kein Kanal ausgewählt ist.

Typ	Request	
Methode:	GET	
URL:	/service/accesspointinfo/nochannel?page={page}&recordsPerPage={recordsPerPage}	
Antwort:	ESLAccessPointPagedResult	Ergebnisliste von ESLAccessPoint.
	Typ	XML-Typ
	Definition und Beispiel	siehe ESLAccessPointPagedResult
Parameter:	page	Die Seite der Ergebnisliste.
	Stil	QUERY
	Erforderlich	Nein
	Typ	Integer
	recordsPerPage	Die maximale Anzahl von Einträgen pro Seite, welche zurückgegeben werden sollen.
	Stil	QUERY
	Erforderlich	Nein
	Typ	Integer

ESLAccessPointPagedResult

Ergebnisliste von ESLAccessPoint.

Typ: XML-Typ

Schema:

```
<xs:schema version="1.0">
  <xs:element name="AccessPoint" type="AccessPoint"/>
  <xs:element name="AccessPointPagedResult" type="AccessPointPagedResult"/>
  <xs:element name="PagedResult" type="PagedResult"/>
  <xs:complexType name="AccessPointPagedResult">
    <xs:complexContent>
      <xs:extension base="PagedResult">
        <xs:sequence>
          <xs:element ref="AccessPoint" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
```

```

<xs:complexType name="PagedResult" abstract="true">
  <xs:sequence/>
  <xs:attribute name="records" type="xs:int" use="required"/>
  <xs:attribute name="totalRecords" type="xs:int" use="required"/>
  <xs:attribute name="totalPages" type="xs:int" use="required"/>
  <xs:attribute name="page" type="xs:int" use="required"/>
  <xs:attribute name="recordsPerPage" type="xs:int" use="required"/>
</xs:complexType>
<xs:complexType name="AccessPoint">
  <xs:sequence>
    <xs:element name="Name" type="xs:string" minOccurs="0"/>
    <xs:element name="AccessPointId" type="xs:int"/>
    <xs:element name="Address" type="xs:string" minOccurs="0"/>
    <xs:element name="Channel" type="xs:int" minOccurs="0"/>
    <xs:element name="ConnectionStatus" type="connectionStatus" minOccurs="0"/>
    <xs:element name="Version" type="xs:string" minOccurs="0"/>
    <xs:element name="UpdateTime" type="xs:dateTime" minOccurs="0"/>
    <xs:element name="AutoConfig" type="xs:boolean" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:simpleType name="connectionStatus">
  <xs:restriction base="xs:string">
    <xs:enumeration value="ONLINE"/>
    <xs:enumeration value="DISCOVERED"/>
    <xs:enumeration value="UNKNOWN"/>
    <xs:enumeration value="OFFLINE"/>
  </xs:restriction>
</xs:simpleType>
</xs:schema>

```

Beispiel:

```

<AccessPointPagedResult records="1" totalRecords="1" totalPages="1" page="0"
recordsPerPage="1">
  <AccessPoint>
    <AccessPoint>
      <AccessPointId>10078</AccessPointId>
      <Address>http://localhost:8080</Address>
      <Channel>9</Channel>
      <ConnectionStatus>ONLINE</ConnectionStatus>
      <Version>1.1.0</Version>
      <UpdateTime>2014-07-22T08:42:36.993+02:00</UpdateTime>
      <Name>192.168.1.1</Name>
    </AccessPoint>
  </AccessPointPagedResult>

```

ConnectionStatus

Der zu filternde Verbindungsstatus.

Typ: Aufzählungstyp

Schema:

```

<xs:schema version="1.0">
  <xs:simpleType name="connectionStatus">
    <xs:restriction base="xs:string">
      <xs:enumeration value="ONLINE"/>
      <xs:enumeration value="DISCOVERED"/>
      <xs:enumeration value="UNKNOWN"/>
      <xs:enumeration value="OFFLINE"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>

```

Aufzählungswerte:

Der Aufzählungswert kann als Pfad oder Abfrageparameter verwendet werden.

Wert	Beschreibung
ONLINE	Gerät ist verbunden
UNKNOWN	Konnte den Verbindungsstatus nicht bestimmen
OFFLINE	Gerät ist nicht verbunden

2 AccessPointResource

Diese Ressource verwaltet die Access Points. Es können bereits registrierte Access Points angezeigt, neue Access Points registriert, die Registrierung zurückgezogen und die Konfiguration eines Access Point aktualisiert werden.

Table 2: Übersicht

Method	URL	Name
GET	/service/accesspoint	getRegisteredAccessPoints
POST	/service/accesspoint	registerAccessPoint
DELETE	/service/accesspoint/{accessPointID}	unregisterAccessPoint

getRegisteredAccessPoints()

Gibt alle registrierten Access Points zurück.

Typ	Request	
Methode:	GET	
URL:	/service/accesspoint?page={page}&recordsPerPage={recordsPerPage}	
Antwort:	ESLAccessPointRegistrationPagedResult	Ergebnisliste von ESLAccessPointRegistration.
	Typ	XML-Typ
	Definition und Beispiel	siehe ESLAccessPointRegistrationPagedResult
Parameter:	page	Die Seite der Ergebnisliste.
	Stil	QUERY
	Erforderlich	Nein
	Typ	Integer
	recordsPerPage	Die maximale Anzahl von Einträgen pro Seite, welche zurückgegeben werden soll.
	Stil	QUERY
	Erforderlich	Nein
	Typ	Integer

registerAccessPoints()

Registriert neue Access Points. Dafür werden Informationen wie die ID, die Serveradresse des Access Points benötigt.

Typ	Request
Methode:	POST

URL:	/service/accesspoint	
Anfrage:	ESLAccessPoint Registrierung	Access Point Registrierung
	Typ	XML-Typ
	Definition und Beispiel	siehe ESLAccessPointRegistrierung
Antwort:	Typ	void

unregisterAccessPoints()

Zieht die Registrierung eines Access Points zurück, der durch die accessPointID angegeben ist.

Typ	Request	
Methode:	DELETE	
URL:	/service/accesspoint/{accessPointId}	
Antwort:	Typ	void
Parameter:	accessPointId	Die ID des Access Points
	Stil	PATH
	Erforderlich	Ja
	Typ	Integer

ESLAccessPointRegistrationPagedResult

Ergebnisliste der ESLAccessPointRegistration-Einträge.

Typ: XML-Typ

Schema:

```
<xs:schema version="1.0">
  <xs:element name="AccessPointRegistration" type="AccessPointRegistration"/>
  <xs:element name="AccessPointRegistrationPagedResult"
type="AccessPointRegistrationPagedResult"/>
  <xs:element name="PagedResult" type="PagedResult"/>
  <xs:complexType name="AccessPointRegistrationPagedResult">
    <xs:complexContent>
      <xs:extension base="PagedResult">
        <xs:sequence>
          <xs:element ref="AccessPointRegistration"maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="PagedResult" abstract="true">
    <xs:sequence/>
    <xs:attribute name="records" type="xs:int" use="required"/>
    <xs:attribute name="totalRecords" type="xs:int" use="required"/>
    <xs:attribute name="totalPages" type="xs:int" use="required"/>
    <xs:attribute name="page" type="xs:int" use="required"/>
    <xs:attribute name="recordsPerPage" type="xs:int" use="required"/>
  </xs:complexType>
</xs:schema>
```

```

</xs:complexType><xs:complexType name="AccessPointRegistration">
  <xs:sequence>
    <xs:element name="AccessPointId" type="xs:int"/>
    <xs:element name="ServiceAddress" type="xs:string" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
</xs:schema>

```

Beispiel:

```

<AccessPointRegistrationPagedResult records="1" totalRecords="1" totalPages="1" page="0"
recordsPerPage="1">
  <AccessPointRegistration>
    <AccessPointId>10078</AccessPointId>
    <ServiceAddress>http://192.168.1.200:8080/</ServiceAddress>
  </AccessPointRegistration>
</AccessPointRegistrationPagedResult>

```

ESLAccessPointRegistration

Konfigurationsinformationen zum Access Point.

Typ: XML-Typ

Schema:

```

<xs:schema version="1.0">
  <xs:element name="AccessPointRegistration" type="AccessPointRegistration"/>
  <xs:complexType name="AccessPointRegistration">
    <xs:sequence>
      <xs:extension base="PagedResult">
        <xs:element name="AccessPointId" type="xs:int"/>
        <xs:element name="ServiceAddress" type="xs:string" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
  </xs:schema>

```

Beispiel:

```

<AccessPointRegistration>
  <AccessPointId>10078</AccessPointId>
  <ServiceAddress>http://192.168.1.200:8080/</ServiceAddress>
</AccessPointRegistration>

```

3 ConfigurationResource

Diese Ressource verwaltet die Konfigurationen. Es können Konfigurationen einzeln oder insgesamt abgerufen werden. Außerdem können veränderte Konfigurationen abgerufen und neue Konfigurationen gesetzt werden.

Table 3: Übersicht

Methode	URL	Name
GET	/service/configuration	getConfiguration
GET	/service/configuration/{key}	getConfiguration
GET	/service/configuration/modified	getModifiedConfiguration
PUT	/service/configuration/{key}	setConfiguration

get Configuration()

Gibt Informationen über Konfigurationen zurück. Die Informationen zu einer Konfiguration sind der Schlüssel, der Typ, der Defaultwert und der aktuell gesetzte Wert.

Typ	Request	
Methode:	GET	
URL:	/service/configuration?page={page}&recordsPerPage={recordsPerPage}	
Antwort:	ESLConfiguration PagedResult	Ergebnisliste der ESLConfiguration-Einträge.
	Typ Definition und Beispiel	XML-Typ siehe ESLConfigurationPagedResult
Parameter:	page Stil Erforderlich Typ	Die Seite der Ergebnisliste. QUERY Nein Integer
	recordsPerPage Stil Erforderlich Typ	Die maximale Anzahl von Einträgen pro Seite, welche zurückgegeben werden soll. QUERY Nein Integer

getConfiguration()

Gibt Informationen über eine Konfiguration zurück, die durch den Schlüssel (key) festgelegt wird.

Typ	Request
-----	---------

3 ConfigurationResource

Methode:	GET	
URL:	/service/configuration/{key}	
Antwort:	ESLConfiguration Typ Definition und Beispiel	Konfiguration mit Angaben des Schlüssels, des Datentyps, des Defaultwerts, des gesetzten Wertes und der erlaubten Datentypen. XML-Typ siehe ESLConfiguration
Parameter:	key Stil Erforderlich Typ	Schlüssel des Attributs PATH Ja String

getModifiedConfiguration()

Gibt Informationen über veränderte Konfigurationen zurück.

Typ	Request	
Methode:	GET	
URL:	/service/configuration/modified?page={page}&recordsPerPage={recordsPerPage}	
Antwort:	ESLConfiguration PagedResult Typ Definition und Beispiel	Ergebnisliste der ESLConfiguration-Einträge. XML-Typ siehe ESLConfigurationPagedResult
Parameter:	page Stil Erforderlich Typ	Die Seite der Ergebnisliste. QUERY Nein Integer
	recordsPerPage Stil Erforderlich Typ	Die maximale Anzahl von Einträgen pro Seite, welche zurückgegeben werden soll. QUERY Nein Integer

setConfiguration()

Setzt eine Konfiguration, die über den Schlüssel (key) festgelegt wird, auf einen anderen Wert (value).

Typ	Request
Methode:	PUT

URL:	/service/configuration/{key}?value={value}	
Antwort:	Typ	void
Parameter:	key	Schlüssel der Konfiguration
	Stil	QUERY
	Erforderlich	Ja
	Typ	Integer
	value	Wert, der gesetzt werden soll
	Stil	QUERY
	Erforderlich	Nein
	Typ	Integer

ESLConfigurationPagedResult

Ergebnisliste der ESLConfiguration-Einträge.

Typ: XML-Typ

Schema:

```
<xs:schema version="1.0">
  <xs:element name="Configuration" type="Configuration"/>
  <xs:element name="ConfigurationPagedResult" type="ConfigurationPagedResult"/>
  <xs:element name="PagedResult" type="PagedResult"/>
  <xs:complexType name="ConfigurationPagedResult">
    <xs:complexContent>
      <xs:extension base="PagedResult">
        <xs:sequence>
          <xs:element ref="Configuration" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="PagedResult" abstract="true">
    <xs:sequence/>
    <xs:attribute name="records" type="xs:int" use="required"/>
    <xs:attribute name="totalRecords" type="xs:int" use="required"/>
    <xs:attribute name="totalPages" type="xs:int" use="required"/>
    <xs:attribute name="page" type="xs:int" use="required"/>
    <xs:attribute name="recordsPerPage" type="xs:int" use="required"/>
  </xs:complexType>
  <xs:complexType name="Configuration">
    <xs:sequence/>
    <xs:attribute name="key" type="xs:string" use="required"/>
    <xs:attribute name="type" type="basicType"/>
    <xs:attribute name="defaultValue" type="xs:string"/>
    <xs:attribute name="value" type="xs:string"/>
  </xs:complexType>
  <xs:simpleType name="basicType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="BOOLEAN"/>
      <xs:enumeration value="INTEGER"/>
      <xs:enumeration value="FLOAT"/>
      <xs:enumeration value="STRING"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

```

    </xs:restriction>
  </xs:simpleType>
</xs:schema>

```

Beispiel:

```

<ConfigurationPagedResult records="1" totalRecords="1" totalPages="1" page="0"
recordsPerPage="1">
  <Configuration key="transmissionRetriesCount" type="INTEGER" defaultValue="5" value="3"/>
</ConfigurationPagedResult>

```

ESLConfiguration

Konfiguration mit Angaben des Schlüssels, des Datentyps, des Defaultwerts, des gesetzten Wertes und der erlaubten Datentypen.

Typ: XML-Typ

Schema:

```

<xs:schema version="1.0">
  <xs:element name="Configuration" type="Configuration"/>
  <xs:complexType name="Configuration">
    <xs:sequence/>
    <xs:attribute name="key" type="xs:string" use="required"/>
    <xs:attribute name="type" type="basicType"/>
    <xs:attribute name="defaultValue" type="xs:string"/>
    <xs:attribute name="value" type="xs:string"/>
  </xs:complexType>
  <xs:simpleType name="basicType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="BOOLEAN"/>
      <xs:enumeration value="INTEGER"/>
      <xs:enumeration value="FLOAT"/>
      <xs:enumeration value="STRING"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>

```

Beispiel:

```

<Configuration key="transmissionRetriesCount" type="INTEGER" defaultValue="5" value="3"/>

```

4 LabelInfoResource

Diese Ressource liefert Statusinformationen für alle Displays, die dem System bekannt sind.

Table 4: Übersicht

Methode	URL	Name
GET	/service/labelinfo/{label}	getLabel
GET	/service/labelinfo	getLabels
GET	/service/labelinfo/status/{connectionStatus}	getLabelsWithConnectionStatus
GET	/service/labelinfo/error	getLabelsWithErrors
GET	/service/labelinfo/powerstatus/{powerStatus}	getLabelsWithPowerStatus
GET	/service/labelinfo/problem	getLabelsWithProblems
GET	/service/labelinfo/tag/{tag}	getLabelsWithTag
GET	/service/labelinfo/unsuccessful	getLabelsWithUnsuccessfullastUpdate
GET	/service/labelinfo/unencrypted	getLabelsWithoutEncryption

getLabel()

Gibt Statusinformationen für ein Display zurück.

Typ	Request	
Methode:	GET	
URL:	/service/labelinfo/{label}	
Antwort:	ESLLabelInfo	Informationen zu den Labeln
	Typ	XML-Typ
	Definition und Beispiel	siehe ESLLabelInfo
Parameter:	label	Die Label-ID, mit der das entsprechende Display im Wireless ePaper Server hinterlegt ist.
	Stil	PATH
	Erforderlich	Ja
	Typ	String

getLabels()

Gibt Statusinformationen für alle bekannten Displays zurück.

Typ	Request
Methode:	GET

URL:	/service/labelinfo?page={page}&recordsPerPage={recordsPerPage}	
Antwort:	ESLabelInfo PagedResult	Ergebnisliste der ESLabelInfo-Einträge.
	Typ	XML-Typ
	Definition und Beispiel	siehe ESLabelInfoPagedResult
Parameter:	page	Die Seite der Ergebnisliste.
	Stil	QUERY
	Erforderlich	Nein
	Typ	Integer
	recordsPerPage	Die maximale Anzahl von Einträgen pro Seite, welche zurückgegeben werden soll.
	Stil	QUERY
	Erforderlich	Nein
	Typ	Integer

getLabelsWithConnectionStatus()

Gibt Statusinformationen zu allen bekannten Displays mit vorhandenem Verbindungsstatus an.

Typ	Request	
Methode:	GET	
URL:	/service/labelinfo/status/{connectionStatus}?page={page}&recordsPerPage={recordsPerPage}	
Antwort:	ESLabelInfo PagedResult	Ergebnisliste der ESLabelInfo-Einträge.
	Typ	XML-Typ
	Definition und Beispiel	siehe ESLabelInfoPagedResult
Parameter:	connectionStatus	Der zu filternde Verbindungsstatus.
	Stil	PATH
	Erforderlich	Ja
	Typ	ConnectionStatus
	page	Die Seite der Ergebnisliste.
	Stil	QUERY
	Erforderlich	Nein
	Typ	Integer
	recordsPerPage	Die maximale Anzahl von Einträgen pro Seite, welche zurückgegeben werden soll.

Stil	QUERY
Erforderlich	Nein
Typ	Integer

getLabelsWithErrors()

Gibt Statusinformationen für alle bekannten Displays mit Problemen zurück.

Typ	Request	
Methode:	GET	
URL:	/service/labelinfo/error?page={page}&recordsPerPage={recordsPerPage}	
Antwort:	ESLabelInfo PagedResult	Ergebnisliste der ESLabelInfo-Einträge.
	Typ	XML-Typ
	Definition und Beispiel	siehe ESLabelInfoPagedResult
Parameter:	page	Die Seite der Ergebnisliste.
	Stil	QUERY
	Erforderlich	Nein
	Typ	Integer
	recordsPerPage	Die maximale Anzahl von Einträgen pro Seite, welche zurückgegeben werden soll.
	Stil	QUERY
Erforderlich	Nein	
Typ	Integer	

getLabelsWithPowerStatus()

Gibt Statusinformationen für alle bekannten Displays mit vorhandenem Power-Status zurück.

Typ	Request	
Methode:	GET	
URL:	/service/labelinfo/powerstatus/{powerStatus}?page={page}&recordsPerPage={recordsPerPage}	
Antwort:	ESLabelInfo PagedResult	Ergebnisliste der ESLabelInfo-Einträge.
	Typ	XML-Typ
	Definition und Beispiel	siehe ESLabelInfoPagedResult
Parameter:	powerStatus	Gibt den Power-Status eines Displays an. Wenn das Display batteriebetrieben ist, wird hier die verbleibende Kapazität der Batterie angezeigt.

4 LabelInfoResource

Stil	PATH
Erforderlich	Ja
Typ	PowerStatus
page	Die Seite der Ergebnisliste.
Stil	QUERY
Erforderlich	Nein
Typ	Integer
recordsPerPage	Die maximale Anzahl von Einträgen pro Seite, welche zurückgegeben werden soll.
Stil	QUERY
Erforderlich	Nein
Typ	Integer

getLabelsWithProblems()

Gibt Statusinformationen für alle bekannten Displays mit Problemen zurück.

Typ	Request																
Methode:	GET																
URL:	<code>/service/labelinfo/problem?page={page}&recordsPerPage={recordsPerPage}</code>																
Antwort:	<table border="1"> <tr> <td>ESLLabelInfo PagedResult</td> <td>Ergebnisliste der ESLLabelInfo-Einträge.</td> </tr> <tr> <td>Typ</td> <td>XML-Typ</td> </tr> <tr> <td>Definition und Beispiel</td> <td>siehe ESLLabelInfoPagedResult</td> </tr> </table>	ESLLabelInfo PagedResult	Ergebnisliste der ESLLabelInfo-Einträge.	Typ	XML-Typ	Definition und Beispiel	siehe ESLLabelInfoPagedResult										
ESLLabelInfo PagedResult	Ergebnisliste der ESLLabelInfo-Einträge.																
Typ	XML-Typ																
Definition und Beispiel	siehe ESLLabelInfoPagedResult																
Parameter:	<table border="1"> <tr> <td>page</td> <td>Die Seite der Ergebnisliste.</td> </tr> <tr> <td>Stil</td> <td>QUERY</td> </tr> <tr> <td>Erforderlich</td> <td>Nein</td> </tr> <tr> <td>Typ</td> <td>Integer</td> </tr> <tr> <td>recordsPerPage</td> <td>Die maximale Anzahl von Einträgen pro Seite, welche zurückgegeben werden soll.</td> </tr> <tr> <td>Stil</td> <td>QUERY</td> </tr> <tr> <td>Erforderlich</td> <td>Nein</td> </tr> <tr> <td>Typ</td> <td>Integer</td> </tr> </table>	page	Die Seite der Ergebnisliste.	Stil	QUERY	Erforderlich	Nein	Typ	Integer	recordsPerPage	Die maximale Anzahl von Einträgen pro Seite, welche zurückgegeben werden soll.	Stil	QUERY	Erforderlich	Nein	Typ	Integer
page	Die Seite der Ergebnisliste.																
Stil	QUERY																
Erforderlich	Nein																
Typ	Integer																
recordsPerPage	Die maximale Anzahl von Einträgen pro Seite, welche zurückgegeben werden soll.																
Stil	QUERY																
Erforderlich	Nein																
Typ	Integer																

getLabelsWithTag()

Gibt Statusinformationen für alle bekannten Displays mit vorhandenen Tags zurück.

Typ	Request
-----	---------

Methode:	GET	
URL:	/service/labelinfo/tag/{tag}?page={page}&recordsPerPage={recordsPerPage}	
Antwort:	ESLabelInfo PagedResult Typ Definition und Beispiel	Ergebnisliste der ESLabelInfo-Einträge. XML-Typ siehe ESLabelInfoPagedResult
Parameter:	tag Stil Erforderlich Typ	Bezeichnung des Labels PATH Ja String
	page Stil Erforderlich Typ	Die Seite der Ergebnisliste. QUERY Nein Integer
	recordsPerPage Stil Erforderlich Typ	Die maximale Anzahl von Einträgen pro Seite, welche zurückgegeben werden soll. QUERY Nein Integer

getLabelsWithUnsuccessfulLastUpdate()

Gibt Statusinformationen für alle bekannten Displays zurück, bei denen das letzte Update nicht erfolgreich war.

Typ	Request	
Methode:	GET	
URL:	/service/labelinfo/unsuccessful?page={page}&recordsPerPage={recordsPerPage}	
Antwort:	ESLabelInfo PagedResult Typ Definition und Beispiel	Ergebnisliste der ESLabelInfo-Einträge. XML-Typ siehe ESLabelInfoPagedResult
Parameter:	page Stil Erforderlich Typ	Die Seite der Ergebnisliste. QUERY Nein Integer

recordsPerPage	Die maximale Anzahl von Einträgen pro Seite, welche zurückgegeben werden soll.
Stil	QUERY
Erforderlich	Nein
Typ	Integer

getLabelsWithoutEncryption()

Gibt Statusinformationen für alle bekannten Displays zurück, die nicht verschlüsselt sind.

Typ	Request	
Methode:	GET	
URL:	/service/labelinfo/unencrypted?page={page}&recordsPerPage={recordsPerPage}	
Antwort:	ESLLabelInfo PagedResult	Ergebnisliste der ESLLabelInfo-Einträge.
	Typ	XML-Typ
	Definition und Beispiel	siehe ESLLabelInfoPagedResult
Parameter:	page	Die Seite der Ergebnisliste.
	Stil	QUERY
	Erforderlich	Nein
	Typ	Integer
	recordsPerPage	Die maximale Anzahl von Einträgen pro Seite, welche zurückgegeben werden soll.
	Stil	QUERY
	Erforderlich	Nein
	Typ	Integer

ESLLabelInfo

Informationen zu den Displays.

Typ: XML-Typ

Schema:

```
<xs:schema version="1.0">
  <xs:element name="LabelInfo" type="LabelInfo"/>
  <xs:element name="Tag" type="Tag"/>
  <xs:complexType name="LabelInfo">
    <xs:sequence>
      <xs:element name="LabelId" type="xs:string"/>
      <xs:element name="Type" type="xs:string" minOccurs="0"/>
      <xs:element ref="Tag" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="WakeupTime" type="xs:dateTime" minOccurs="0"/>
      <xs:element name="FirmwareVersion" type="xs:string" minOccurs="0"/>
      <xs:element name="SecurityStatus" type="securityStatus" minOccurs="0"/>
      <xs:element name="PowerStatus" type="powerStatus" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

```

<xs:element name="SyncQuality" type="xs:int" minOccurs="0"/>
<xs:element name="LabelErrors" type="xs:int" minOccurs="0"/>
<xs:element name="CurrentPage" type="xs:string" minOccurs="0"/>
<xs:element name="Rssi" type="xs:int" minOccurs="0"/>
<xs:element name="Lqi" type="xs:int" minOccurs="0"/>
<xs:element name="AccessPointId" type="xs:int" minOccurs="0"/>
<xs:element name="ConnectionStatus" type="connectionStatus" minOccurs="0"/>
<xs:element name="TaskId" type="xs:string" minOccurs="0"/>
<xs:element name="TaskType" type="taskType" minOccurs="0"/>
<xs:element name="Page" type="xs:int" minOccurs="0"/>
<xs:element name="Status" type="status" minOccurs="0"/>
<xs:element name="ErrorCode" type="xs:int" minOccurs="0"/>
<xs:element name="UpdatedAt" type="xs:dateTime" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
<xs:simpleType name="Tag">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="securityStatus">
  <xs:restriction base="xs:string">
    <xs:enumeration value="AES128"/>
    <xs:enumeration value="UNSUPPORTED"/>
    <xs:enumeration value="KEY_UNSET"/>
    <xs:enumeration value="NO_PIN"/>
    <xs:enumeration value="UNKNOWN"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="powerStatus">
  <xs:restriction base="xs:string">
    <xs:enumeration value="GOOD"/>
    <xs:enumeration value="LOW"/>
    <xs:enumeration value="BAD"/>
    <xs:enumeration value="UNKNOWN"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="connectionStatus">
  <xs:restriction base="xs:string">
    <xs:enumeration value="ONLINE"/>
    <xs:enumeration value="DISCOVERED"/>
    <xs:enumeration value="UNKNOWN"/>
    <xs:enumeration value="OFFLINE"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="taskType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="IMAGE"/>
    <xs:enumeration value="PRELOAD_IMAGE"/>
    <xs:enumeration value="FIRMWARE"/>
    <xs:enumeration value="SWITCH_PAGE"/>
    <xs:enumeration value="REFRESH_DISPLAY"/>
    <xs:enumeration value="QUERY_SENSOR"/>
    <xs:enumeration value="PROGRAM_KEYS"/>
    <xs:enumeration value="SET_CONFIG"/>
    <xs:enumeration value="GET_CONFIG"/>
    <xs:enumeration value="RESET_ALL_COUNTERS"/>
    <xs:enumeration value="RESET_COUNTER"/>
    <xs:enumeration value="QUERY_COUNTER"/>
    <xs:enumeration value="FORCE_CHANNEL_SCAN"/>
    <xs:enumeration value="REBOOT"/>
    <xs:enumeration value="PING"/>
    <xs:enumeration value="REGISTER_LABEL"/>
    <xs:enumeration value="UNREGISTER_LABEL"/>
    <xs:enumeration value="UNLOCK_LABEL"/>
    <xs:enumeration value="UNKNOWN"/>
  </xs:restriction>
</xs:simpleType>

```

```

    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="status">
    <xs:restriction base="xs:string">
      <xs:enumeration value="WAITING" />
      <xs:enumeration value="DELAYED" />
      <xs:enumeration value="REPLACE_REQUESTED" />
      <xs:enumeration value="CANCEL_REQUESTED" />
      <xs:enumeration value="SUCCESSFUL" />
      <xs:enumeration value="REPLACED" />
      <xs:enumeration value="CANCELED" />
      <xs:enumeration value="REMOVED" />
      <xs:enumeration value="FAILED" />
      <xs:enumeration value="TIMEOUT" />
      <xs:enumeration value="ERROR" />
    </xs:restriction>
  </xs:simpleType>
</xs:schema>

```

Beispiel:

```

<LabelInfo>
  <LabelId>B1001C4A</LabelId>
  <Type>G1 2.7</Type>
  <Tag>BEVERAGES</Tag>
  <WakeupTime>2014-07-22T08:42:36.664+02:00</WakeupTime>
  <FirmwareVersion>3.0.0</FirmwareVersion>
  <SecurityStatus>AES128</SecurityStatus>
  <PowerStatus>GOOD</PowerStatus>
  <SyncQuality>255</SyncQuality>
  <LabelErrors>0</LabelErrors>
  <CurrentPage>0</CurrentPage>
  <Rssi>-61</Rssi>
  <Lqi>37</Lqi>
  <AccessPointId>10323</AccessPointId>
  <ConnectionStatus>ONLINE</ConnectionStatus>
  <TaskId>09ba8cee-2a9a-4a38-b012-dbd58f20</TaskId>
  <TaskType>PING</TaskType>
  <Status>SUCCESSFUL</Status>
  <UpdatedAt>2014-07-22T08:42:36.664+02:00</UpdatedAt>
</LabelInfo>

```

ESLabelInfoPagedResult

Ergebnisliste der ESLabelInfo-Einträge.

Typ: XML-Typ

Schema:

```

<xs:schema version="1.0">
  <xs:element name="LabelInfo" type="LabelInfo" />
  <xs:element name="LabelInfoPagedResult" type="LabelInfoPagedResult" />
  <xs:element name="PagedResult" type="PagedResult" />
  <xs:element name="Tag" type="Tag" />
  <xs:complexType name="LabelInfoPagedResult">
    <xs:complexContent>
      <xs:extension base="PagedResult">
        <xs:sequence>
          <xs:element ref="LabelInfo" maxOccurs="unbounded" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="PagedResult" abstract="true">
    <xs:sequence />
  </xs:complexType>

```

```

<xs:attribute name="records" type="xs:int" use="required"/>
<xs:attribute name="totalRecords" type="xs:int" use="required"/>
<xs:attribute name="totalPages" type="xs:int" use="required"/>
<xs:attribute name="page" type="xs:int" use="required"/>
<xs:attribute name="recordsPerPage" type="xs:int" use="required"/>
</xs:complexType>
<xs:complexType name="LabelInfo">
  <xs:sequence>
    <xs:element name="LabelId" type="xs:string"/>
    <xs:element name="Type" type="xs:string" minOccurs="0"/>
    <xs:element ref="Tag" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="WakeupTime" type="xs:dateTime" minOccurs="0"/>
    <xs:element name="FirmwareVersion" type="xs:string" minOccurs="0"/>
    <xs:element name="SecurityStatus" type="securityStatus" minOccurs="0"/>
    <xs:element name="PowerStatus" type="powerStatus" minOccurs="0"/>
    <xs:element name="SyncQuality" type="xs:int" minOccurs="0"/>
    <xs:element name="LabelErrors" type="xs:int" minOccurs="0"/>
    <xs:element name="CurrentPage" type="xs:string" minOccurs="0"/>
    <xs:element name="Rssi" type="xs:int" minOccurs="0"/>
    <xs:element name="Lqi" type="xs:int" minOccurs="0"/>
    <xs:element name="AccessPointId" type="xs:int" minOccurs="0"/>
    <xs:element name="ConnectionStatus" type="connectionStatus" minOccurs="0"/>
    <xs:element name="TaskId" type="xs:string" minOccurs="0"/>
    <xs:element name="TaskType" type="taskType" minOccurs="0"/>
    <xs:element name="Page" type="xs:int" minOccurs="0"/>
    <xs:element name="Status" type="status" minOccurs="0"/>
    <xs:element name="ErrorCode" type="xs:int" minOccurs="0"/>
    <xs:element name="UpdatedAt" type="xs:dateTime" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:simpleType name="Tag">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="securityStatus">
  <xs:restriction base="xs:string">
    <xs:enumeration value="AES128"/>
    <xs:enumeration value="UNSUPPORTED"/>
    <xs:enumeration value="KEY_UNSET"/>
    <xs:enumeration value="NO_PIN"/>
    <xs:enumeration value="UNKNOWN"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="powerStatus">
  <xs:restriction base="xs:string">
    <xs:enumeration value="GOOD"/>
    <xs:enumeration value="LOW"/>
    <xs:enumeration value="BAD"/>
    <xs:enumeration value="UNKNOWN"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="connectionStatus">
  <xs:restriction base="xs:string">
    <xs:enumeration value="ONLINE"/>
    <xs:enumeration value="DISCOVERED"/>
    <xs:enumeration value="UNKNOWN"/>
    <xs:enumeration value="OFFLINE"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="taskType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="IMAGE"/>
    <xs:enumeration value="PRELOAD_IMAGE"/>
    <xs:enumeration value="FIRMWARE"/>
    <xs:enumeration value="SWITCH_PAGE"/>
  </xs:restriction>
</xs:simpleType>

```

```

<xs:enumeration value="REFRESH_DISPLAY" />
<xs:enumeration value="QUERY_SENSOR" />
<xs:enumeration value="PROGRAM_KEYS" />
<xs:enumeration value="SET_CONFIG" />
<xs:enumeration value="GET_CONFIG" />
<xs:enumeration value="RESET_ALL_COUNTERS" />
<xs:enumeration value="RESET_COUNTER" />
<xs:enumeration value="QUERY_COUNTER" />
<xs:enumeration value="FORCE_CHANNEL_SCAN" />
<xs:enumeration value="REBOOT" />
<xs:enumeration value="PING" />
<xs:enumeration value="REGISTER_LABEL" />
<xs:enumeration value="UNREGISTER_LABEL" />
<xs:enumeration value="UNLOCK_LABEL" />
<xs:enumeration value="UNKNOWN" />
</xs:restriction>
</xs:simpleType>
<xs:simpleType name="status">
  <xs:restriction base="xs:string">
    <xs:enumeration value="WAITING" />
    <xs:enumeration value="DELAYED" />
    <xs:enumeration value="REPLACE_REQUESTED" />
    <xs:enumeration value="CANCEL_REQUESTED" />
    <xs:enumeration value="SUCCESSFUL" />
    <xs:enumeration value="REPLACED" />
    <xs:enumeration value="CANCELED" />
    <xs:enumeration value="REMOVED" />
    <xs:enumeration value="FAILED" />
    <xs:enumeration value="TIMEOUT" />
    <xs:enumeration value="ERROR" />
  </xs:restriction>
</xs:simpleType>
</xs:schema>

```

Beispiel:

```

<LabelInfoPagedResult records="1" totalRecords="1" totalPages="1" page="0"
recordsPerPage="1">
  <LabelInfo>
    <LabelId>B1001C4A</LabelId>
    <Type>G1 2.7</Type>
    <Tag>BEVERAGES</Tag>
    <WakeupTime>2014-07-22T08:42:36.905+02:00</WakeupTime>
    <FirmwareVersion>3.0.0</FirmwareVersion>
    <SecurityStatus>AES128</SecurityStatus>
    <PowerStatus>GOOD</PowerStatus>
    <SyncQuality>255</SyncQuality>
    <LabelErrors>0</LabelErrors>
    <CurrentPage>0</CurrentPage>
    <Rssi>-61</Rssi>
    <Lqi>37</Lqi>
    <AccessPointId>10323</AccessPointId>
    <ConnectionStatus>ONLINE</ConnectionStatus>
    <TaskId>c6659176-b003-4a84-a266-38c99053348f</TaskId>
    <TaskType>PING</TaskType>
    <Status>SUCCESSFUL</Status>
    <UpdatedAt>2014-07-22T08:42:36.905+02:00</UpdatedAt>
  </LabelInfo>
</LabelInfoPagedResult>

```

ConnectionStatus

Der zu filternde Verbindungsstatus.

Typ: Aufzählungstyp

Schema:

```
<xs:schema version="1.0">
  <xs:simpleType name="connectionStatus">
    <xs:restriction base="xs:string">
      <xs:enumeration value="ONLINE" />
      <xs:enumeration value="DISCOVERED" />
      <xs:enumeration value="UNKNOWN" />
      <xs:enumeration value="OFFLINE" />
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

Aufzählungswerte:

Der Aufzählungswert kann als Pfad oder Abfrageparameter verwendet werden.

Wert	Beschreibung
ONLINE	Gerät ist verbunden
UNKNOWN	Konnte den Verbindungsstatus nicht bestimmen
OFFLINE	Gerät ist nicht verbunden

PowerStatus

Gibt den Power-Status eines Displays an. Wenn das Display batteriebetrieben ist, wird hier die verbleibende Kapazität der Batterie angezeigt.

Typ: Aufzählungstyp

Schema:

```
<xs:schema version="1.0">
  <xs:simpleType name="powerStatus">
    <xs:restriction base="xs:string">
      <xs:enumeration value="GOOD" />
      <xs:enumeration value="LOW" />
      <xs:enumeration value="BAD" />
      <xs:enumeration value="UNKNOWN" />
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

Aufzählungswerte:

Der Aufzählungswert kann als Pfad oder Abfrageparameter verwendet werden.

Wert	Beschreibung
GOOD	Guter Power-Status
LOW	Ein Batteriewechsel ist in einigen Wochen erforderlich
BAD	Ein Batteriewechsel ist sofort erforderlich
UNKNOWN	Kein aktueller Power-Sensorwert verfügbar

5 LabelResource

Diese Ressource verwaltet die Displays. Es können bereits registrierte Displays angezeigt, neue Displays registriert, die Registrierung zurückgezogen und Displays entsperrt werden.

Table 5: Übersicht

Method	URL	Name
GET	/service/label	getRegisteredLabels
GET	/service/label/register/{registrationCode}	registerLabel
POST	/service/label	registerLabels
POST	/service/label/unlock	unlockLabels
DELETE	/service/label/{labelId}	unregisterLabel
POST	/service/label/unregister	unregisterLabels

getRegisteredLabels()

Gibt alle registrierten Displays zurück.

Typ	Request	
Methode:	GET	
URL:	/service/label	
Antwort:	ESLLabelList	Liste der ESLLabel-Einträge
	Typ	XML-Typ
	Definition und Beispiel	siehe ESLLabelList

registerLabel()

Registriert ein neues Display.

Typ	Request	
Methode:	GET	
URL:	/service/label/register/{registrationCode}	
Antwort:	ESLTransaction	Ein Aktualisierungsauftrag an das Display.
	Typ	XML-Typ
	Definition und Beispiel	siehe ESLTransaction
Parameter:	registrationCode	Der Registrierungs-Code, mit dem das entsprechende Display im Wireless ePaper Server registriert wird.
	Stil	PATH

Erforderlich	Ja
Typ	String

registerLabels()

Registriert mehrere neue Displays, die in der ESLLabelList angegeben sind.

Typ	Request	
Methode:	POST	
URL:	/service/label	
Anfrage:	ESLLabelList Typ Definition und Beispiel	Liste der ESLLabel-Einträge XML-Typ siehe ESLLabelList
Antwort:	ESLTransaction Typ Definition und Beispiel	Ein Aktualisierungsauftrag an das Display. XML-Typ siehe ESLTransaction

unlockLabels()

Entsperrt Displays mit Angabe eines oder mehrerer UnlockCodes in einer ESLUnlockCodeList.

Typ	Request	
Methode:	POST	
URL:	/service/label/unlock	
Anfrage:	ESLUnlockCodeList Typ Definition und Beispiel	Liste der ESLUnlockCode-Einträge XML-Typ siehe ESLUnlockCodeList
Antwort:	ESLTransaction Typ Definition und Beispiel	Ein Aktualisierungsauftrag an das Display. XML-Typ siehe ESLTransaction

unregisterLabel()

Zieht die Registrierung eines Displays zurück, das durch die labelID angegeben ist.

Typ	Request	
Methode:	DELETE	
URL:	/service/label/{labelId}	
Antwort:	ESLTransaction	Ein Aktualisierungsauftrag an das Display.

	Typ Definition und Beispiel	XML-Typ siehe ESLTransaction
Parameter:	labelID Stil Erforderlich Typ	ID des Displays PATH Ja String

unregisterLabels()

Zieht die Registrierung mehrerer Displays zurück, die in der ESLLabelList angegeben sind.

Typ	Request	
Methode:	POST	
URL:	/service/label/unregister	
Anfrage:	ESLLabelList Typ Definition und Beispiel	Liste der ESLLabel Einträge XML-Typ siehe ESLLabelList
Antwort:	ESLTransaction Typ Definition und Beispiel	Ein Aktualisierungsauftrag an das Display. XML-Typ siehe ESLTransaction

ESLLabelList

Liste der ESLLabel Einträge.

Typ: XML-Typ

Schema:

```
<xs:schema version="1.0">
  <xs:element name="Label" type="Label"/>
  <xs:element name="LabelList" type="LabelList"/>
  <xs:element name="Tag" type="Tag"/>
  <xs:complexType name="LabelList">
    <xs:sequence>
      <xs:element ref="Label" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="Label">
    <xs:sequence>
      <xs:element ref="Tag" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:string" use="required"/>
  </xs:complexType>
  <xs:simpleType name="Tag">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>
</xs:schema>
```

Beispiel:

```
<LabelList>
  <Label id="XYZWFGK1234"/>
  <Label id="XYZWFGK1235"/>
  <Label id="XYZWFGK1236"/>
</LabelList>
```

ESLTransaction

Ein Aktualisierungsauftrag an das Display.

Typ: XML-Typ

Schema:

```
<xs:schema version="1.0">
  <xs:element name="Transaction" type="Transaction"/>
  <xs:complexType name="Transaction">
    <xs:sequence/>
    <xs:attribute name="id" type="xs:long" use="required"/>
  </xs:complexType>
</xs:schema>
```

ESLUnlockCodeList

Liste der ESLUnlockCode Einträge.

Typ: XML-Typ

Schema:

```
<xs:schema version="1.0">
  <xs:element name="UnlockCode" type="UnlockCode"/>
  <xs:element name="UnlockCodeList" type="UnlockCodeList"/>
  <xs:complexType name="UnlockCodeList">
    <xs:sequence>
      <xs:element ref="UnlockCode" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="UnlockCode">
    <xs:sequence/>
    <xs:attribute name="code" type="xs:string" use="required"/>
  </xs:complexType>
</xs:schema>
```

Beispiel:

```
<UnlockCodeList>
  <UnlockCode code="1112h2n2NbnAvsfz2m4Zn8RHnA9"/>
</UnlockCodeList>
```

6 Level1ExportResource

Diese Ressource verwaltet die Exports auf der ersten Ebene. Dazu zählen der Export von Access Points, Konfigurationen, Displays und Lizenzen.

Table 6: Übersicht

Methode	URL	Name
GET	/service/export/level1/accesspoint	<i>exportAccessPoints</i>
GET	/service/export/level1/configuration	<i>exportConfigurationKeys</i>
GET	/service/export/level1/label	<i>exportLabels</i>
GET	/service/export/level1/license	<i>exportLicenses</i>

exportAccessPoints()

Exportiert alle bekannten Access Points.

Typ	Request						
Methode:	GET						
URL:	/service/export/level1/accesspoint						
Antwort:	<table border="1"> <tr> <td>ESLAccessPoint RegistrationList</td> <td>Liste der ESLAccessPointRegistration-Einträge</td> </tr> <tr> <td>Typ</td> <td>XML-Typ</td> </tr> <tr> <td>Definition und Beispiel</td> <td>siehe <i>ESLAccessPointRegistrationList</i></td> </tr> </table>	ESLAccessPoint RegistrationList	Liste der ESLAccessPointRegistration-Einträge	Typ	XML-Typ	Definition und Beispiel	siehe <i>ESLAccessPointRegistrationList</i>
ESLAccessPoint RegistrationList	Liste der ESLAccessPointRegistration-Einträge						
Typ	XML-Typ						
Definition und Beispiel	siehe <i>ESLAccessPointRegistrationList</i>						

exportConfigurationKeys()

Exportiert alle bekannten Konfigurationen.

Typ	Request						
Methode:	GET						
URL:	/service/export/level1/configuration						
Antwort:	<table border="1"> <tr> <td>ESLConfigurationList</td> <td>Liste der ESLConfiguration-Einträge</td> </tr> <tr> <td>Typ</td> <td>XML-Typ</td> </tr> <tr> <td>Definition und Beispiel</td> <td>siehe <i>ESLConfigurationList</i></td> </tr> </table>	ESLConfigurationList	Liste der ESLConfiguration-Einträge	Typ	XML-Typ	Definition und Beispiel	siehe <i>ESLConfigurationList</i>
ESLConfigurationList	Liste der ESLConfiguration-Einträge						
Typ	XML-Typ						
Definition und Beispiel	siehe <i>ESLConfigurationList</i>						

exportLabels()

Exportiert alle bekannten Displays.

Typ	Request
Methode:	GET

URL:	/service/export/level1/label	
Antwort:	ESLLabelList	Liste der ESLLabel-Einträge
	Typ	XML-Typ
	Definition und Beispiel	siehe ESLLabelList

exportLicenses()

Exportiert alle bekannten Lizenzen.

Typ	Request	
Methode:	GET	
URL:	/service/export/level1/license	
Antwort:	ESLLicenseList	Liste der ESLLicense-Einträge
	Typ	XML-Typ
	Definition und Beispiel	siehe ESLLicenseList

ESLAccessPointRegistrationList

Liste der ESLAccessPointRegistration-Einträge.

Typ: XML-Typ

Schema:

```
<xs:schema version="1.0">
  <xs:element name="AccessPointRegistration" type="AccessPointRegistration"/>
  <xs:element name="AccessPointRegistrationList" type="AccessPointRegistrationList"/>
  <xs:complexType name="AccessPointRegistrationList">
    <xs:sequence>
      <xs:element ref="AccessPointRegistration" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="AccessPointRegistration">
    <xs:sequence>
      <xs:element name="AccessPointId" type="xs:int"/>
      <xs:element name="ServiceAddress" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

Beispiel:

```
<AccessPointRegistrationList>
  <AccessPointRegistration>
    <AccessPointId>10078</AccessPointId>
    <ServiceAddress>http://192.168.1.200:8080</ServiceAddress>
  </AccessPointRegistration>
</AccessPointRegistrationList>
```

ESLConfigurationList

Liste der ESLConfiguration-Einträge.

Typ: XML-Typ

Schema:

```

<xs:schema version="1.0">
  <xs:element name="Configuration" type="Configuration"/>
  <xs:element name="ConfigurationList" type="ConfigurationList"/>
  <xs:complexType name="ConfigurationList">
    <xs:sequence>
      <xs:element ref="Configuration" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="Configuration">
    <xs:sequence/>
    <xs:attribute name="key" type="xs:string" use="required"/>
    <xs:attribute name="type" type="basicType"/>
    <xs:attribute name="defaultValue" type="xs:string"/>
    <xs:attribute name="value" type="xs:string"/>
  </xs:complexType>
  <xs:simpleType name="basicType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="BOOLEAN"/>
      <xs:enumeration value="INTEGER"/>
      <xs:enumeration value="FLOAT"/>
      <xs:enumeration value="STRING"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>

```

Beispiel:

```

<ConfigurationList>
  <Configuration key="transmissionRetriesCount" type="INTEGER" defaultValue="5" value="3"/>
</ConfigurationList>

```

ESLLabelList

Liste der ESLLabel-Einträge.

Typ: XML-Typ

Schema:

```

<xs:schema version="1.0">
  <xs:element name="Label" type="Label"/>
  <xs:element name="LabelList" type="LabelList"/>
  <xs:element name="Tag" type="Tag"/>
  <xs:complexType name="LabelList">
    <xs:sequence>
      <xs:element ref="Label" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="Label">
    <xs:sequence>
      <xs:element ref="Tag" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:string" use="required"/>
  </xs:complexType>
  <xs:simpleType name="Tag">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>
</xs:schema>

```

Beispiel:

```

<LabelList>
  <Label id="XYZWFGK1234"/>
  <Label id="XYZWFGK1235"/>

```



```
<Label id="XYZWFGK1236"/>
</LabelList>
```

ESLLicenseList

Liste der ESLLicense-Einträge.

Typ: XML-Typ

Schema:

```
<xs:schema version="1.0">
  <xs:element name="License" type="License"/>
  <xs:element name="LicenseList" type="LicenseList"/>
  <xs:complexType name="LicenseList">
    <xs:sequence>
      <xs:element ref="License" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="License">
    <xs:sequence>
      <xs:element name="Key" type="xs:string"/>
    </xs:sequence>
    <xs:attribute name="comment" type="xs:string" use="required"/>
  </xs:complexType>
</xs:schema>
```

Beispiel:

```
<LicenseList>
  <License comment="LANCOM Wireless ePaper Server License S - 8263071700158211">
    <Key>J7K4-3M8F-T45K-BSD6-82JS-B8G1
  </Key>
  </License>
</LicenseList>
```

7 Level1ImportResource

Diese Ressource verwaltet die Imports auf der ersten Ebene. Dazu zählen der Import von Access Points, Konfigurationen, Displays und Lizenzen.

Table 7: Übersicht

Methode	URL	Name
POST	/service/import/level1/accesspoint	<i>importAccessPoints</i>
POST	/service/import/level1/configuration	<i>importConfigurationKeys</i>
POST	/service/import/level1/label	<i>importLabels</i>
POST	/service/import/level1/license	<i>importLicenses</i>

importAccessPoints()

Importiert die Access Points, die in der ESLAccessPointRegistrationList angegeben sind.

Typ	Request								
Methode:	POST								
URL:	/service/import/level1/accesspoint?clear={clear}								
Anfrage:	<table border="1"> <tbody> <tr> <td>ESLAccessPointRegistrationList</td> <td>Liste der ESLAccessPointRegistration-Einträge</td> </tr> <tr> <td>Typ</td> <td>XML-Typ</td> </tr> <tr> <td>Definition und Beispiel</td> <td>siehe ESLAccessPointRegistrationList</td> </tr> </tbody> </table>	ESLAccessPointRegistrationList	Liste der ESLAccessPointRegistration-Einträge	Typ	XML-Typ	Definition und Beispiel	siehe ESLAccessPointRegistrationList		
ESLAccessPointRegistrationList	Liste der ESLAccessPointRegistration-Einträge								
Typ	XML-Typ								
Definition und Beispiel	siehe ESLAccessPointRegistrationList								
Antwort:	<table border="1"> <tbody> <tr> <td>Typ</td> <td>ESLMessage</td> </tr> </tbody> </table>	Typ	ESLMessage						
Typ	ESLMessage								
Parameter:	<table border="1"> <tbody> <tr> <td>Parameter</td> <td>clear</td> </tr> <tr> <td>Stil</td> <td>QUERY</td> </tr> <tr> <td>Erforderlich</td> <td>Ja</td> </tr> <tr> <td>Typ</td> <td>Boolean</td> </tr> </tbody> </table>	Parameter	clear	Stil	QUERY	Erforderlich	Ja	Typ	Boolean
Parameter	clear								
Stil	QUERY								
Erforderlich	Ja								
Typ	Boolean								

importConfigurationKeys()

Importiert die Konfigurationen, die in der ESLConfigurationList angegeben sind.

Typ	Request		
Methode:	POST		
URL:	/service/import/level1/configuration?clear={clear}		
Anfrage:	<table border="1"> <tbody> <tr> <td>ESLConfigurationList</td> <td>Liste der ESLConfiguration-Einträge</td> </tr> </tbody> </table>	ESLConfigurationList	Liste der ESLConfiguration-Einträge
ESLConfigurationList	Liste der ESLConfiguration-Einträge		

	Typ	XML-Typ
	Definition und Beispiel	siehe ESLConfigurationList
Antwort:	Typ	ESLMessage
Parameter:	Parameter	clear
	Stil	QUERY
	Erforderlich	Ja
	Typ	Boolean

importLabels()

Importiert die Displays, die in der ESLLabelList angegeben sind.

Typ	Request	
Methode:	POST	
URL:	/service/import/level1/label?clear={clear}	
Anfrage:	ESLLabelList	Liste der ESLabel-Einträge
	Typ	XML-Typ
	Definition und Beispiel	siehe ESLLabelList
Antwort:	Typ	ESLMessage
Parameter:	Parameter	clear
	Stil	QUERY
	Erforderlich	Ja
	Typ	Boolean

importLicenses()

Importiert die Lizenzen, die in der ESLLicenseList angegeben sind.

Typ	Request	
Methode:	POST	
URL:	/service/import/level1/license?clear={clear}	
Anfrage:	ESLLicenseList	Liste der ESLLicense-Einträge
	Typ	XML-Typ
	Definition und Beispiel	siehe ESLLicenseList
Antwort:	Typ	ESLMessage

Parameter:	Parameter	clear
	Stil	QUERY
	Erforderlich	Ja
	Typ	Boolean

ESLAccessPointRegistrationList

Liste der ESLAccessPointRegistration-Einträge.

Typ: XML-Typ

Schema:

```
<xs:schema version="1.0">
  <xs:element name="AccessPointRegistration" type="AccessPointRegistration"/>
  <xs:element name="AccessPointRegistrationList" type="AccessPointRegistrationList"/>
  <xs:complexType name="AccessPointRegistrationList">
    <xs:sequence>
      <xs:element ref="AccessPointRegistration" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="AccessPointRegistration">
    <xs:sequence>
      <xs:element name="AccessPointId" type="xs:int"/>
      <xs:element name="ServiceAddress" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

Beispiel:

```
<AccessPointRegistrationList>
  <AccessPointRegistration>
    <AccessPointId>10078</AccessPointId>
    <ServiceAddress>http://192.168.1.200:8080</ServiceAddress>
  </AccessPointRegistration>
</AccessPointRegistrationList>
```

ESLConfigurationList

Liste der ESLConfiguration-Einträge.

Typ: XML-Typ

Schema:

```
<xs:schema version="1.0">
  <xs:element name="Configuration" type="Configuration"/>
  <xs:element name="ConfigurationList" type="ConfigurationList"/>
  <xs:complexType name="ConfigurationList">
    <xs:sequence>
      <xs:element ref="Configuration" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="Configuration">
    <xs:sequence/>
    <xs:attribute name="key" type="xs:string" use="required"/>
    <xs:attribute name="type" type="basicType"/>
    <xs:attribute name="defaultValue" type="xs:string"/>
    <xs:attribute name="value" type="xs:string"/>
  </xs:complexType>
  <xs:simpleType name="basicType">
```

```

    <xs:restriction base="xs:string">
      <xs:enumeration value="BOOLEAN"/>
      <xs:enumeration value="INTEGER"/>
      <xs:enumeration value="FLOAT"/>
      <xs:enumeration value="STRING"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>

```

Beispiel:

```

<ConfigurationList>
  <Configuration key="transmissionRetriesCount" type="INTEGER" defaultValue="5" value="3"/>
</ConfigurationList>

```

ESLLabelList

Liste der ESLLabel-Einträge.

Typ: XML-Typ

Schema:

```

<xs:schema version="1.0">
  <xs:element name="Label" type="Label"/>
  <xs:element name="LabelList" type="LabelList"/>
  <xs:element name="Tag" type="Tag"/>
  <xs:complexType name="LabelList">
    <xs:sequence>
      <xs:element ref="Label" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="Label">
    <xs:sequence>
      <xs:element ref="Tag" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:string" use="required"/>
  </xs:complexType>
  <xs:simpleType name="Tag">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>
</xs:schema>

```

Beispiel:

```

<LabelList>
  <Label id="XYZWFGK1234"/>
  <Label id="XYZWFGK1235"/>
  <Label id="XYZWFGK1236"/>
</LabelList>

```

ESLLicenseList

Liste der ESLLicense-Einträge.

Typ: XML-Typ

Schema:

```

<xs:schema version="1.0">
  <xs:element name="License" type="License"/>
  <xs:element name="LicenseList" type="LicenseList"/>
  <xs:complexType name="LicenseList">
    <xs:sequence>
      <xs:element ref="License" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

```

```
<xs:complexType name="License">
  <xs:sequence>
    <xs:element name="Key" type="xs:string"/>
  </xs:sequence> <xs:attribute name="comment" type="xs:string" use="required"/>
</xs:complexType>
</xs:schema>
```

Beispiel:

```
<LicenseList>
  <License comment="LANCOM Wireless ePaper Server License S - 8263071700158211">
    <Key>J7K4-3M8F-T45K-BSD6-82JS-B8G1
  </Key>
  </License>
</LicenseList>
```

8 LicenseResource

Diese Ressource verwaltet die Lizenzen. Es können gespeicherte Lizenzen angezeigt, neue Lizenzen gespeichert und alle Lizenzen gelöscht werden.

Table 8: Übersicht

Methode	URL	Name
GET	/service/license	<i>getLicenses</i>
DELETE	/service/license	<i>removeAllLicenses</i>
POST	/service/license	<i>storeLicense</i>

getLicenses()

Gibt alle gespeicherten Lizenzen mit Kommentaren und dem Schlüsselwert der Lizenz zurück.

Typ	Request	
Methode:	GET	
URL:	/service/license?page={page}&recordsPerPage={recordsPerPage}	
Antwort:	ESLLicense PagedResult	Ergebnisliste der ESLLicense-Einträge.
	Typ	XML-Typ
	Definition und Beispiel	siehe ESLLicensePagedResult
Parameter:	page	Die Seite der Ergebnisliste.
	Stil	QUERY
	Erforderlich	Nein
	Typ	Integer
	recordsPerPage	Die maximale Anzahl von Einträgen pro Seite, welche zurückgegeben werden soll.
	Stil	QUERY
	Erforderlich	Nein
	Typ	Integer

removeAllLicenses()

Löscht alle Lizenzen.

Typ	Request
Methode:	DELETE

URL:	/service/license	
Antwort:	Typ	void

storeLicenses()

Speichert eine neue Lizenz mit Angabe von einem Kommentar zur Lizenz und dem Schlüsselwert der Lizenz.

Typ	Request	
Methode:	POST	
URL:	/service/license	
Anfrage:	ESLLicense Typ Definition und Beispiel	Repräsentation einer Server-Lizenz mit Kommentar und Schlüsselwert. XML-Typ siehe ESLLicense
Antwort:	Typ	void

ESLLicensePagedResultList

Ergebnisliste der ESLLicense-Einträge.

Typ: XML-Typ

Schema:

```
<xs:schema version="1.0">
  <xs:element name="License" type="License"/>
  <xs:element name="LicensePagedResult" type="LicensePagedResult"/>
  <xs:element name="PagedResult" type="PagedResult"/>
  <xs:complexType name="LicensePagedResult">
    <xs:complexContent>
      <xs:extension base="PagedResult">
        <xs:sequence>
          <xs:element ref="License" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="PagedResult" abstract="true">
    <xs:sequence/>
    <xs:attribute name="records" type="xs:int" use="required"/>
    <xs:attribute name="totalRecords" type="xs:int" use="required"/>
    <xs:attribute name="totalPages" type="xs:int" use="required"/>
    <xs:attribute name="page" type="xs:int" use="required"/>
    <xs:attribute name="recordsPerPage" type="xs:int" use="required"/>
  </xs:complexType>
  <xs:complexType name="License">
    <xs:sequence>
      <xs:element name="Key" type="xs:string"/>
    </xs:sequence>
    <xs:attribute name="comment" type="xs:string" use="required"/>
  </xs:complexType>
</xs:schema>
```


Beispiel:

```
<LicensePagedResult records="1" totalRecords="1" totalPages="1" page="0" recordsPerPage="1">
  <License comment="LANCOM Wireless ePaper Server License S - 8263071700158211">
    <Key>J7K4-3M8F-T45K-BSD6-82JS-B8G1
  </Key>
  </License>
</LicensePagedResult>
```

ESLLicense

Repräsentation einer Server-Lizenz mit Kommentar und Schlüsselwert.

Typ: XML-Typ

Schema:

```
<xs:schema version="1.0">
  <xs:element name="License" type="License"/>
  <xs:complexType name="License">
    <xs:sequence>
      <xs:element name="Key" type="xs:string"/>
    </xs:sequence>
    <xs:attribute name="comment" type="xs:string" use="required"/>
  </xs:complexType>
</xs:schema>
```

Beispiel:

```
<License comment="LANCOM Wireless ePaper Server License S - 8263071700158211">
  <Key>J7K4-3M8F-T45K-BSD6-82JS-B8G1
</Key>
</License>
```

9 ProblemResource

Diese Ressource liefert Informationen zu allen Problemen.

Table 9: Übersicht

Methode	URL	Name
GET	/service/problem	<i>getProblems</i>

getProblems()

Gibt Informationen zu allen bekannten Problemen zurück.

Typ	Request							
Methode:	GET							
URL:	/service/problem							
Antwort:	<table border="1"> <tbody> <tr> <td>ESLProblemList</td> <td>Liste der ESLProblem-Einträge</td> </tr> <tr> <td>Typ</td> <td>XML-Typ</td> </tr> <tr> <td>Definition und Beispiel</td> <td>siehe ESLProblemList</td> </tr> </tbody> </table>		ESLProblemList	Liste der ESLProblem-Einträge	Typ	XML-Typ	Definition und Beispiel	siehe ESLProblemList
ESLProblemList	Liste der ESLProblem-Einträge							
Typ	XML-Typ							
Definition und Beispiel	siehe ESLProblemList							

ESLProblemList

Liste der ESLProblem-Einträge

Typ: XML-Typ

Schema:

```
<xs:schema version="1.0">
  <xs:element name="Problem" type="Problem"/>
  <xs:element name="ProblemList" type="ProblemList"/>
  <xs:complexType name="ProblemList">
    <xs:sequence>
      <xs:element ref="Problem" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="Problem">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="identifier" type="xs:string" use="required"/>
        <xs:attribute name="severity" type="problemSeverity" use="required"/>
        <xs:attribute name="queryUrl" type="xs:string"/>
        <xs:attribute name="resultType" type="xs:string"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <xs:simpleType name="problemSeverity">
    <xs:restriction base="xs:string">
      <xs:enumeration value="INFO"/>
      <xs:enumeration value="WARNING"/>
      <xs:enumeration value="ERROR"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

```
</xs:simpleType>  
</xs:schema>
```

Beispiel:

```
<ProblemList>  
  <Problem identifier="access-point-offline" severity="ERROR"  
    queryUrl="accesspointinfo/status/offline" resultType="AccessPoint">1 offline access  
    point(s)</Problem>  
</ProblemList>
```

10 ServiceStatusResource

Diese Ressource liefert Informationen zum Service-Status.

Table 10: Übersicht

Methode	URL	Name
GET	/service/status	<i>getServiceStatus</i>

getServiceStatus()

Gibt Informationen zum Service-Status zurück.

Typ	Request							
Methode:	GET							
URL:	/service/status							
Antwort:	<table border="1"> <tbody> <tr> <td>ESLServiceStatus</td> <td>Der Status des Dienstes.</td> </tr> <tr> <td>Typ</td> <td>XML-Typ</td> </tr> <tr> <td>Definition und Beispiel</td> <td>siehe ESLServiceStatus</td> </tr> </tbody> </table>		ESLServiceStatus	Der Status des Dienstes.	Typ	XML-Typ	Definition und Beispiel	siehe ESLServiceStatus
ESLServiceStatus	Der Status des Dienstes.							
Typ	XML-Typ							
Definition und Beispiel	siehe ESLServiceStatus							

ESLServiceStatus

Der Service-Status.

Typ: XML-Typ

Schema:

```
<xs:schema version="1.0">
  <xs:element name="Property" type="Property"/>
  <xs:element name="ServiceStatus" type="ServiceStatus"/>
  <xs:complexType name="ServiceStatus">
    <xs:sequence>
      <xs:element ref="Property" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="serviceName" type="xs:string" use="required"/>
  </xs:complexType>
  <xs:complexType name="Property">
    <xs:sequence/>
    <xs:attribute name="key" type="xs:string" use="required"/>
    <xs:attribute name="value" type="xs:string" use="required"/>
  </xs:complexType>
</xs:schema>
```

11 TagResource

Diese Ressource verwaltet die Tags. Es können Tags zu Displays hinzugefügt, bzw. entfernt und eine Übersicht über bereits genutzte Bezeichnungen abgerufen werden.

Table 11: Übersicht

Methode	URL	Name
POST	/service/tag/add	addTagsToLabels
GET	/service/tag	getTags
POST	/service/tag/remove	removeTagsFromLabels

addTagsToLabels()

Fügt den Displays, die in der ESLLabelList angegeben sind, eine Bezeichnung hinzu.

Typ	Request	
Methode:	POST	
URL:	/service/tag/add	
Anfrage:	ESLLabelList	Liste der ESLLabel-Einträge
	Typ	XML-Typ
	Definition und Beispiel	siehe ESLLabelList
Antwort:	Typ	void

getTags()

Gibt Informationen zu allen bekannten Tags zurück.

Typ	Request	
Methode:	GET	
URL:	/service/tag?page={page}&recordsPerPage={recordsPerPage}	
Antwort:	ESLTagPagedResult	Ergebnisliste der ESLTag-Einträge.
	Typ	XML-Typ
	Definition und Beispiel	siehe ESLTagPagedResult
Parameter:	page	Die Seite der Ergebnisliste.
	Stil	QUERY
	Erforderlich	Nein
	Typ	Integer

recordsPerPage	Die maximale Anzahl von Einträgen pro Seite, welche zurückgegeben werden soll.
Stil	QUERY
Erforderlich	Nein
Typ	Integer

removeTagsFromLabels()

Entfernt eine Bezeichnung von den Displays, die in der ESLLabelList angegeben sind.

Typ	Request	
Methode:	POST	
URL:	/service/tag/remove	
Antwort:	ESLLabelList	Liste der ESLLabel-Einträge
	Typ	XML-Typ
	Definition und Beispiel	siehe ESLLabelList
Antwort:	Typ	void

ESLLabelList

Liste der ESLLabel-Einträge.

Typ: XML-Typ

Schema:

```
<xs:schema version="1.0">
  <xs:element name="Label" type="Label" />
  <xs:element name="LabelList" type="LabelList" />
  <xs:element name="Tag" type="Tag" />
  <xs:complexType name="LabelList">
    <xs:sequence>
      <xs:element ref="Label" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="Label">
    <xs:sequence>
      <xs:element ref="Tag" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
    <xs:attribute name="id" type="xs:string" use="required" />
  </xs:complexType>
  <xs:simpleType name="Tag">
    <xs:restriction base="xs:string" />
  </xs:simpleType>
</xs:schema>
```

Beispiel:

```
<LabelList>
  <Label id="XYZWFGK1234" />
  <Label id="XYZWFGK1235" />
  <Label id="XYZWFGK1236" />
</LabelList>
```

ESLTagPagedResult

Ergebnisliste der ESLTag-Einträge.

Typ: XML-Typ

Schema:

```
<xs:schema version="1.0">
  <xs:element name="PagedResult" type="PagedResult"/>
  <xs:element name="Tag" type="Tag"/>
  <xs:element name="TagPagedResult" type="TagPagedResult"/>
  <xs:complexType name="TagPagedResult">
    <xs:complexContent>
      <xs:extension base="PagedResult">
        <xs:sequence>
          <xs:element ref="Tag" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="PagedResult" abstract="true">
    <xs:sequence/>
    <xs:attribute name="records" type="xs:int" use="required"/>
    <xs:attribute name="totalRecords" type="xs:int" use="required"/>
    <xs:attribute name="totalPages" type="xs:int" use="required"/>
    <xs:attribute name="page" type="xs:int" use="required"/>
    <xs:attribute name="recordsPerPage" type="xs:int" use="required"/>
  </xs:complexType>
  <xs:simpleType name="Tag">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>
</xs:schema>
```

Beispiel:

```
<TagPagedResult records="1" totalRecords="1" totalPages="1" page="0" recordsPerPage="1">
  <Tag>BEVERAGES</Tag>
</TagPagedResult>
```

12 TaskResource

Diese Ressource erteilt Aufträge zum Rendering von Display-Inhalten.

Table 12: Übersicht

Methode	URL	Name
POST	/service/task/preview/document	<i>getRenderingDocument</i>
POST	/service/task/preview/image	<i>getRenderingImage</i>
POST	/service/task/preview/source	<i>getRenderingSourceRecord</i>
POST	/service/task	<i>scheduleTasks</i>

getRenderingDocument()

Typ	Request							
Methode:	POST							
URL:	/service/task/preview/document							
Anfrage:	<table border="1"> <tr> <td>ESLTemplateTask</td> <td>Dieser Auftrag rendert eine Grafik unter Verwendung der angegebenen Formatvorlage und des Inhaltsdokuments, welches dem Auftrag zugeordnet ist. Die gerenderte Grafik wird dann an das Display übertragen.</td> </tr> <tr> <td>Typ</td> <td>XML-Typ</td> </tr> <tr> <td>Definition und Beispiel</td> <td>siehe ESLTemplateTask</td> </tr> </table>	ESLTemplateTask	Dieser Auftrag rendert eine Grafik unter Verwendung der angegebenen Formatvorlage und des Inhaltsdokuments, welches dem Auftrag zugeordnet ist. Die gerenderte Grafik wird dann an das Display übertragen.	Typ	XML-Typ	Definition und Beispiel	siehe ESLTemplateTask	
ESLTemplateTask	Dieser Auftrag rendert eine Grafik unter Verwendung der angegebenen Formatvorlage und des Inhaltsdokuments, welches dem Auftrag zugeordnet ist. Die gerenderte Grafik wird dann an das Display übertragen.							
Typ	XML-Typ							
Definition und Beispiel	siehe ESLTemplateTask							
Antwort:	<table border="1"> <tr> <td>Content-type: application/xml</td> <td></td> </tr> <tr> <td>Typ</td> <td>XML-Typ</td> </tr> </table>	Content-type: application/xml		Typ	XML-Typ			
Content-type: application/xml								
Typ	XML-Typ							

getRenderingImage()

Typ	Request							
Methode:	POST							
URL:	/service/task/preview/image							
Anfrage:	<table border="1"> <tr> <td>ESLTemplateTask</td> <td>Dieser Auftrag rendert eine Grafik unter Verwendung der angegebenen Formatvorlage und des Inhaltsdokuments, welches dem Auftrag zugeordnet ist. Die gerenderte Grafik wird dann an das Display übertragen.</td> </tr> <tr> <td>Typ</td> <td>XML-Typ</td> </tr> <tr> <td>Definition und Beispiel</td> <td>siehe ESLTemplateTask</td> </tr> </table>	ESLTemplateTask	Dieser Auftrag rendert eine Grafik unter Verwendung der angegebenen Formatvorlage und des Inhaltsdokuments, welches dem Auftrag zugeordnet ist. Die gerenderte Grafik wird dann an das Display übertragen.	Typ	XML-Typ	Definition und Beispiel	siehe ESLTemplateTask	
ESLTemplateTask	Dieser Auftrag rendert eine Grafik unter Verwendung der angegebenen Formatvorlage und des Inhaltsdokuments, welches dem Auftrag zugeordnet ist. Die gerenderte Grafik wird dann an das Display übertragen.							
Typ	XML-Typ							
Definition und Beispiel	siehe ESLTemplateTask							
Antwort:	<table border="1"> <tr> <td>Content-type: image/png</td> <td></td> </tr> </table>	Content-type: image/png						
Content-type: image/png								

Typ	PNG-Typ
-----	---------

getRenderingSourceRecord()

Typ	Request							
Methode:	POST							
URL:	/service/task/preview/source							
Anfrage:	<table border="1"> <tr> <td>ESLTemplateTask</td> <td>Dieser Auftrag rendert eine Grafik unter Verwendung der angegebenen Formatvorlage und des Inhaltsdokuments, welches dem Auftrag zugeordnet ist. Die gerenderte Grafik wird dann an das Display übertragen.</td> </tr> <tr> <td>Typ</td> <td>XML-Typ</td> </tr> <tr> <td>Definition und Beispiel</td> <td>siehe ESLTemplateTask</td> </tr> </table>	ESLTemplateTask	Dieser Auftrag rendert eine Grafik unter Verwendung der angegebenen Formatvorlage und des Inhaltsdokuments, welches dem Auftrag zugeordnet ist. Die gerenderte Grafik wird dann an das Display übertragen.	Typ	XML-Typ	Definition und Beispiel	siehe ESLTemplateTask	
ESLTemplateTask	Dieser Auftrag rendert eine Grafik unter Verwendung der angegebenen Formatvorlage und des Inhaltsdokuments, welches dem Auftrag zugeordnet ist. Die gerenderte Grafik wird dann an das Display übertragen.							
Typ	XML-Typ							
Definition und Beispiel	siehe ESLTemplateTask							
Antwort:	<table border="1"> <tr> <td>Content-type: application/xml</td> <td></td> </tr> <tr> <td>Typ</td> <td>XML-Typ</td> </tr> </table>	Content-type: application/xml		Typ	XML-Typ			
Content-type: application/xml								
Typ	XML-Typ							

scheduleTasks()

Reiht einen oder mehrere Aufträge zur Ausführung ein. Ein Auftrag weist den Server an, eine Aktualisierung an ein bestimmtes Display zu senden. Abhängig vom Auftragsstyp wird dabei das angezeigte Bild geändert oder eine andere Aktion auf dem Display ausgeführt.

Typ	Request							
Methode:	POST							
URL:	/service/task							
Anfrage:	<table border="1"> <tr> <td>ESLTaskOrder</td> <td>Eine Auftragsabfolge enthält eine Liste von Aufträgen, die zusammengefasst in einer Transaktion verarbeitet werden.</td> </tr> <tr> <td>Typ</td> <td>XML-Typ</td> </tr> <tr> <td>Definition und Beispiel</td> <td>siehe ESLTaskOrder</td> </tr> </table>	ESLTaskOrder	Eine Auftragsabfolge enthält eine Liste von Aufträgen, die zusammengefasst in einer Transaktion verarbeitet werden.	Typ	XML-Typ	Definition und Beispiel	siehe ESLTaskOrder	
ESLTaskOrder	Eine Auftragsabfolge enthält eine Liste von Aufträgen, die zusammengefasst in einer Transaktion verarbeitet werden.							
Typ	XML-Typ							
Definition und Beispiel	siehe ESLTaskOrder							
Antwort:	<table border="1"> <tr> <td>ESLTransaction</td> <td>Ein Aktualisierungsauftrag an das Display.</td> </tr> <tr> <td>Typ</td> <td>XML-Typ</td> </tr> <tr> <td>Definition und Beispiel</td> <td>siehe ESLTransaction</td> </tr> </table>	ESLTransaction	Ein Aktualisierungsauftrag an das Display.	Typ	XML-Typ	Definition und Beispiel	siehe ESLTransaction	
ESLTransaction	Ein Aktualisierungsauftrag an das Display.							
Typ	XML-Typ							
Definition und Beispiel	siehe ESLTransaction							

ESLTemplateTask

Dieser Auftrag rendert eine Grafik unter Verwendung der angegebenen Formatvorlage und des Inhaltsdokuments, welches dem Auftrag zugeordnet ist. Die gerenderte Grafik wird dann an das Display übertragen.

Typ: XML-Typ

Schema:

```

<xs:schema version="1.0">
  <xs:element name="Task" type="Task"/>
  <xs:element name="TemplateTask" type="TemplateTask"/>
  <xs:complexType name="TemplateTask">
    <xs:complexContent>
      <xs:extension base="Task">
        <xs:sequence>
          <xs:any processContents="skip" namespace="##other"/>
        </xs:sequence>
        <xs:attribute name="page" type="xs:int"/>
        <xs:attribute name="preload" type="xs:boolean"/>
        <xs:attribute name="template" type="xs:string" use="required"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="Task">
    <xs:sequence/>
    <xs:attribute name="labelId" type="xs:string" use="required"/>
    <xs:attribute name="taskPriority" type="taskPriority"/>
    <xs:attribute name="externalId" type="xs:long"/>
    <xs:attribute name="replaceWaitingTasks" type="xs:boolean"/>
  </xs:complexType>
  <xs:simpleType name="taskPriority">
    <xs:restriction base="xs:string">
      <xs:enumeration value="LOW"/>
      <xs:enumeration value="NORMAL"/>
      <xs:enumeration value="HIGH"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>

```

Beispiel:

```

<TemplateTask page="0" preload="false" template="template.xml" labelId="B1001C4A"
taskPriority="NORMAL" externalId="4711">
  <Article>
    <Name>Coca Cola</Name>
    <Price>0.65</Price>
  </Article>
</TemplateTask>

```

ESLTaskOrder

Eine Auftragsabfolge enthält eine Liste von Aufträgen, die zusammengefasst in einer Transaktion verarbeitet werden.

Typ: XML-Typ

Schema:

```

<xs:schema version="1.0">
  <xs:element name="ImageTask" type="ImageTask"/>
  <xs:element name="RefreshTask" type="RefreshTask"/>
  <xs:element name="ResetTask" type="ResetTask"/>
  <xs:element name="SwitchPageTask" type="SwitchPageTask"/>
  <xs:element name="Task" type="Task"/>
  <xs:element name="TaskOrder" type="TaskOrder"/>
  <xs:element name="TemplateTask" type="TemplateTask"/>
  <xs:complexType name="TaskOrder">
    <xs:sequence>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element ref="SwitchPageTask"/>
        <xs:element ref="ResetTask"/>
        <xs:element ref="RefreshTask"/>
        <xs:element ref="ImageTask"/>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>

```

```

    <xs:element ref="TemplateTask"/>
  </xs:choice>
</xs:sequence>
<xs:attribute name="title" type="xs:string"/>
<xs:attribute name="externalId" type="xs:long"/>
</xs:complexType>
<xs:complexType name="SwitchPageTask">
  <xs:complexContent>
    <xs:extension base="Task">
      <xs:sequence/>
      <xs:attribute name="page" type="xs:int" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="Task">
  <xs:sequence/>
  <xs:attribute name="labelId" type="xs:string" use="required"/>
  <xs:attribute name="taskPriority" type="taskPriority"/>
  <xs:attribute name="externalId" type="xs:long"/>
  <xs:attribute name="replaceWaitingTasks" type="xs:boolean"/>
</xs:complexType>
<xs:complexType name="ResetTask">
  <xs:complexContent>
    <xs:extension base="Task">
      <xs:sequence/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="RefreshTask">
  <xs:complexContent>
    <xs:extension base="Task">
      <xs:sequence/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="ImageTask">
  <xs:complexContent>
    <xs:extension base="Task">
      <xs:sequence>
        <xs:element name="Image" type="xs:base64Binary"/>
      </xs:sequence>
      <xs:attribute name="page" type="xs:int"/>
      <xs:attribute name="preload" type="xs:boolean"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="TemplateTask">
  <xs:complexContent>
    <xs:extension base="Task">
      <xs:sequence>
        <xs:any processContents="skip" namespace="##other"/>
      </xs:sequence>
      <xs:attribute name="page" type="xs:int"/>
      <xs:attribute name="preload" type="xs:boolean"/>
      <xs:attribute name="template" type="xs:string" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:simpleType name="taskPriority">
  <xs:restriction base="xs:string">
    <xs:enumeration value="LOW"/>
    <xs:enumeration value="NORMAL"/>
    <xs:enumeration value="HIGH"/>
  </xs:restriction>

```

```
</xs:simpleType>
</xs:schema>
```

Beispiel:

```
<TaskOrder title="Perform some example tasks">
  <ImageTask page="3" preload="true" labelId="B1001C4A" taskPriority="NORMAL"
externalId="4711">
    <Image>iVBORw0KGgoAAAANSUhfEUGAAAQgAAACwCAIAAAB8emcEAAACVElEQVR42u3ZoQ0
AQAgEweu/ab4BcIhPGPQoklWXDFfD8fwFH4/geWHwvDB4Xhg8LwyeFwbPC4Pnvw
jDg3i+8R7B88LgeWHwvDB4Xhg8LwyeFwbP2zd43vLN88LgeWHwvDB4XhgexwvD4
3heGDxvr+B5yzfPC4PnhcHwzUB5YfC8MHheGDxv3+B5yzfPC8PjeGF4HC8Mj+B5
YfC8MHheGDxvr+B5yzfPC4PnhcHwzUB5YfC8MHheGDxv3/Ag3vLtcTwvDJ4XBs8
Lg+eFwfPC4Hlh8Ly9guct3zwvDJ4XBs8Lg+eF4XG8MDyOF4ZH8Lx9g+ct3zwvDJ
4XBs8Lg+eFwfPC4Hlh8Ly9guct3zwvDJ4XhsfxwvA4nhcGzwd54XB8/YNnrD88
7wweF4YPC8MnhcGzwd54XB8/YKnrd8exwvDI/jeWHwvDB4Xhg8LwyeFwbP2zd4
3vLN88LgeWHwvDB4Xhg8Lwyp44XhcbY9woN43vLN88LgeWHwvDB4Xhg8LwyeFwb
P2zd43vLN88LgeWHwvDA8jheGx/G8MHheGDxvr+B5yzfPC4PnhcHwzUB5YfC8MH
heGDxv3+B5y7fH8cLwOF4YHsHwzUB5YfC8MHheGDxvr+B5yzfPC4PnhcHwzUB5Y
fC8MHheGB7H8x7E85ZvnhcGzwd54XB88LgeWHwvDB4317B85ZvnhcGzwd54Xh
cbwwPI4XhkfwwDB43r7B85ZvnhcGzwd54XB88LgeWHwvDB4317B85ZvnheGx/H
C8Die9wieFwbPC4PnhcHzm/4BMro/olaImjgAAAAASUVORK5CYII=
    </Image>
  </ImageTask>
  <RefreshTask labelId="B1001C4A" taskPriority="LOW" externalId="4711"/>
  <ResetTask labelId="B1001C4A" taskPriority="NORMAL" externalId="4711"/>
  <SwitchPageTask page="2" labelId="B1001C4A" taskPriority="NORMAL" externalId="4711"/>

  <TemplateTask page="0" preload="false" template="template.xsl" labelId="B1001C4A"
taskPriority="NORMAL" externalId="4711">
    <Article>
      <Name>Coca Cola</Name>
      <Price>0.65</Price>
    </Article>
  </TemplateTask>
</TaskOrder>
```

ESLTransaction

Ein Aktualisierungsauftrag an das Display.

Typ: XML-Typ

Schema:

```
<xs:schema version="1.0">
  <xs:element name="Transaction" type="Transaction"/>
  <xs:complexType name="Transaction">
    <xs:sequence/>
    <xs:attribute name="id" type="xs:long" use="required"/>
  </xs:complexType>
</xs:schema>
```

13 TemplateResource

Diese Ressource liefert Informationen zu allen Templates.

Table 13: Übersicht

Methode	URL	Name
GET	/service/template/invalid	getInvalidTemplates
GET	/service/template	getTemplates

getInvalidTemplates()

Gibt Informationen zu ungültigen Templates zurück.

Typ	Request	
Methode:	GET	
URL:	/service/template/invalid?page={page}&recordsPerPage={recordsPerPage}	
Antwort:	ESLTemplatePagedResult	Ergebnisliste der ESLTemplate-Einträge.
	Typ	XML-Typ
	Definition und Beispiel	siehe ESLTemplatePagedResult
Parameter:	page	Die Seite der Ergebnisliste.
	Stil	QUERY
	Erforderlich	Nein
	Typ	Integer
	recordsPerPage	Die maximale Anzahl von Einträgen pro Seite, welche zurückgegeben werden soll.
	Stil	QUERY
	Erforderlich	Nein
	Typ	Integer

getTemplates()

Gibt Informationen zu allen Templates zurück.

Typ	Request	
Methode:	GET	
URL:	/service/template?page={page}&recordsPerPage={recordsPerPage}	
Antwort:	ESLTemplatePagedResult	Ergebnisliste der ESLTemplate-Einträge.
	Typ	XML-Typ

	Definition und Beispiel	siehe ESLTemplatePagedResult
Parameter:	page	Die Seite der Ergebnisliste.
	Stil	QUERY
	Erforderlich	Nein
	Typ	Integer
	recordsPerPage	Die maximale Anzahl von Einträgen pro Seite, welche zurückgegeben werden soll.
	Stil	QUERY
	Erforderlich	Nein
	Typ	Integer

ESLTemplatePagedResult

Ergebnisliste der ESLTemplate-Einträge.

Typ: XML-Typ

Schema:

```
<xs:schema version="1.0">
  <xs:element name="PagedResult" type="PagedResult"/>
  <xs:element name="Template" type="Template"/>
  <xs:element name="TemplatePagedResult" type="TemplatePagedResult"/>
  <xs:complexType name="TemplatePagedResult">
    <xs:complexContent>
      <xs:extension base="PagedResult">
        <xs:sequence>
          <xs:element ref="Template" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="PagedResult" abstract="true">
    <xs:sequence/>
    <xs:attribute name="records" type="xs:int" use="required"/>
    <xs:attribute name="totalRecords" type="xs:int" use="required"/>
    <xs:attribute name="totalPages" type="xs:int" use="required"/>
    <xs:attribute name="page" type="xs:int" use="required"/>
    <xs:attribute name="recordsPerPage" type="xs:int" use="required"/>
  </xs:complexType>
  <xs:complexType name="Template">
    <xs:sequence/>
    <xs:attribute name="name" type="xs:string"/>
  </xs:complexType>
</xs:schema>
```

Beispiel:

```
<TemplatePagedResult records="1" totalRecords="1" totalPages="1" page="0" recordsPerPage="1">
  <Template name="default.xsl"/>
</TemplatePagedResult>
```

14 TransactionResource

Diese Ressource verwaltet die Transaktionen. Es können Transaktionen abgebrochen oder deren Status angezeigt werden.

Table 14: Übersicht

Methode	URL	Name
DELETE	/service/transaction/{transactionId}	abortTransaction
GET	/service/transaction/{transactionId}/status	getTransactionStatus

abortTransaction()

Bricht die Transaktion ab, die durch die transactionId vorgegeben ist.

Typ	Request	
Methode:	DELETE	
URL:	/service/transaction/{transactionId}	
Antwort:	Typ	void
Parameter:	transactionId	ID der Transaktion
	Stil	PATH
	Erforderlich	Ja
	Typ	long

getTransactionStatus()

Gibt die Statusinformationen zu allen bekannten Transaktionen zurück.

Typ	Request	
Methode:	GET	
URL:	/service/transaction/{transactionId}/status	
Antwort:	ESLTransactionStatusInfo	Status Informationen zu einer Transaktion
	Typ	XML-Typ
	Definition und Beispiel	siehe ESLTransactionStatusInfo
Parameter:	transactionId	ID der Transaktion
	Stil	PATH
	Erforderlich	Ja
	Typ	long

ESLTransactionStatusInfo

Statusinformationen zu einer Transaktion.

Typ: XML-Typ

Schema:

```
<xs:schema version="1.0">
  <xs:element name="TransactionStatusInfo" type="TransactionStatusInfo"/>
  <xs:complexType name="TransactionStatusInfo">
    <xs:sequence>
      <xs:element name="Title" type="xs:string" minOccurs="0"/>
      <xs:element name="ExternalId" type="xs:long" minOccurs="0"/>
      <xs:element name="TotalUpdates" type="xs:int"/>
      <xs:element name="FinishedUpdates" type="xs:int"/>
      <xs:element name="ErrorUpdates" type="xs:int"/>
      <xs:element name="StartedTime" type="xs:dateTime"/>
      <xs:element name="LastUpdateTime" type="xs:dateTime"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:long" use="required"/>
    <xs:attribute name="finished" type="xs:boolean" use="required"/>
    <xs:attribute name="failed" type="xs:boolean" use="required"/>
  </xs:complexType>
</xs:schema>
```


15 UpdateStatusResource

Diese Ressource liefert Informationen über den Update-Status.

Table 15: Übersicht

Methode	URL	Name
DELETE	/service/updatestatus/{taskId}	<i>abortTask</i>
GET	/service/updatestatus	<i>getAllUpdates</i>
GET	/service/updatestatus/{taskId}/image	<i>getImageForTask</i>
GET	/service/updatestatus/unsuccessful	<i>getUnsuccessfulUpdates</i>
GET	/service/updatestatus/{taskId}	<i>getUpdateDetail</i>
GET	/service/updatestatus/{taskId}/history	<i>getUpdateHistoryForTask</i>
GET	/service/updatestatus/externalid/{externalId}	<i>getUpdatesForExternalId</i>
GET	/service/updatestatus/label/{label}	<i>getUpdatesForLabel</i>
GET	/service/updatestatus/transaction/{transactionId}	<i>getUpdatesForTransaction</i>
GET	/service/updatestatus/waiting	<i>getWaitingUpdates</i>

abortTask()

Bricht die Aufgabe ab, die durch die taskId festgelegt wurde.

Typ	Request	
Methode:	DELETE	
URL:	/service/updatestatus/{taskId}	
Antwort:	Typ	void
Parameter:	taskId	ID der Aufgabe
	Stil	PATH
	Erforderlich	Ja
	Typ	UUID

getAllUpdates()

Gibt Informationen zu allen bekannten Updates zurück.

Typ	Request
Methode:	GET
URL:	/service/updatestatus?page={page}&recordsPerPage={recordsPerPage}

Antwort:	<table border="1"> <tr> <td>ESLUpdateStatus PagedResult</td> <td>Ergebnisliste der ESLUpdateStatus-Einträge.</td> </tr> <tr> <td>Typ</td> <td>XML-Typ</td> </tr> <tr> <td>Definition und Beispiel</td> <td>siehe ESLUpdateStatusPagedResult</td> </tr> </table>	ESLUpdateStatus PagedResult	Ergebnisliste der ESLUpdateStatus-Einträge.	Typ	XML-Typ	Definition und Beispiel	siehe ESLUpdateStatusPagedResult										
ESLUpdateStatus PagedResult	Ergebnisliste der ESLUpdateStatus-Einträge.																
Typ	XML-Typ																
Definition und Beispiel	siehe ESLUpdateStatusPagedResult																
Parameter:	<table border="1"> <tr> <td>page</td> <td>Die Seite der Ergebnisliste.</td> </tr> <tr> <td>Stil</td> <td>QUERY</td> </tr> <tr> <td>Erforderlich</td> <td>Nein</td> </tr> <tr> <td>Typ</td> <td>Integer</td> </tr> </table> <table border="1"> <tr> <td>recordsPerPage</td> <td>Die maximale Anzahl von Einträgen pro Seite, welche zurückgegeben werden soll.</td> </tr> <tr> <td>Stil</td> <td>QUERY</td> </tr> <tr> <td>Erforderlich</td> <td>Nein</td> </tr> <tr> <td>Typ</td> <td>Integer</td> </tr> </table>	page	Die Seite der Ergebnisliste.	Stil	QUERY	Erforderlich	Nein	Typ	Integer	recordsPerPage	Die maximale Anzahl von Einträgen pro Seite, welche zurückgegeben werden soll.	Stil	QUERY	Erforderlich	Nein	Typ	Integer
page	Die Seite der Ergebnisliste.																
Stil	QUERY																
Erforderlich	Nein																
Typ	Integer																
recordsPerPage	Die maximale Anzahl von Einträgen pro Seite, welche zurückgegeben werden soll.																
Stil	QUERY																
Erforderlich	Nein																
Typ	Integer																

getImageForTask()

Gibt das mit der Aufgabe verknüpfte Bild zurück.

Typ	Request								
Methode:	GET								
URL:	/service/updatestatus/{taskId}/image								
Antwort:	<table border="1"> <tr> <td>Content-type: image/png</td> <td></td> </tr> <tr> <td>Typ</td> <td>PNG</td> </tr> </table>	Content-type: image/png		Typ	PNG				
Content-type: image/png									
Typ	PNG								
Parameter:	<table border="1"> <tr> <td>taskId</td> <td>ID der Aufgabe</td> </tr> <tr> <td>Stil</td> <td>PATH</td> </tr> <tr> <td>Erforderlich</td> <td>Ja</td> </tr> <tr> <td>Typ</td> <td>UUID</td> </tr> </table>	taskId	ID der Aufgabe	Stil	PATH	Erforderlich	Ja	Typ	UUID
taskId	ID der Aufgabe								
Stil	PATH								
Erforderlich	Ja								
Typ	UUID								

getUnsuccessfulUpdates()

Gibt alle gescheiterten Updates zurück.

Typ	Request		
Methode:	GET		
URL:	/service/updatestatus/unsuccessful?page={page}&recordsPerPage={recordsPerPage}		
Antwort:	<table border="1"> <tr> <td>ESLUpdateStatus PagedResult</td> <td>Ergebnisliste der ESLUpdateStatus-Einträge.</td> </tr> </table>	ESLUpdateStatus PagedResult	Ergebnisliste der ESLUpdateStatus-Einträge.
ESLUpdateStatus PagedResult	Ergebnisliste der ESLUpdateStatus-Einträge.		

	Typ	XML-Typ
	Definition und Beispiel	siehe ESLUpdateStatusPagedResult
Parameter:	page	Die Seite der Ergebnisliste.
	Stil	QUERY
	Erforderlich	Nein
	Typ	Integer
	recordsPerPage	Die maximale Anzahl von Einträgen pro Seite, welche zurückgegeben werden soll.
	Stil	QUERY
	Erforderlich	Nein
	Typ	Integer

getUpdateDetail()

Gibt Details zu einem Update an, das durch die taskId angegeben ist.

Typ	Request	
Methode:	GET	
URL:	/service/updatestatus/{taskId}	
Antwort:	ESLUpdateStatus	Der aktuelle Status eines Auftrags
	Typ	XML-Typ
	Definition und Beispiel	siehe ESLUpdateStatus
Parameter:	taskId	ID der Aufgabe
	Stil	PATH
	Erforderlich	Ja
	Typ	UUID

getUpdateHistoryForTask()

Gibt eine Übersicht über bereits durchgeführte Updates für einen Task an, der durch die taskId vorgegeben wird.

Typ	Request	
Methode:	GET	
URL:	/service/updatestatus/{taskId}/history?page={page}&recordsPerPage={recordsPerPage}	
Antwort:	ESLUpdateStatus PagedResult	Ergebnisliste der ESLUpdateStatus-Einträge.
	Typ	XML-Typ

15 UpdateStatusResource

	Definition und Beispiel	siehe ESLUpdateStatusPagedResult
Parameter:	taskId	ID der Aufgabe
	Stil	PATH
	Erforderlich	Ja
	Typ	UUID
	page	Die Seite der Ergebnisliste.
	Stil	QUERY
	Erforderlich	Nein
	Typ	Integer
	recordsPerPage	Die maximale Anzahl von Einträgen pro Seite, welche zurückgegeben werden soll.
	Stil	QUERY
	Erforderlich	Nein
	Typ	Integer

getUpdatesForExternalId()

Gibt Informationen zum Update-Status für eine externe ID an.

Typ	Request	
Methode:	GET	
URL:	/service/updatestatus/externalid/{externalId}?page={page}&recordsPerPage={recordsPerPage}	
Antwort:	ESLUpdateStatusPagedResult	Ergebnisliste der ESLUpdateStatus-Einträge.
	Typ	XML-Typ
	Definition und Beispiel	siehe ESLUpdateStatusPagedResult
Parameter:	taskId	ID der Aufgabe
	Stil	PATH
	Erforderlich	Ja
	Typ	UUID
	page	Die Seite der Ergebnisliste.
	Stil	QUERY
	Erforderlich	Nein
	Typ	Integer

recordsPerPage	Die maximale Anzahl von Einträgen pro Seite, welche zurückgegeben werden soll.
Stil	QUERY
Erforderlich	Nein
Typ	Integer

getUpdatesForLabel()

Gibt Informationen zum Update-Status für die Displays an.

Typ	Request									
Methode:	GET									
URL:	/service/updatestatus/label/{label}?page={page}&recordsPerPage={recordsPerPage}									
Antwort:	<table border="1"> <tr> <td>ESLUpdateStatus PagedResult</td> <td>Ergebnisliste der ESLUpdateStatus-Einträge.</td> </tr> <tr> <td>Typ</td> <td>XML-Typ</td> </tr> <tr> <td>Definition und Beispiel</td> <td>siehe ESLUpdateStatusPagedResult</td> </tr> </table>	ESLUpdateStatus PagedResult	Ergebnisliste der ESLUpdateStatus-Einträge.	Typ	XML-Typ	Definition und Beispiel	siehe ESLUpdateStatusPagedResult			
ESLUpdateStatus PagedResult	Ergebnisliste der ESLUpdateStatus-Einträge.									
Typ	XML-Typ									
Definition und Beispiel	siehe ESLUpdateStatusPagedResult									
Parameter:	<table border="1"> <tr> <td>taskId</td> <td>ID der Aufgabe</td> </tr> <tr> <td>Stil</td> <td>PATH</td> </tr> <tr> <td>Erforderlich</td> <td>Ja</td> </tr> <tr> <td>Typ</td> <td>UUID</td> </tr> </table>	taskId	ID der Aufgabe	Stil	PATH	Erforderlich	Ja	Typ	UUID	
taskId	ID der Aufgabe									
Stil	PATH									
Erforderlich	Ja									
Typ	UUID									
	<table border="1"> <tr> <td>page</td> <td>Die Seite der Ergebnisliste.</td> </tr> <tr> <td>Stil</td> <td>QUERY</td> </tr> <tr> <td>Erforderlich</td> <td>Nein</td> </tr> <tr> <td>Typ</td> <td>Integer</td> </tr> </table>	page	Die Seite der Ergebnisliste.	Stil	QUERY	Erforderlich	Nein	Typ	Integer	
page	Die Seite der Ergebnisliste.									
Stil	QUERY									
Erforderlich	Nein									
Typ	Integer									
	<table border="1"> <tr> <td>recordsPerPage</td> <td>Die maximale Anzahl von Einträgen pro Seite, welche zurückgegeben werden soll.</td> </tr> <tr> <td>Stil</td> <td>QUERY</td> </tr> <tr> <td>Erforderlich</td> <td>Nein</td> </tr> <tr> <td>Typ</td> <td>Integer</td> </tr> </table>	recordsPerPage	Die maximale Anzahl von Einträgen pro Seite, welche zurückgegeben werden soll.	Stil	QUERY	Erforderlich	Nein	Typ	Integer	
recordsPerPage	Die maximale Anzahl von Einträgen pro Seite, welche zurückgegeben werden soll.									
Stil	QUERY									
Erforderlich	Nein									
Typ	Integer									

getUpdatesForTransaction()

Gibt Informationen zum Update-Status für Transaktionen an.

Typ	Request	
Methode:	GET	
URL:	/service/updatestatus/transaction/{transactionId}?page={page}&recordsPerPage={recordsPerPage}	

15 UpdateStatusResource

Antwort:	<table border="1"> <tr> <td>ESLUpdateStatus PagedResult</td> <td>Ergebnisliste der ESLUpdateStatus-Einträge.</td> </tr> <tr> <td>Typ</td> <td>XML-Typ</td> </tr> <tr> <td>Definition und Beispiel</td> <td>siehe ESLUpdateStatusPagedResult</td> </tr> </table>	ESLUpdateStatus PagedResult	Ergebnisliste der ESLUpdateStatus-Einträge.	Typ	XML-Typ	Definition und Beispiel	siehe ESLUpdateStatusPagedResult		
ESLUpdateStatus PagedResult	Ergebnisliste der ESLUpdateStatus-Einträge.								
Typ	XML-Typ								
Definition und Beispiel	siehe ESLUpdateStatusPagedResult								
Parameter:	<table border="1"> <tr> <td>taskId</td> <td>ID der Aufgabe</td> </tr> <tr> <td>Stil</td> <td>PATH</td> </tr> <tr> <td>Erforderlich</td> <td>Ja</td> </tr> <tr> <td>Typ</td> <td>UUID</td> </tr> </table>	taskId	ID der Aufgabe	Stil	PATH	Erforderlich	Ja	Typ	UUID
	taskId	ID der Aufgabe							
	Stil	PATH							
Erforderlich	Ja								
Typ	UUID								
<table border="1"> <tr> <td>page</td> <td>Die Seite der Ergebnisliste.</td> </tr> <tr> <td>Stil</td> <td>QUERY</td> </tr> <tr> <td>Erforderlich</td> <td>Nein</td> </tr> <tr> <td>Typ</td> <td>Integer</td> </tr> </table>	page	Die Seite der Ergebnisliste.	Stil	QUERY	Erforderlich	Nein	Typ	Integer	
page	Die Seite der Ergebnisliste.								
Stil	QUERY								
Erforderlich	Nein								
Typ	Integer								
<table border="1"> <tr> <td>recordsPerPage</td> <td>Die maximale Anzahl von Einträgen pro Seite, welche zurückgegeben werden soll.</td> </tr> <tr> <td>Stil</td> <td>QUERY</td> </tr> <tr> <td>Erforderlich</td> <td>Nein</td> </tr> <tr> <td>Typ</td> <td>Integer</td> </tr> </table>	recordsPerPage	Die maximale Anzahl von Einträgen pro Seite, welche zurückgegeben werden soll.	Stil	QUERY	Erforderlich	Nein	Typ	Integer	
recordsPerPage	Die maximale Anzahl von Einträgen pro Seite, welche zurückgegeben werden soll.								
Stil	QUERY								
Erforderlich	Nein								
Typ	Integer								

getWaitingUpdates()

Gibt alle ausstehenden Updates zurück.

Typ	Request								
Methode:	GET								
URL:	/service/updatestatus/waiting?page={page}&recordsPerPage={recordsPerPage}								
Antwort:	<table border="1"> <tr> <td>ESLUpdateStatus PagedResult</td> <td>Ergebnisliste der ESLUpdateStatus-Einträge.</td> </tr> <tr> <td>Typ</td> <td>XML-Typ</td> </tr> <tr> <td>Definition und Beispiel</td> <td>siehe ESLUpdateStatusPagedResult</td> </tr> </table>	ESLUpdateStatus PagedResult	Ergebnisliste der ESLUpdateStatus-Einträge.	Typ	XML-Typ	Definition und Beispiel	siehe ESLUpdateStatusPagedResult		
ESLUpdateStatus PagedResult	Ergebnisliste der ESLUpdateStatus-Einträge.								
Typ	XML-Typ								
Definition und Beispiel	siehe ESLUpdateStatusPagedResult								
Parameter:	<table border="1"> <tr> <td>page</td> <td>Die Seite der Ergebnisliste.</td> </tr> <tr> <td>Stil</td> <td>QUERY</td> </tr> <tr> <td>Erforderlich</td> <td>Nein</td> </tr> <tr> <td>Typ</td> <td>Integer</td> </tr> </table>	page	Die Seite der Ergebnisliste.	Stil	QUERY	Erforderlich	Nein	Typ	Integer
	page	Die Seite der Ergebnisliste.							
Stil	QUERY								
Erforderlich	Nein								
Typ	Integer								
<table border="1"> <tr> <td>recordsPerPage</td> <td>Die maximale Anzahl von Einträgen pro Seite, welche zurückgegeben werden soll.</td> </tr> <tr> <td>Stil</td> <td>QUERY</td> </tr> </table>	recordsPerPage	Die maximale Anzahl von Einträgen pro Seite, welche zurückgegeben werden soll.	Stil	QUERY					
recordsPerPage	Die maximale Anzahl von Einträgen pro Seite, welche zurückgegeben werden soll.								
Stil	QUERY								

	Erforderlich	Nein
	Typ	Integer

ESLUpdateStatusPagedResult

Ergebnisliste der ESLUpdateStatus-Einträge.

Typ: XML-Typ

Schema:

```
<xs:schema version="1.0">
  <xs:element name="PagedResult" type="PagedResult"/>
  <xs:element name="UpdateStatus" type="UpdateStatus"/>
  <xs:element name="UpdateStatusPagedResult" type="UpdateStatusPagedResult"/>
  <xs:complexType name="UpdateStatusPagedResult">
    <xs:complexContent>
      <xs:extension base="PagedResult">
        <xs:sequence>
          <xs:element ref="UpdateStatus" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="PagedResult" abstract="true">
    <xs:sequence/>
    <xs:attribute name="records" type="xs:int" use="required"/>
    <xs:attribute name="totalRecords" type="xs:int" use="required"/>
    <xs:attribute name="totalPages" type="xs:int" use="required"/>
    <xs:attribute name="page" type="xs:int" use="required"/>
    <xs:attribute name="recordsPerPage" type="xs:int" use="required"/>
  </xs:complexType>
  <xs:complexType name="UpdateStatus">
    <xs:sequence>
      <xs:element name="LabelId" type="xs:string"/>
      <xs:element name="TaskType" type="taskType"/>
      <xs:element name="Page" type="xs:int" minOccurs="0"/>
      <xs:element name="TaskPriority" type="taskPriority"/>
      <xs:element name="ExternalId" type="xs:long" minOccurs="0"/>
      <xs:element name="TransactionId" type="xs:long"/>
      <xs:element name="Status" type="status"/>
      <xs:element name="ErrorCode" type="xs:int" minOccurs="0"/>
      <xs:element name="AccessPointId" type="xs:int" minOccurs="0"/>
      <xs:element name="PowerStatus" type="powerStatus" minOccurs="0"/>
      <xs:element name="Rssi" type="xs:int" minOccurs="0"/>
      <xs:element name="Lqi" type="xs:int" minOccurs="0"/>
      <xs:element name="TransmissionTime" type="xs:int" minOccurs="0"/>
      <xs:element name="RetriesLeft" type="xs:int"/>
      <xs:element name="LastChangedAt" type="xs:dateTime" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:string" use="required"/>
    <xs:attribute name="finished" type="xs:boolean" use="required"/>
  </xs:complexType>
  <xs:simpleType name="taskType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="IMAGE"/>
      <xs:enumeration value="PRELOAD_IMAGE"/>
      <xs:enumeration value="FIRMWARE"/>
      <xs:enumeration value="SWITCH_PAGE"/>
      <xs:enumeration value="REFRESH_DISPLAY"/>
      <xs:enumeration value="QUERY_SENSOR"/>
      <xs:enumeration value="PROGRAM_KEYS"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

```

    <xs:enumeration value="SET_CONFIG" />
    <xs:enumeration value="GET_CONFIG" />
    <xs:enumeration value="RESET_ALL_COUNTERS" />
    <xs:enumeration value="RESET_COUNTER" />
    <xs:enumeration value="QUERY_COUNTER" />
    <xs:enumeration value="FORCE_CHANNEL_SCAN" />
    <xs:enumeration value="REBOOT" />
    <xs:enumeration value="PING" />
    <xs:enumeration value="REGISTER_LABEL" />
    <xs:enumeration value="UNREGISTER_LABEL" />
    <xs:enumeration value="UNLOCK_LABEL" />
    <xs:enumeration value="UNKNOWN" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="taskPriority">
  <xs:restriction base="xs:string">
    <xs:enumeration value="LOW" />
    <xs:enumeration value="NORMAL" />
    <xs:enumeration value="HIGH" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="status">
  <xs:restriction base="xs:string">
    <xs:enumeration value="WAITING" />
    <xs:enumeration value="DELAYED" />
    <xs:enumeration value="REPLACE_REQUESTED" />
    <xs:enumeration value="CANCEL_REQUESTED" />
    <xs:enumeration value="SUCCESSFUL" />
    <xs:enumeration value="REPLACED" />
    <xs:enumeration value="CANCELED" />
    <xs:enumeration value="REMOVED" />
    <xs:enumeration value="FAILED" />
    <xs:enumeration value="TIMEOUT" />
    <xs:enumeration value="ERROR" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="powerStatus">
  <xs:restriction base="xs:string">
    <xs:enumeration value="GOOD" />
    <xs:enumeration value="LOW" />
    <xs:enumeration value="BAD" />
    <xs:enumeration value="UNKNOWN" />
  </xs:restriction>
</xs:simpleType>
</xs:schema>

```

Beispiel:

```

<UpdateStatusPagedResult records="1" totalRecords="1" totalPages="1" page="0"
recordsPerPage="1">
  <UpdateStatus id="4d22f521-7fb9-4333-a085-698e315e57a3" finished="false">
    <LabelId>B3001634</LabelId>
    <TaskType>IMAGE</TaskType>
    <Page>0</Page>
    <TaskPriority>HIGH</TaskPriority>
    <ExternalId>200202392</ExternalId>
    <TransactionId>5401</TransactionId>
    <Status>FAILED</Status>
    <AccessPointId>10078</AccessPointId>
    <PowerStatus>GOOD</PowerStatus>
    <Rssi>-76</Rssi>
    <Lqi>24</Lqi>
    <TransmissionTime>840</TransmissionTime>
    <RetriesLeft>4</RetriesLeft>
    <LastChangedAt>2014-07-22T08:42:36.282+02:00</LastChangedAt>
  </UpdateStatus>
</UpdateStatusPagedResult>

```



```
</UpdateStatus>
</UpdateStatusPagedResult>
```

ESLUpdateStatus

Der aktuelle Status eines Auftrags.

Typ: XML-Typ

Schema:

```
<xs:schema version="1.0">
  <xs:element name="UpdateStatus" type="UpdateStatus"/>
  <xs:complexType name="UpdateStatus">
    <xs:sequence>
      <xs:element name="LabelId" type="xs:string"/>
      <xs:element name="TaskType" type="taskType"/>
      <xs:element name="Page" type="xs:int" minOccurs="0"/>
      <xs:element name="TaskPriority" type="taskPriority"/>
      <xs:element name="ExternalId" type="xs:long" minOccurs="0"/>
      <xs:element name="TransactionId" type="xs:long"/>
      <xs:element name="Status" type="status"/>
      <xs:element name="ErrorCode" type="xs:int" minOccurs="0"/>
      <xs:element name="AccessPointId" type="xs:int" minOccurs="0"/>
      <xs:element name="PowerStatus" type="powerStatus" minOccurs="0"/>
      <xs:element name="Rssi" type="xs:int" minOccurs="0"/>
      <xs:element name="Lqi" type="xs:int" minOccurs="0"/>
      <xs:element name="TransmissionTime" type="xs:int" minOccurs="0"/>
      <xs:element name="RetriesLeft" type="xs:int"/>
      <xs:element name="LastChangedAt" type="xs:dateTime" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:string" use="required"/>
    <xs:attribute name="finished" type="xs:boolean" use="required"/>
  </xs:complexType>
  <xs:simpleType name="taskType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="IMAGE"/>
      <xs:enumeration value="PRELOAD_IMAGE"/>
      <xs:enumeration value="FIRMWARE"/>
      <xs:enumeration value="SWITCH_PAGE"/>
      <xs:enumeration value="REFRESH_DISPLAY"/>
      <xs:enumeration value="QUERY_SENSOR"/>
      <xs:enumeration value="PROGRAM_KEYS"/>
      <xs:enumeration value="SET_CONFIG"/>
      <xs:enumeration value="GET_CONFIG"/>
      <xs:enumeration value="RESET_ALL_COUNTERS"/>
      <xs:enumeration value="RESET_COUNTER"/>
      <xs:enumeration value="QUERY_COUNTER"/>
      <xs:enumeration value="FORCE_CHANNEL_SCAN"/>
      <xs:enumeration value="REBOOT"/>
      <xs:enumeration value="PING"/>
      <xs:enumeration value="REGISTER_LABEL"/>
      <xs:enumeration value="UNREGISTER_LABEL"/>
      <xs:enumeration value="UNLOCK_LABEL"/>
      <xs:enumeration value="UNKNOWN"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="taskPriority">
    <xs:restriction base="xs:string">
      <xs:enumeration value="LOW"/>
      <xs:enumeration value="NORMAL"/>
      <xs:enumeration value="HIGH"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="status">
```

```
<xs:restriction base="xs:string">
  <xs:enumeration value="WAITING" />
  <xs:enumeration value="DELAYED" />
  <xs:enumeration value="REPLACE_REQUESTED" />
  <xs:enumeration value="CANCEL_REQUESTED" />
  <xs:enumeration value="SUCCESSFUL" />
  <xs:enumeration value="REPLACED" />
  <xs:enumeration value="CANCELED" />
  <xs:enumeration value="REMOVED" />
  <xs:enumeration value="FAILED" />
  <xs:enumeration value="TIMEOUT" />
  <xs:enumeration value="ERROR" />
</xs:restriction>
</xs:simpleType>
<xs:simpleType name="powerStatus">
  <xs:restriction base="xs:string">
    <xs:enumeration value="GOOD" />
    <xs:enumeration value="LOW" />
    <xs:enumeration value="BAD" />
    <xs:enumeration value="UNKNOWN" />
  </xs:restriction>
</xs:simpleType>
</xs:schema>
```

Beispiel:

```
<UpdateStatus id="3ed2216c-dc92-4fda-8c1a-292fc24903ff" finished="false">
  <LabelId>B3001634</LabelId>
  <TaskType>IMAGE</TaskType>
  <Page>0</Page>
  <TaskPriority>HIGH</TaskPriority>
  <ExternalId>200202392</ExternalId>
  <TransactionId>5401</TransactionId>
  <Status>FAILED</Status>
  <AccessPointId>10078</AccessPointId>
  <PowerStatus>GOOD</PowerStatus>
  <Rssi>-76</Rssi>
  <Lqi>24</Lqi>
  <TransmissionTime>840</TransmissionTime>
  <RetriesLeft>4</RetriesLeft>
  <LastChangedAt>2014-07-22T08:42:36.752+02:00</LastChangedAt>
</UpdateStatus>
```