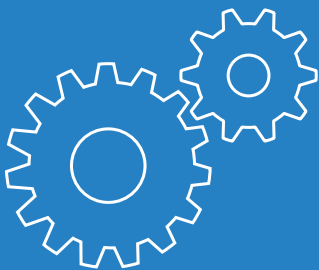


LANCOM Wireless ePaper Server API Reference



Contents

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

This resource provides status information about all of the access points registered at the server. The status information includes details such as the access point ID, IP address, used radio channel, and connection status.

Table 1: Overview

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

getAccessPoints()

Returns status information about all registered access points.

Type	Request									
Method:	GET									
URL:	/service/accesspointinfo?page={page}&recordsPerPage={recordsPerPage}									
Answer:	<table border="1"> <tbody> <tr> <td>ESLAccessPoint PagedResult</td> <td>Result list from ESLAccessPoint.</td> </tr> <tr> <td>Type</td> <td>XML type</td> </tr> <tr> <td>Definition and example</td> <td>See ESLAccessPointPagedResult</td> </tr> </tbody> </table>	ESLAccessPoint PagedResult	Result list from ESLAccessPoint.	Type	XML type	Definition and example	See ESLAccessPointPagedResult			
ESLAccessPoint PagedResult	Result list from ESLAccessPoint.									
Type	XML type									
Definition and example	See ESLAccessPointPagedResult									
Parameters:	<table border="1"> <tbody> <tr> <td>page</td> <td>The page of the result list.</td> </tr> <tr> <td>Style</td> <td>QUERY</td> </tr> <tr> <td>Mandatory</td> <td>No</td> </tr> <tr> <td>Type</td> <td>Integer</td> </tr> </tbody> </table>	page	The page of the result list.	Style	QUERY	Mandatory	No	Type	Integer	
page	The page of the result list.									
Style	QUERY									
Mandatory	No									
Type	Integer									
	<table border="1"> <tbody> <tr> <td>recordsPerPage</td> <td>The maximum number of entries per page to be returned.</td> </tr> <tr> <td>Style</td> <td>QUERY</td> </tr> <tr> <td>Mandatory</td> <td>No</td> </tr> <tr> <td>Type</td> <td>Integer</td> </tr> </tbody> </table>	recordsPerPage	The maximum number of entries per page to be returned.	Style	QUERY	Mandatory	No	Type	Integer	
recordsPerPage	The maximum number of entries per page to be returned.									
Style	QUERY									
Mandatory	No									
Type	Integer									

getAccessPointsWithChannelConflict()

Returns status information about all registered access points with channel conflicts. A channel conflict occurs when a channel is selected by more than one access point.

Type	Request
------	---------

1 AccessPointInfoResource

Method:	GET									
URL:	/service/accesspointinfo/channelconflict?page={page}&recordsPerPage={recordsPerPage}									
Answer:	ESLAccessPoint PagedResult	Result list from ESLAccessPoint.								
	Type	XML type								
	Definition and example	See ESLAccessPointPagedResult								
Parameters:	<table border="1"> <tr> <td>page</td> <td>The page of the result list.</td> </tr> <tr> <td>Style</td> <td>QUERY</td> </tr> <tr> <td>Mandatory</td> <td>No</td> </tr> <tr> <td>Type</td> <td>Integer</td> </tr> </table>		page	The page of the result list.	Style	QUERY	Mandatory	No	Type	Integer
page	The page of the result list.									
Style	QUERY									
Mandatory	No									
Type	Integer									
	<table border="1"> <tr> <td>recordsPerPage</td> <td>The maximum number of entries per page to be returned.</td> </tr> <tr> <td>Style</td> <td>QUERY</td> </tr> <tr> <td>Mandatory</td> <td>No</td> </tr> <tr> <td>Type</td> <td>Integer</td> </tr> </table>		recordsPerPage	The maximum number of entries per page to be returned.	Style	QUERY	Mandatory	No	Type	Integer
recordsPerPage	The maximum number of entries per page to be returned.									
Style	QUERY									
Mandatory	No									
Type	Integer									

getAccessPointsWithConnectionStatus()

Returns status information about all registered access points, filtered by connection status.

Type	Request									
Method:	GET									
URL:	/service/accesspointinfo/status/{connectionStatus}?page={page}&recordsPerPage={recordsPerPage}									
Answer:	ESLAccessPoint PagedResult	Result list from ESLAccessPoint.								
	Type	XML type								
	Definition and example	See ESLAccessPointPagedResult								
Parameters:	<table border="1"> <tr> <td>connectionStatus</td> <td>The connection status to be filtered.</td> </tr> <tr> <td>Style</td> <td>PATH</td> </tr> <tr> <td>Mandatory</td> <td>Yes</td> </tr> <tr> <td>Type</td> <td>ConnectionStatus</td> </tr> </table>		connectionStatus	The connection status to be filtered.	Style	PATH	Mandatory	Yes	Type	ConnectionStatus
connectionStatus	The connection status to be filtered.									
Style	PATH									
Mandatory	Yes									
Type	ConnectionStatus									
	<table border="1"> <tr> <td>page</td> <td>The page of the result list.</td> </tr> <tr> <td>Style</td> <td>QUERY</td> </tr> <tr> <td>Mandatory</td> <td>No</td> </tr> <tr> <td>Type</td> <td>Integer</td> </tr> </table>		page	The page of the result list.	Style	QUERY	Mandatory	No	Type	Integer
page	The page of the result list.									
Style	QUERY									
Mandatory	No									
Type	Integer									

recordsPerPage	The maximum number of entries per page to be returned.
Style	QUERY
Mandatory	No
Type	Integer

getAccessPointsWithoutChannel()

Returns status information for all registered access points that have not selected a channel.

Type	Request	
Method:	GET	
URL:	/service/accesspointinfo/nochannel?page={page}&recordsPerPage={recordsPerPage}	
Answer:	ESLAccessPointPagedResult	Result list from ESLAccessPoint.
	Type	XML type
	Definition and example	See ESLAccessPointPagedResult
Parameters:	page	The page of the result list.
	Style	QUERY
	Mandatory	No
	Type	Integer
	recordsPerPage	The maximum number of entries per page to be returned.
	Style	QUERY
	Mandatory	No
	Type	Integer

ESLAccessPointPagedResult

Result list from ESLAccessPoint.

Type: XML type

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>

```

Example:

```

<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

The connection status to be filtered.

Type: List type

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>

```

List values:

The list value can be used as a path or query parameter.

Value	Description
ONLINE	Device is connected
UNKNOWN	Not possible to determine the connection status
OFFLINE	Device is not connected

2 AccessPointResource

This resource manages the access points. Options include; display registered access points, register new access points, revoke the registration, and update the configuration of an access point.

Table 2: Overview

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

getRegisteredAccessPoints()

Returns all registered access points.

Type	Request																	
Method:	GET																	
URL:	/service/accesspoint?page={page}&recordsPerPage={recordsPerPage}																	
Answer:	<table border="1"> <tbody> <tr> <td>ESLAccessPointRegistrationPagedResult</td> <td>Result list from ESLAccessPointRegistration.</td> </tr> <tr> <td>Type</td> <td>XML type</td> </tr> <tr> <td>Definition and example</td> <td>See ESLAccessPointRegistrationPagedResult</td> </tr> </tbody> </table>		ESLAccessPointRegistrationPagedResult	Result list from ESLAccessPointRegistration.	Type	XML type	Definition and example	See ESLAccessPointRegistrationPagedResult										
ESLAccessPointRegistrationPagedResult	Result list from ESLAccessPointRegistration.																	
Type	XML type																	
Definition and example	See ESLAccessPointRegistrationPagedResult																	
Parameters:	<table border="1"> <tbody> <tr> <td>page</td> <td>The page of the result list.</td> </tr> <tr> <td>Style</td> <td>QUERY</td> </tr> <tr> <td>Mandatory</td> <td>No</td> </tr> <tr> <td>Type</td> <td>Integer</td> </tr> </tbody> </table> <table border="1"> <tbody> <tr> <td>recordsPerPage</td> <td>The maximum number of entries per page to be returned.</td> </tr> <tr> <td>Style</td> <td>QUERY</td> </tr> <tr> <td>Mandatory</td> <td>No</td> </tr> <tr> <td>Type</td> <td>Integer</td> </tr> </tbody> </table>		page	The page of the result list.	Style	QUERY	Mandatory	No	Type	Integer	recordsPerPage	The maximum number of entries per page to be returned.	Style	QUERY	Mandatory	No	Type	Integer
page	The page of the result list.																	
Style	QUERY																	
Mandatory	No																	
Type	Integer																	
recordsPerPage	The maximum number of entries per page to be returned.																	
Style	QUERY																	
Mandatory	No																	
Type	Integer																	

registerAccessPoints()

Registers new access points. This requires information such as the ID or the server address of the access point.

Type	Request
Method:	POST

URL:	/service/accesspoint	
Request:	ESLAccessPoint Registration	Access point registration
	Type	XML type
	Definition and example	See ESLAccessPointRegistration
Answer:	Type	void

unregisterAccessPoints()

Revokes the registration of an access point with the corresponding accessPointID.

Type	Request	
Method:	DELETE	
URL:	/service/accesspoint/{accessPointId}	
Answer:	Type	void
Parameters:	accessPointId	The ID of the access point
	Style	PATH
	Mandatory	Yes
	Type	Integer

ESLAccessPointRegistrationPagedResult

Result list of the ESLAccessPointRegistration entries.

Type: XML type

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>
```

Example:

```
<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

Configuration information for the access point.

Type: XML type

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>
```

Example:

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

3 ConfigurationResource

This resource manages the configurations. Configurations can be retrieved individually or all together. Also, modified configurations can be retrieved and new configurations can be set.

Table 3: Overview

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

get Configuration()

Returns information about configurations. The information about a configuration is; the key, the type, the default value, and the currently set value.

Type	Request	
Method:	GET	
URL:	/service/configuration?page={page}&recordsPerPage={recordsPerPage}	
Answer:	ESLConfiguration PagedResult	Result list of the ESLConfiguration entries.
	Type	XML type
	Definition and example	See ESLConfigurationPagedResult
Parameters:	page	The page of the result list.
	Style	QUERY
	Mandatory	No
	Type	Integer
	recordsPerPage	The maximum number of entries per page to be returned.
	Style	QUERY
	Mandatory	No
	Type	Integer

getConfiguration()

Returns information about a configuration as specified by the key.

Type	Request
------	---------

3 ConfigurationResource

Method:	GET	
URL:	/service/configuration/{key}	
Answer:	ESLConfiguration Type Definition and example	Configuration including the key, the data type, the default value, the set value, and the permitted data types. XML type See ESLConfiguration
Parameters:	key Style Mandatory Type	Key of the attribute PATH Yes String

getModifiedConfiguration()

Returns information about modified configurations.

Type	Request	
Method:	GET	
URL:	/service/configuration/modified?page={page}&recordsPerPage={recordsPerPage}	
Answer:	ESLConfiguration PagedResult Type Definition and example	Result list of the ESLConfiguration entries. XML type See ESLConfigurationPagedResult
Parameters:	page Style Mandatory Type	The page of the result list. QUERY No Integer
	recordsPerPage Style Mandatory Type	The maximum number of entries per page to be returned. QUERY No Integer

setConfiguration()

The configuration as specified by the key is set to another value.

Type	Request
Method:	PUT

URL:	/service/configuration/{key}?value={value}	
Answer:	Type	void
Parameters:	key	Key of the configuration
	Style	QUERY
	Mandatory	Yes
	Type	Integer
	value	The value to be used
	Style	QUERY
	Mandatory	No
	Type	Integer

ESLConfigurationPagedResult

Result list of the ESLConfiguration entries.

Type: XML type

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>

```

Example:

```

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

```

ESLConfiguration

Configuration including the key, the data type, the default value, the set value, and the permitted data types.

Type: XML type

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>

```

Example:

```

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

```

4 LabelInfoResource

This resource returns status information for all displays known to the system.

Table 4: Overview

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

getLabel()

Returns status information for a display.

Type	Request	
Method:	GET	
URL:	/service/labelinfo/{label}	
Answer:	ESLLabelInfo	Information about the labels
	Type	XML type
	Definition and example	See ESLLabelInfo
Parameters:	label	The Label ID the respective display is identified with in the Wireless ePaper Server.
	Style	PATH
	Mandatory	Yes
	Type	String

getLabels()

Returns status information for all known displays.

Type	Request
Method:	GET
URL:	/service/labelinfo?page={page}&recordsPerPage={recordsPerPage}

4 LabelInfoResource

Answer:	ESLabelInfo PagedResult Type Definition and example	Result list of the ESLabelInfo entries. XML type See ESLabelInfoPagedResult
Parameters:	page Style Mandatory Type	The page of the result list. QUERY No Integer
	recordsPerPage Style Mandatory Type	The maximum number of entries per page to be returned. QUERY No Integer

getLabelsWithConnectionStatus()

Returns status information for all of the known displays that have a connection status.

Type	Request	
Method:	GET	
URL:	/service/labelinfo/status/{connectionStatus}?page={page}&recordsPerPage={recordsPerPage}	
Answer:	ESLabelInfo PagedResult Type Definition and example	Result list of the ESLabelInfo entries. XML type See ESLabelInfoPagedResult
Parameters:	connectionStatus Style Mandatory Type	The connection status to be filtered. PATH Yes ConnectionStatus
	page Style Mandatory Type	The page of the result list. QUERY No Integer
	recordsPerPage Style	The maximum number of entries per page to be returned. QUERY

Mandatory	No
Type	Integer

getLabelsWithErrors()

Returns status information for all of the known displays that have errors.

Type	Request	
Method:	GET	
URL:	/service/labelinfo/error?page={page}&recordsPerPage={recordsPerPage}	
Answer:	ESLabelInfo PagedResult	Result list of the ESLabelInfo entries.
	Type	XML type
	Definition and example	See ESLabelInfoPagedResult
Parameters:	page	The page of the result list.
	Style	QUERY
	Mandatory	No
	Type	Integer
	recordsPerPage	The maximum number of entries per page to be returned.
	Style	QUERY
	Mandatory	No
	Type	Integer

getLabelsWithPowerStatus()

Returns status information for all of the known displays that have a power status.

Type	Request	
Method:	GET	
URL:	/service/labelinfo/powerstatus/{powerStatus}?page={page}&recordsPerPage={recordsPerPage}	
Answer:	ESLabelInfo PagedResult	Result list of the ESLabelInfo entries.
	Type	XML type
	Definition and example	See ESLabelInfoPagedResult
Parameters:	powerStatus	Indicates the power status of a Display. If the Display is powered by batteries, this shows the remaining battery charge.
	Style	PATH

4 LabelInfoResource

Mandatory	Yes
Type	PowerStatus
page	The page of the result list.
Style	QUERY
Mandatory	No
Type	Integer
recordsPerPage	The maximum number of entries per page to be returned.
Style	QUERY
Mandatory	No
Type	Integer

getLabelsWithProblems()

Returns status information for all of the known displays that have errors.

Type	Request	
Method:	GET	
URL:	/service/labelinfo/problem?page={page}&recordsPerPage={recordsPerPage}	
Answer:	ESLabelInfo PagedResult	Result list of the ESLabelInfo entries.
	Type	XML type
	Definition and example	See ESLabelInfoPagedResult
Parameters:	page	The page of the result list.
	Style	QUERY
	Mandatory	No
	Type	Integer
	recordsPerPage	The maximum number of entries per page to be returned.
	Style	QUERY
	Mandatory	No
	Type	Integer

getLabelsWithTag()

Returns status information for all of the known displays that have tags.

Type	Request
Method:	GET

URL:	/service/labelinfo/tag/{tag}?page={page}&recordsPerPage={recordsPerPage}	
Answer:	ESLabelInfo PagedResult Type Definition and example	Result list of the ESLabelInfo entries. XML type See ESLabelInfoPagedResult
Parameters:	tag Style Mandatory Type	Designation of the label PATH Yes String
	page Style Mandatory Type	The page of the result list. QUERY No Integer
	recordsPerPage Style Mandatory Type	The maximum number of entries per page to be returned. QUERY No Integer

getLabelsWithUnsuccessfulLastUpdate()

Returns status information for all known displays for which the last update was unsuccessful.

Type	Request	
Method:	GET	
URL:	/service/labelinfo/unsuccessful?page={page}&recordsPerPage={recordsPerPage}	
Answer:	ESLabelInfo PagedResult Type Definition and example	Result list of the ESLabelInfo entries. XML type See ESLabelInfoPagedResult
Parameters:	page Style Mandatory Type	The page of the result list. QUERY No Integer

recordsPerPage	The maximum number of entries per page to be returned.
Style	QUERY
Mandatory	No
Type	Integer

getLabelsWithoutEncryption()

Returns status information for all known displays that are not encrypted.

Type	Request	
Method:	GET	
URL:	/service/labelinfo/unencrypted?page={page}&recordsPerPage={recordsPerPage}	
Answer:	ESLLabelInfo PagedResult	Result list of the ESLLabelInfo entries.
	Type	XML type
	Definition and example	See ESLLabelInfoPagedResult
Parameters:	page	The page of the result list.
	Style	QUERY
	Mandatory	No
	Type	Integer
	recordsPerPage	The maximum number of entries per page to be returned.
	Style	QUERY
	Mandatory	No
	Type	Integer

ESLLabelInfo

Information about the displays.

Type: XML type

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>

```

Example:

```

<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

Result list of the ESLabelInfo entries.

Type: XML type

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>

```

Example:

```

<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

The connection status to be filtered.

Type: List type

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>
```

List values:

The list value can be used as a path or query parameter.

Value	Description
ONLINE	Device is connected
UNKNOWN	Not possible to determine the connection status
OFFLINE	Device is not connected

PowerStatus

Indicates the power status of a Display. If the Display is powered by batteries, this shows the remaining battery charge.

Type: List type

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>
```

List values:

The list value can be used as a path or query parameter.

Value	Description
GOOD	Good power status
LOW	Battery replacement required in the next few weeks
BAD	Immediate battery replacement necessary
UNKNOWN	No power-sensor value available

5 LabelResource

This resource manages the displays. It enables you to show the already registered displays, to register new displays, invoke registrations, and unlock the displays.

Table 5: Overview

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()

Returns all registered displays.

Type	Request	
Method:	GET	
URL:	/service/label	
Answer:	ESLLabelList	List of the ESLLabel entries.
	Type	XML type
	Definition and example	See ESLLabelList

registerLabel()

Registers a new Display.

Type	Request	
Method:	GET	
URL:	/service/label/register/{registrationCode}	
Answer:	ESLTransaction	A refresh transaction to the Display.
	Type	XML type
	Definition and example	See ESLTransaction
Parameters:	registrationCode	The registration code the respective display is registered at the Wireless ePaper Server.
	Style	PATH
	Mandatory	Yes

Type	String
------	--------

registerLabels()

Registers several new displays given in the ESLLabelList.

Type	Request	
Method:	POST	
URL:	/service/label	
Request:	ESLLabelList	List of the ESLLabel entries.
	Type	XML type
	Definition and example	See ESLLabelList
Answer:	ESLTransaction	A refresh transaction to the Display.
	Type	XML type
	Definition and example	See ESLTransaction

unlockLabels()

Unlocks displays by specifying one or more UnlockCodes in an ESLUnlockCodeList.

Type	Request	
Method:	POST	
URL:	/service/label/unlock	
Request:	ESLUnlockCodeList	List of the ESLUnlockCode entries.
	Type	XML type
	Definition and example	See ESLUnlockCodeList
Answer:	ESLTransaction	A refresh transaction to the Display.
	Type	XML type
	Definition and example	See ESLTransaction

unregisterLabel()

Revokes the registration of a Display as specified by the labelID.

Type	Request	
Method:	DELETE	
URL:	/service/label/{labelID}	
Answer:	ESLTransaction	A refresh transaction to the Display.
	Type	XML type

	Definition and example	See ESLTransaction
Parameters:	labelID	ID of the Display
	Style	PATH
	Mandatory	Yes
	Type	String

unregisterLabels()

Revokes the registration of multiple displays as given in the ESLLabelList.

Type	Request	
Method:	POST	
URL:	/service/label/unregister	
Request:	ESLLabelList	List of the ESLLabel entries.
	Type	XML type
	Definition and example	See ESLLabelList
Answer:	ESLTransaction	A refresh transaction to the Display.
	Type	XML type
	Definition and example	See ESLTransaction

ESLLabelList

List of the ESLLabel entries.

Type: XML type

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>
```

Example:

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

ESLTransaction

A refresh transaction to the Display.

Type: XML type

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

List of the ESLUnlockCode entries.

Type: XML type

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>
```

Example:

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

6 Level1ExportResource

This resource manages the exports on the first level. This includes the export of access points, configurations, displays, and licenses.

Table 6: Overview

Method	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()

Exports all known access points.

Type	Request	
Method:	GET	
URL:	/service/export/level1/accesspoint	
Answer:	ESLAccessPointRegistrationList	List of the ESLAccessPointRegistration entries.
	Type	XML type
	Definition and example	See ESLAccessPointRegistrationList

exportConfigurationKeys()

Exports all known configurations.

Type	Request	
Method:	GET	
URL:	/service/export/level1/configuration	
Answer:	ESLConfigurationList	List of the ESLConfiguration entries.
	Type	XML type
	Definition and example	See ESLConfigurationList

exportLabels()

Exports all known displays.

Type	Request	
Method:	GET	

URL:	/service/export/level1/label	
Answer:	ESLLabelList	List of the ESLLabel entries.
	Type	XML type
	Definition and example	See ESLLabelList

exportLicenses()

Exports all known licenses.

Type	Request	
Method:	GET	
URL:	/service/export/level1/license	
Answer:	ESLLicenseList	List of the ESLLicense entries.
	Type	XML type
	Definition and example	See ESLLicenseList

ESLAccessPointRegistrationList

List of the ESLAccessPointRegistration entries.

Type: XML type

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>
```

Example:

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

ESLConfigurationList

List of the ESLConfiguration entries.

Type: XML type

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>

```

Example:

```

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

```

ESLLabelList

List of the ESLLabel entries.

Type: XML type

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>

```

Example:

```

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

```



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

ESLLicenseList

List of the ESLLicense entries.

Type: XML type

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>
```

Example:

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

7 Level1ImportResource

This resource manages the imports on the first level. This includes the import of access points, configurations, displays, and licenses.

Table 7: Overview

Method	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()

Imports the access points, which are specified in the `ESLAccessPointRegistrationList`.

Type	Request									
Method:	POST									
URL:	/service/import/level1/accesspoint?clear={clear}									
Request:	<table border="1"> <tbody> <tr> <td>ESLAccessPointRegistrationList</td> <td>List of the <code>ESLAccessPointRegistration</code> entries.</td> </tr> <tr> <td>Type</td> <td>XML type</td> </tr> <tr> <td>Definition and example</td> <td>See ESLAccessPointRegistrationList</td> </tr> </tbody> </table>		ESLAccessPointRegistrationList	List of the <code>ESLAccessPointRegistration</code> entries.	Type	XML type	Definition and example	See ESLAccessPointRegistrationList		
ESLAccessPointRegistrationList	List of the <code>ESLAccessPointRegistration</code> entries.									
Type	XML type									
Definition and example	See ESLAccessPointRegistrationList									
Answer:	<table border="1"> <tbody> <tr> <td>Type</td> <td>ESLMessage</td> </tr> </tbody> </table>		Type	ESLMessage						
Type	ESLMessage									
Parameters:	<table border="1"> <tbody> <tr> <td>Parameter</td> <td>clear</td> </tr> <tr> <td>Style</td> <td>QUERY</td> </tr> <tr> <td>Mandatory</td> <td>Yes</td> </tr> <tr> <td>Type</td> <td>Boolean</td> </tr> </tbody> </table>		Parameter	clear	Style	QUERY	Mandatory	Yes	Type	Boolean
Parameter	clear									
Style	QUERY									
Mandatory	Yes									
Type	Boolean									

importConfigurationKeys()

Imports the configurations that are specified in the `ESLConfigurationList`.

Type	Request			
Method:	POST			
URL:	/service/import/level1/configuration?clear={clear}			
Request:	<table border="1"> <tbody> <tr> <td>ESLConfigurationList</td> <td>List of the <code>ESLConfiguration</code> entries.</td> </tr> </tbody> </table>		ESLConfigurationList	List of the <code>ESLConfiguration</code> entries.
ESLConfigurationList	List of the <code>ESLConfiguration</code> entries.			

	Type	XML type
	Definition and example	See ESLConfigurationList
Answer:	Type	ESLMessage
Parameters:	Parameter	clear
	Style	QUERY
	Mandatory	Yes
	Type	Boolean

importLabels()

Imports the displays that are specified in the ESLLabelList.

Type	Request	
Method:	POST	
URL:	/service/import/level1/label?clear={clear}	
Request:	ESLLabelList	List of the ESLLabel entries.
	Type	XML type
	Definition and example	See ESLLabelList
Answer:	Type	ESLMessage
Parameters:	Parameter	clear
	Style	QUERY
	Mandatory	Yes
	Type	Boolean

importLicenses()

Imports the licenses that are specified in the ESLLicenseList.

Type	Request	
Method:	POST	
URL:	/service/import/level1/license?clear={clear}	
Request:	ESLLicenseList	List of the ESLLicense entries.
	Type	XML type
	Definition and example	See ESLLicenseList
Answer:	Type	ESLMessage

Parameters:	Parameter	clear
	Style	QUERY
	Mandatory	Yes
	Type	Boolean

ESLAccessPointRegistrationList

List of the ESLAccessPointRegistration entries.

Type: XML type

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>
```

Example:

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

ESLConfigurationList

List of the ESLConfiguration entries.

Type: XML type

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>

```

Example:

```

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

```

ESLLabelList

List of the ESLLabel entries.

Type: XML type

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>

```

Example:

```

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

```

ESLLicenseList

List of the ESLLicense entries.

Type: XML type

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>
```

Example:

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

8 LicenseResource

This resource manages the licenses. This is used to show stored licenses, save new licenses, and delete all of the licenses.

Table 8: Overview

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

getLicenses()

Returns all stored licenses with comments and the key value of each license.

Type	Request	
Method:	GET	
URL:	/service/license?page={page}&recordsPerPage={recordsPerPage}	
Answer:	ESLicense PagedResult	Result list of the ESLicense entries.
	Type	XML type
	Definition and example	See ESLicensePagedResult
Parameters:	page	The page of the result list.
	Style	QUERY
	Mandatory	No
	Type	Integer
	recordsPerPage	The maximum number of entries per page to be returned.
	Style	QUERY
	Mandatory	No
	Type	Integer

removeAllLicenses()

Deletes all licenses.

Type	Request
Method:	DELETE
URL:	/service/license

Answer:	Type	void
---------	------	------

storeLicenses()

Stores a new license along with a comment about the license and the key value of the license.

Type	Request	
Method:	POST	
URL:	/service/license	
Request:	ESLLicense Type Definition and example	Representation of a server license with comment and key value. XML type See ESLLicense
Answer:	Type	void

ESLLicensePagedResultList

Result list of the ESLLicense entries.

Type: XML type

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>
```

Example:

```
<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

Representation of a server license with comment and key value.

Type: XML type

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>
```

Example:

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

9 ProblemResource

This resource provides information about all problems.

Table 9: Overview

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

getProblems()

Returns information about all known problems.

Type	Request							
Method:	GET							
URL:	/service/problem							
Answer:	<table border="1"> <tbody> <tr> <td>ESLProblemList</td> <td>List of the ESLProblem entries.</td> </tr> <tr> <td>Type</td> <td>XML type</td> </tr> <tr> <td>Definition and example</td> <td>See ESLProblemList</td> </tr> </tbody> </table>		ESLProblemList	List of the ESLProblem entries.	Type	XML type	Definition and example	See ESLProblemList
ESLProblemList	List of the ESLProblem entries.							
Type	XML type							
Definition and example	See ESLProblemList							

ESLProblemList

List of the ESLProblem entries.

Type: XML type

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>
```

Example:

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

10 ServiceStatusResource

This resource provides information about the service status.

Table 10: Overview

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

getServiceStatus()

Returns information about the service status.

Type	Request							
Method:	GET							
URL:	/service/status							
Answer:	<table border="1"> <tbody> <tr> <td>ESLServiceStatus</td> <td>The status of the service</td> </tr> <tr> <td>Type</td> <td>XML type</td> </tr> <tr> <td>Definition and example</td> <td>See ESLServiceStatus</td> </tr> </tbody> </table>		ESLServiceStatus	The status of the service	Type	XML type	Definition and example	See ESLServiceStatus
ESLServiceStatus	The status of the service							
Type	XML type							
Definition and example	See ESLServiceStatus							

ESLServiceStatus

The service status.

Type: XML type

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

This resource manages the tags. It can be used to add tags to displays or remove tags from them, and to retrieve an overview of labels that are already in use.

Table 11: Overview

Method	URL	Name
POST	/service/tag/add	<i>addTagsToLabels</i>
GET	/service/tag	<i>getTags</i>
POST	/service/tag/remove	<i>removeTagsFromLabels</i>

addTagsToLabels()

Adds a label to the displays that are specified in the ESLLabelList.

Type	Request							
Method:	POST							
URL:	/service/tag/add							
Request:	<table border="1"> <tbody> <tr> <td>ESLLabelList</td> <td>List of the ESLLabel entries.</td> </tr> <tr> <td>Type</td> <td>XML type</td> </tr> <tr> <td>Definition and example</td> <td>See ESLLabelList</td> </tr> </tbody> </table>		ESLLabelList	List of the ESLLabel entries.	Type	XML type	Definition and example	See ESLLabelList
ESLLabelList	List of the ESLLabel entries.							
Type	XML type							
Definition and example	See ESLLabelList							
Answer:	<table border="1"> <tbody> <tr> <td>Type</td> <td>void</td> </tr> </tbody> </table>		Type	void				
Type	void							

getTags()

Returns information about all known tags.

Type	Request									
Method:	GET									
URL:	/service/tag?page={page}&recordsPerPage={recordsPerPage}									
Answer:	<table border="1"> <tbody> <tr> <td>ESLTagPagedResult</td> <td>Result list of the ESLTag entries.</td> </tr> <tr> <td>Type</td> <td>XML type</td> </tr> <tr> <td>Definition and example</td> <td>See ESLTagPagedResult</td> </tr> </tbody> </table>		ESLTagPagedResult	Result list of the ESLTag entries.	Type	XML type	Definition and example	See ESLTagPagedResult		
ESLTagPagedResult	Result list of the ESLTag entries.									
Type	XML type									
Definition and example	See ESLTagPagedResult									
Parameters:	<table border="1"> <tbody> <tr> <td>page</td> <td>The page of the result list.</td> </tr> <tr> <td>Style</td> <td>QUERY</td> </tr> <tr> <td>Mandatory</td> <td>No</td> </tr> <tr> <td>Type</td> <td>Integer</td> </tr> </tbody> </table>		page	The page of the result list.	Style	QUERY	Mandatory	No	Type	Integer
page	The page of the result list.									
Style	QUERY									
Mandatory	No									
Type	Integer									

recordsPerPage	The maximum number of entries per page to be returned.
Style	QUERY
Mandatory	No
Type	Integer

removeTagsFromLabels()

Removes a label from the displays given in the ESLLabelList.

Type	Request	
Method:	POST	
URL:	/service/tag/remove	
Answer:	ESLLabelList	List of the ESLLabel entries.
	Type	XML type
	Definition and example	See ESLLabelList
Answer:	Type	void

ESLLabelList

List of the ESLLabel entries.

Type: XML type

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>
```

Example:

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

ESLTagPagedResult

Result list of the ESLTag entries.

Type: XML type

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>
```

Example:

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

12 TaskResource

This resource places orders for rendering display contents.

Table 12: Overview

Method	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()

Type	Request							
Method:	POST							
URL:	/service/task/preview/document							
Request:	<table border="1"> <tbody> <tr> <td>ESLTemplateTask</td> <td>This task renders a graphic using the specified style and the content document associated with the task. The rendered image is then transferred to the display.</td> </tr> <tr> <td>Type</td> <td>XML type</td> </tr> <tr> <td>Definition and example</td> <td>See ESLTemplateTask</td> </tr> </tbody> </table>	ESLTemplateTask	This task renders a graphic using the specified style and the content document associated with the task. The rendered image is then transferred to the display.	Type	XML type	Definition and example	See ESLTemplateTask	
ESLTemplateTask	This task renders a graphic using the specified style and the content document associated with the task. The rendered image is then transferred to the display.							
Type	XML type							
Definition and example	See ESLTemplateTask							
Answer:	<table border="1"> <tbody> <tr> <td>Content-type: application/xml</td> <td></td> </tr> <tr> <td>Type</td> <td>XML type</td> </tr> </tbody> </table>	Content-type: application/xml		Type	XML type			
Content-type: application/xml								
Type	XML type							

getRenderingImage()

Type	Request							
Method:	POST							
URL:	/service/task/preview/image							
Request:	<table border="1"> <tbody> <tr> <td>ESLTemplateTask</td> <td>This task renders a graphic using the specified style and the content document associated with the task. The rendered image is then transferred to the display.</td> </tr> <tr> <td>Type</td> <td>XML type</td> </tr> <tr> <td>Definition and example</td> <td>See ESLTemplateTask</td> </tr> </tbody> </table>	ESLTemplateTask	This task renders a graphic using the specified style and the content document associated with the task. The rendered image is then transferred to the display.	Type	XML type	Definition and example	See ESLTemplateTask	
ESLTemplateTask	This task renders a graphic using the specified style and the content document associated with the task. The rendered image is then transferred to the display.							
Type	XML type							
Definition and example	See ESLTemplateTask							
Answer:	<table border="1"> <tbody> <tr> <td>Content type: image/png</td> <td></td> </tr> <tr> <td>Type</td> <td>PNG type</td> </tr> </tbody> </table>	Content type: image/png		Type	PNG type			
Content type: image/png								
Type	PNG type							

getRenderingSourceRecord()

Type	Request						
Method:	POST						
URL:	/service/task/preview/source						
Request:	<table border="1"> <tr> <td>ESLTemplateTask</td> <td>This task renders a graphic using the specified style and the content document associated with the task. The rendered image is then transferred to the display.</td> </tr> <tr> <td>Type</td> <td>XML type</td> </tr> <tr> <td>Definition and example</td> <td>See ESLTemplateTask</td> </tr> </table>	ESLTemplateTask	This task renders a graphic using the specified style and the content document associated with the task. The rendered image is then transferred to the display.	Type	XML type	Definition and example	See ESLTemplateTask
ESLTemplateTask	This task renders a graphic using the specified style and the content document associated with the task. The rendered image is then transferred to the display.						
Type	XML type						
Definition and example	See ESLTemplateTask						
Answer:	<table border="1"> <tr> <td>Content-type: application/xml</td> <td></td> </tr> <tr> <td>Type</td> <td>XML type</td> </tr> </table>	Content-type: application/xml		Type	XML type		
Content-type: application/xml							
Type	XML type						

scheduleTasks()

Schedules up one or more tasks for execution. A task instructs the server to send an update to a particular display. Depending on the task type, the displayed image is changed or a different action is performed on the display.

Type	Request						
Method:	POST						
URL:	/service/task						
Request:	<table border="1"> <tr> <td>ESLTaskOrder</td> <td>A task order contains a list of tasks that are grouped together and processed in one transaction.</td> </tr> <tr> <td>Type</td> <td>XML type</td> </tr> <tr> <td>Definition and example</td> <td>See ESLTaskOrder</td> </tr> </table>	ESLTaskOrder	A task order contains a list of tasks that are grouped together and processed in one transaction.	Type	XML type	Definition and example	See ESLTaskOrder
ESLTaskOrder	A task order contains a list of tasks that are grouped together and processed in one transaction.						
Type	XML type						
Definition and example	See ESLTaskOrder						
Answer:	<table border="1"> <tr> <td>ESLTransaction</td> <td>A refresh transaction to the display.</td> </tr> <tr> <td>Type</td> <td>XML type</td> </tr> <tr> <td>Definition and example</td> <td>See ESLTransaction</td> </tr> </table>	ESLTransaction	A refresh transaction to the display.	Type	XML type	Definition and example	See ESLTransaction
ESLTransaction	A refresh transaction to the display.						
Type	XML type						
Definition and example	See ESLTransaction						

ESLTemplateTask

This task renders a graphic using the specified style and the content document associated with the task. The rendered image is then transferred to the display.

Type: XML type

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:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:schema>
```

```

    </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>

```

Example:

```

<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

A task order contains a list of tasks that are grouped together and processed in one transaction.

Type: XML type

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: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:extension>
    </xs:complexContent>
  </xs:complexType>

```

```

    <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>

```

Example:

```

<TaskOrder title="Perform some example tasks">
  <ImageTask page="3" preload="true" labelId="B1001C4A" taskPriority="NORMAL"
externalId="4711">
    <Image>iVBORw0KGgoAAAANSUUhEUgAAQgAAACwCAIAAAB8emcEAAACVElEQVR42u3ZoQ0
AQAgEweu/ab4BcIhPGPQoklWXDFfD8fwFH4/geWHwvDB4Xhg8LwyeFwbPC4Pnvw
jDg3i+8R7B88LgeWHwvDB4Xhg8LwyeFwbP2zd43vLN88LgeWHwvDB4XhgexwvD4

```

```

3heGDxvr+B5yzfPC4PnhcHwzUB5YfC8MHheGDxv3+B5yzfPC8PjeGF4HC8Mj+B5
YfC8MHheGDxvr+B5yzfPC4PnhcHwzUB5YfC8MHheGDxv3/Ag3vLtcTvwDJ4XBs8
Lg+eFwfPC4Hlh8Ly9guct3zvwDJ4XBs8Lg+eF4XG8MDyOF4ZH8Lx9g+ct3zvwDJ
4XBs8Lg+eFwfPC4Hlh8Ly9guct3zvwDJ4XhsfxwvA4nhcGzwuD54XB8/YNnrD88
7weF4YPC8MnhcGzwuD54XB8/YKnrD8exwvDI/jeWHwvDB4Xhg8LwyeFwbP2zd4
3vLN88LgeWHwvDB4Xhg8Lwyp44XhcbY9woN43vLN88LgeWHwvDB4Xhg8LwyeFwb
P2zd43vLN88LgeWHwvDA8jheGx/G8MHheGDxvr+B5yzfPC4PnhcHwzUB5YfC8MH
heGDxv3+B5y7fH8cLwOF4YHsHwzUB5YfC8MHheGDxvr+B5yzfPC4PnhcHwzUB5Y
fC8MHheGB7H8x7E85ZvnhcGzwuD54XB88LgeWHwvDB4317B85ZvnhcGzwuD54Xh
cbwwPI4XhkfwvDB43r7B85ZvnhcGzwuD54XB88LgeWHwvDB4317B85ZvneGx/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

A refresh transaction to the display.

Type: XML type

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

This resource provides information about all templates.

Table 13: Overview

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

getInvalidTemplates()

Returns information about invalid templates.

Type	Request	
Method:	GET	
URL:	/service/template/invalid?page={page}&recordsPerPage={recordsPerPage}	
Answer:	ESLTemplatePagedResult	Result list of the ESLTemplate entries.
	Type	XML type
	Definition and example	See ESLTemplatePagedResult
Parameters:	page	The page of the result list.
	Style	QUERY
	Mandatory	No
	Type	Integer
	recordsPerPage	The maximum number of entries per page to be returned.
	Style	QUERY
	Mandatory	No
	Type	Integer

getTemplates()

Returns information about all templates.

Type	Request	
Method:	GET	
URL:	/service/template?page={page}&recordsPerPage={recordsPerPage}	
Answer:	ESLTemplatePagedResult	Result list of the ESLTemplate entries.
	Type	XML type

	Definition and example	See ESLTemplatePagedResult
Parameters:	page	The page of the result list.
	Style	QUERY
	Mandatory	No
	Type	Integer
	recordsPerPage	The maximum number of entries per page to be returned.
	Style	QUERY
	Mandatory	No
	Type	Integer

ESLTemplatePagedResult

Result list of the ESLTemplate entries.

Type: XML type

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>
```

Example:

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

14 TransactionResource

This resource manages the transactions. Transactions can be aborted or their status can be displayed.

Table 14: Overview

Method	URL	Name
DELETE	/service/transaction/{transactionId}	<i>abortTransaction</i>
GET	/service/transaction/{transactionId}/status	<i>getTransactionStatus</i>

abortTransaction()

Aborts the transaction with the corresponding transactionId.

Type	Request	
Method:	DELETE	
URL:	/service/transaction/{transactionId}	
Answer:	Type	void
Parameters:	transactionId	The transaction ID
	Style	PATH
	Mandatory	Yes
	Type	long

getTransactionStatus()

Returns the status information about all known transactions.

Type	Request	
Method:	GET	
URL:	/service/transaction/{transactionId}/status	
Answer:	ESLTransactionStatusInfo	Status information about a transaction
	Type	XML type
	Definition and example	See ESLTransactionStatusInfo
Parameters:	transactionId	The transaction ID
	Style	PATH
	Mandatory	Yes
	Type	long

ESLTransactionStatusInfo

Status information about a transaction.

Type: XML type

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

This resource provides information about the update status.

Table 15: Overview

Method	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()

Aborts the task specified by the taskId.

Type	Request	
Method:	DELETE	
URL:	/service/updatestatus/{taskId}	
Answer:	Type	void
Parameters:	taskId	ID of the task
	Style	PATH
	Mandatory	Yes
	Type	UUID

getAllUpdates()

Returns information about all known updates.

Type	Request
Method:	GET
URL:	/service/updatestatus?page={page}&recordsPerPage={recordsPerPage}

15 UpdateStatusResource

Answer:	<table border="1"> <tr> <td>ESLUpdateStatus PagedResult</td> <td>Result list of the ESLUpdateStatus entries.</td> </tr> <tr> <td>Type</td> <td>XML type</td> </tr> <tr> <td>Definition and example</td> <td>See ESLUpdateStatusPagedResult</td> </tr> </table>	ESLUpdateStatus PagedResult	Result list of the ESLUpdateStatus entries.	Type	XML type	Definition and example	See ESLUpdateStatusPagedResult										
ESLUpdateStatus PagedResult	Result list of the ESLUpdateStatus entries.																
Type	XML type																
Definition and example	See ESLUpdateStatusPagedResult																
Parameters:	<table border="1"> <tr> <td>page</td> <td>The page of the result list.</td> </tr> <tr> <td>Style</td> <td>QUERY</td> </tr> <tr> <td>Mandatory</td> <td>No</td> </tr> <tr> <td>Type</td> <td>Integer</td> </tr> </table> <table border="1"> <tr> <td>recordsPerPage</td> <td>The maximum number of entries per page to be returned.</td> </tr> <tr> <td>Style</td> <td>QUERY</td> </tr> <tr> <td>Mandatory</td> <td>No</td> </tr> <tr> <td>Type</td> <td>Integer</td> </tr> </table>	page	The page of the result list.	Style	QUERY	Mandatory	No	Type	Integer	recordsPerPage	The maximum number of entries per page to be returned.	Style	QUERY	Mandatory	No	Type	Integer
page	The page of the result list.																
Style	QUERY																
Mandatory	No																
Type	Integer																
recordsPerPage	The maximum number of entries per page to be returned.																
Style	QUERY																
Mandatory	No																
Type	Integer																

getImageForTask()

Returns the image associated with the task.

Type	Request								
Method:	GET								
URL:	<code>/service/updatestatus/{taskId}/image</code>								
Answer:	<table border="1"> <tr> <td>Content type: image/png</td> <td></td> </tr> <tr> <td>Type</td> <td>PNG</td> </tr> </table>	Content type: image/png		Type	PNG				
Content type: image/png									
Type	PNG								
Parameters:	<table border="1"> <tr> <td>taskId</td> <td>ID of the task</td> </tr> <tr> <td>Style</td> <td>PATH</td> </tr> <tr> <td>Mandatory</td> <td>Yes</td> </tr> <tr> <td>Type</td> <td>UUID</td> </tr> </table>	taskId	ID of the task	Style	PATH	Mandatory	Yes	Type	UUID
taskId	ID of the task								
Style	PATH								
Mandatory	Yes								
Type	UUID								

getUnsuccessfulUpdates()

Returns all failed updates.

Type	Request		
Method:	GET		
URL:	<code>/service/updatestatus/unsuccessful?page={page}&recordsPerPage={recordsPerPage}</code>		
Answer:	<table border="1"> <tr> <td>ESLUpdateStatus PagedResult</td> <td>Result list of the ESLUpdateStatus entries.</td> </tr> </table>	ESLUpdateStatus PagedResult	Result list of the ESLUpdateStatus entries.
ESLUpdateStatus PagedResult	Result list of the ESLUpdateStatus entries.		

	Type	XML type
	Definition and example	See ESLUpdateStatusPagedResult
Parameters:	page	The page of the result list.
	Style	QUERY
	Mandatory	No
	Type	Integer
	recordsPerPage	The maximum number of entries per page to be returned.
	Style	QUERY
	Mandatory	No
	Type	Integer

getUpdateDetail()

Provides details about the update with the corresponding taskId.

Type	Request	
Method:	GET	
URL:	/service/updatestatus/{taskId}	
Answer:	ESLUpdateStatus	The current status of a task
	Type	XML type
	Definition and example	See ESLUpdateStatus
Parameters:	taskId	ID of the task
	Style	PATH
	Mandatory	Yes
	Type	UUID

getUpdateHistoryForTask()

Provides an overview of previously conducted updates for a task as specified by its taskId.

Type	Request	
Method:	GET	
URL:	/service/updatestatus/{taskId}/history?page={page}&recordsPerPage={recordsPerPage}	
Answer:	ESLUpdateStatus PagedResult	Result list of the ESLUpdateStatus entries.
	Type	XML type

15 UpdateStatusResource

	Definition and example	See ESLUpdateStatusPagedResult
Parameters:	taskId	ID of the task
	Style	PATH
	Mandatory	Yes
	Type	UUID
	page	The page of the result list.
	Style	QUERY
	Mandatory	No
	Type	Integer
	recordsPerPage	The maximum number of entries per page to be returned.
	Style	QUERY
	Mandatory	No
	Type	Integer

getUpdatesForExternalId()

Provides information about the update status for an external ID.

Type	Request	
Method:	GET	
URL:	/service/updatestatus/externalid/{externalId}?page={page}&recordsPerPage={recordsPerPage}	
Answer:	ESLUpdateStatusPagedResult	Result list of the ESLUpdateStatus entries.
	Type	XML type
	Definition and example	See ESLUpdateStatusPagedResult
Parameters:	taskId	ID of the task
	Style	PATH
	Mandatory	Yes
	Type	UUID
	page	The page of the result list.
	Style	QUERY
	Mandatory	No
	Type	Integer

recordsPerPage	The maximum number of entries per page to be returned.
Style	QUERY
Mandatory	No
Type	Integer

getUpdatesForLabel()

Provides information about the update status for the displays.

Type	Request								
Method:	GET								
URL:	/service/updatestatus/label/{label}?page={page}&recordsPerPage={recordsPerPage}								
Answer:	<table border="1"> <tr> <td>ESLUpdateStatus PagedResult</td> <td>Result list of the ESLUpdateStatus entries.</td> </tr> <tr> <td>Type</td> <td>XML type</td> </tr> <tr> <td>Definition and example</td> <td>See ESLUpdateStatusPagedResult</td> </tr> </table>	ESLUpdateStatus PagedResult	Result list of the ESLUpdateStatus entries.	Type	XML type	Definition and example	See ESLUpdateStatusPagedResult		
ESLUpdateStatus PagedResult	Result list of the ESLUpdateStatus entries.								
Type	XML type								
Definition and example	See ESLUpdateStatusPagedResult								
Parameters:	<table border="1"> <tr> <td>taskId</td> <td>ID of the task</td> </tr> <tr> <td>Style</td> <td>PATH</td> </tr> <tr> <td>Mandatory</td> <td>Yes</td> </tr> <tr> <td>Type</td> <td>UUID</td> </tr> </table>	taskId	ID of the task	Style	PATH	Mandatory	Yes	Type	UUID
	taskId	ID of the task							
	Style	PATH							
Mandatory	Yes								
Type	UUID								
<table border="1"> <tr> <td>page</td> <td>The page of the result list.</td> </tr> <tr> <td>Style</td> <td>QUERY</td> </tr> <tr> <td>Mandatory</td> <td>No</td> </tr> <tr> <td>Type</td> <td>Integer</td> </tr> </table>	page	The page of the result list.	Style	QUERY	Mandatory	No	Type	Integer	
page	The page of the result list.								
Style	QUERY								
Mandatory	No								
Type	Integer								
<table border="1"> <tr> <td>recordsPerPage</td> <td>The maximum number of entries per page to be returned.</td> </tr> <tr> <td>Style</td> <td>QUERY</td> </tr> <tr> <td>Mandatory</td> <td>No</td> </tr> <tr> <td>Type</td> <td>Integer</td> </tr> </table>	recordsPerPage	The maximum number of entries per page to be returned.	Style	QUERY	Mandatory	No	Type	Integer	
recordsPerPage	The maximum number of entries per page to be returned.								
Style	QUERY								
Mandatory	No								
Type	Integer								

getUpdatesForTransaction()

Provides information about the update status for transactions.

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

15 UpdateStatusResource

Answer:	<table border="1"> <tr> <td>ESLUpdateStatus PagedResult</td> <td>Result list of the ESLUpdateStatus entries.</td> </tr> <tr> <td>Type</td> <td>XML type</td> </tr> <tr> <td>Definition and example</td> <td>See ESLUpdateStatusPagedResult</td> </tr> </table>	ESLUpdateStatus PagedResult	Result list of the ESLUpdateStatus entries.	Type	XML type	Definition and example	See ESLUpdateStatusPagedResult																		
ESLUpdateStatus PagedResult	Result list of the ESLUpdateStatus entries.																								
Type	XML type																								
Definition and example	See ESLUpdateStatusPagedResult																								
Parameters:	<table border="1"> <tr> <td>taskId</td> <td>ID of the task</td> </tr> <tr> <td>Style</td> <td>PATH</td> </tr> <tr> <td>Mandatory</td> <td>Yes</td> </tr> <tr> <td>Type</td> <td>UUID</td> </tr> </table> <table border="1"> <tr> <td>page</td> <td>The page of the result list.</td> </tr> <tr> <td>Style</td> <td>QUERY</td> </tr> <tr> <td>Mandatory</td> <td>No</td> </tr> <tr> <td>Type</td> <td>Integer</td> </tr> </table> <table border="1"> <tr> <td>recordsPerPage</td> <td>The maximum number of entries per page to be returned.</td> </tr> <tr> <td>Style</td> <td>QUERY</td> </tr> <tr> <td>Mandatory</td> <td>No</td> </tr> <tr> <td>Type</td> <td>Integer</td> </tr> </table>	taskId	ID of the task	Style	PATH	Mandatory	Yes	Type	UUID	page	The page of the result list.	Style	QUERY	Mandatory	No	Type	Integer	recordsPerPage	The maximum number of entries per page to be returned.	Style	QUERY	Mandatory	No	Type	Integer
taskId	ID of the task																								
Style	PATH																								
Mandatory	Yes																								
Type	UUID																								
page	The page of the result list.																								
Style	QUERY																								
Mandatory	No																								
Type	Integer																								
recordsPerPage	The maximum number of entries per page to be returned.																								
Style	QUERY																								
Mandatory	No																								
Type	Integer																								

getWaitingUpdates()

Returns all pending updates.

Type	Request												
Method:	GET												
URL:	<code>/service/updatestatus/waiting?page={page}&recordsPerPage={recordsPerPage}</code>												
Answer:	<table border="1"> <tr> <td>ESLUpdateStatus PagedResult</td> <td>Result list of the ESLUpdateStatus entries.</td> </tr> <tr> <td>Type</td> <td>XML type</td> </tr> <tr> <td>Definition and example</td> <td>See ESLUpdateStatusPagedResult</td> </tr> </table>	ESLUpdateStatus PagedResult	Result list of the ESLUpdateStatus entries.	Type	XML type	Definition and example	See ESLUpdateStatusPagedResult						
ESLUpdateStatus PagedResult	Result list of the ESLUpdateStatus entries.												
Type	XML type												
Definition and example	See ESLUpdateStatusPagedResult												
Parameters:	<table border="1"> <tr> <td>page</td> <td>The page of the result list.</td> </tr> <tr> <td>Style</td> <td>QUERY</td> </tr> <tr> <td>Mandatory</td> <td>No</td> </tr> <tr> <td>Type</td> <td>Integer</td> </tr> </table> <table border="1"> <tr> <td>recordsPerPage</td> <td>The maximum number of entries per page to be returned.</td> </tr> <tr> <td>Style</td> <td>QUERY</td> </tr> </table>	page	The page of the result list.	Style	QUERY	Mandatory	No	Type	Integer	recordsPerPage	The maximum number of entries per page to be returned.	Style	QUERY
page	The page of the result list.												
Style	QUERY												
Mandatory	No												
Type	Integer												
recordsPerPage	The maximum number of entries per page to be returned.												
Style	QUERY												

	Mandatory	No
	Type	Integer

ESLUpdateStatusPagedResult

Result list of the ESLUpdateStatus entries.

Type: XML type

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>

```

Example:

```

<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

The current status of a task.

Type: XML type

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>
```

Example:

```
<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>
```