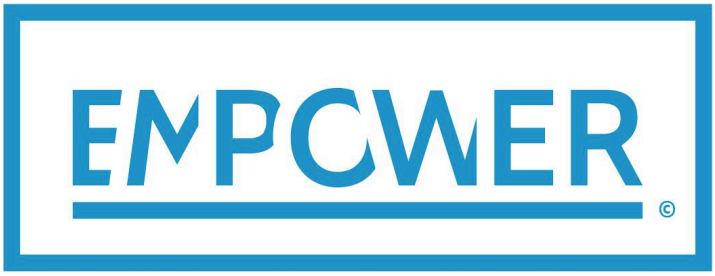


*local **E**lectricity retail **M**arkets for **P**rosumer smart grid p**OWER** services*

Deliverable nº:	**D4.6**
Deliverable name:	**API-functions implemented, tested and verified**
Version:	**1.0**
Release date:	**15/07/2016**
Dissemination level:	**PU** (PU, PP, RE, CO, Internal)
Status:	**Submitted** (Draft, Peer-reviewed, Submitted, Approved)
Author:	**Schneider - Cristóbal Cordobés**
Contributors	Schneider, eSmart

This project has received funding from the European Union's Horizon 2020 Research and Innovation programme under Grant Agreement No 646476.

Document history:

Version	Date of issue	Content and changes	Edited by
1	17/07/16	First draft version	Cristobal cordobes

Peer reviewed by:

Partner	Contributor

Deliverable beneficiaries:

WP / Task	Responsible
WP5	eSmart
WP3	UPC

Table of contents

Abbreviations and Acronyms

Acronym	Description
CA	Consortium Agreement
DoA	Description of Action (annex I of the Grant Agreement)
EC	European Commission
GA	Grant Agreement
PC-A	Project Coordinator-Administrative
PC-T	Project Coordinator-Technical
PMC	Project Management Committee
PO	Project Officer
QM	Quality Management
TMT	Technical Management Team
ToC	Table of Contents
WP	Work Package
WPL	Work Package Leader

Executive summary

This document is an attachment to the deliverable D4.2 where is specified some changes in API´s software and code.

Here is exposed both the interface description from SESP control cloud and local controller and description for Scada webServices.

AI tests and results have been redacted in deliverable ***D4.5_Test results***

1 Telemetry

Local controllers send metering values from sensors either via HTTPS or AMQP to dedicated telemetry endpoint, returned from the Endpoints API call. These messages are received by an Azure Event Hub. Message format: eSmart Event Hub Format and JSON (see: chapter 1.1)

Details on Event Hub is found her: https://azure.microsoft.com/en-us/services/event-hubs/

1.1 Message format JSON

Property name	Type	Optional	Description
deviceid	string	N	This id is a unique identifier for the device
property	string	N	This uniquely identify the property of the value sent. Valid properties are listed in 0.
format	string	Y	Identifies the format – including version – of this message. This field is optional. Current version is: “eSmart.TsCollection.JSON.1.0”
timezone	string	Y	Time zone of the start and end times in the Value object (see below). This is optional, and preferred practice is to include time zone in the time fields
unit	string	Y	Unit of value. This is optional. E.g. “KW”, “MW”, “C”
values	array	N	This is an array of value objects. The value object is described below. The values array is required and must have at least one value object

Valid values for property are (eSmart can map any name to a time series):

- InstantaneousDemand (550 - Power)
- CurrentSummationDelivered (100 – TotalConsumption)
- SwitchPosition (2100 - Switch Position) where:
    - 0 = closed →live
    - 1 = open →not live (no electricity)

1.2 Content in values table:

Property name	Type	Description
starttime	string datetime	Start time of the value. This is normally required, but may be omitted for a device continuously sending instantaneous values. The time is given as an ISO 8601 date time formatted string, including time zone: E.g. “2015-04-30T14:32:52Z” If no time zone is given UTC is assumed.
endtime	string datetime	End time of a value that applies for a duration of time (e.g. metered consumption for an hour, maximum temperature for a day). This is optional and would normally not be included for instantaneous values. The time is given as an ISO 8601 date time formatted string, including time zone: E.g. 2015-04-30T14:32:52Z If no time zone is given UTC is assumed.
value	number	The value. Note that “.” is always used as decimal separator for JSON messages. value is an optional field, but either the value or the message field must be present in a Value object.
message	string	This may either be a status modifier for the value or a notification message from the device. message is an optional field, but either the value or the message field must be present in a Value object.
starttime	string datetime	Start time of the value. This is normally required, but may be omitted for a device continuously sending instantaneous values. The time is given as an ISO 8601 date time formatted string, including time zone: E.g. “2015-04-30T14:32:52Z” If no time zone is given UTC is assumed.
endtime	string datetime	End time of a value that applies for a duration of time (e.g. metered consumption for an hour, maximum temperature for a day). This is optional and would normally not be included for instantaneous values. The time is given as an ISO 8601 date time formatted string, including time zone: E.g. 2015-04-30T14:32:52Z If no time zone is given UTC is assumed.

2 Command and Control

For Command & Control, the local controller creates a AMQP connection to an Azure IoT Hub.

Details on IoT Hub is found here: https://azure.microsoft.com/en-us/services/iot-hub/.

Messages sent from an IoT Hub to a local controller is an extended version of the eSmart Event Hub Format including command [“action”] and message ID ["messageid"]. The command is either ["get"] to get values or ["set"] to set values or alter status. If a command is specific for the local controller (and not a device), the dash sign ["-"] is inserted as ["deviceid"].

2.1 "Get" command

For a GET command from the IoT Hub to a local controller:

- Action is set to “get”
- Property is set to desired content

In the values table

- Value message is set to: “N/A”

The response returned from the local controller contains the same “messageid” and “property”. Action can be omitted in the response. The Values table will contain requested values and a time stamp.

"Get" command example

Request: { messageid: "123", deviceid: "246", action: "get", property: "instantaneousdemand",

values: [{ message: "N/A" }] }

Response: { messageid: "123", deviceid: "246", property: "instantaneousdemand",

unit: "W", values: [{ starttime: "2016-05-09T14:32:52Z", value: "66.99" }] }

2.2 "Set" command

When a SET command causes a status change for a device, e.g. a relay is turned on, in addition to regular procedure (ACK) a telemetry messages is sent with information on the status change.

"Set" command example

Request: {messageid: "123", deviceid: "246", action: "set", property: "switchposition", values: [{starttime: "2016-05-09T14:32:52Z", value: 1 }] }

Response: { messageid: "123", deviceid: "246", property: "switchposition", values: [{ starttime: "2016-05-09T14:32:52Z", value: 1 }] }

Same or equivalent message to telemetry endpoint.

endtime is not used to prevent ambiguity

"Set" command message format

Property name	Type	Optional	Description
messageid	string	N	This id is a unique message identifier, and this message identifier must be included in the ACKnowledge message returned
deviceid	string	N	This id is a unique identifier for the device.
action	String	N	Define the action to execute – valid values are: - "get" - "set"
property	string	N	This uniquely identify the property of the action sent. Valid properties are listed below
format	string	Y	Identifies the format – including version – of this message. This field is optional. Current version is: "eSmart.TsCollection.JSON.1.0"
timezone	string	Y	Time zone of the start and end times in the Value object (see below). This is optional, and preferred practice is to include time zone in the time fields
unit	string	Y	Unit of value. This is optional. E.g. "Watt", "MW", "C"
values	array	N	This is an array of value objects. The value object is described below. The values array is required and must have at least one value object

Value table

Property name	Type	Description
starttime	string datetime	Start time for when to perform the action. If omitted, the action will be executed immediately. The time is given as an ISO 8601 date time formatted string, including time zone: E.g. "2015-04-30T14:32:52Z" If no time zone is given UTC is assumed
endtime	string datetime	The end time for when to end the interval for a control signal. The time is given as an ISO 8601 date time formatted string, including time zone: E.g. 2015-04-30T14:32:52Z If no time zone is given UTC is assumed
value	number	The value. Note that "." is always used as decimal separator for JSON messages. value is an optional field, but either the value or the message field must be present in a Value object.

Property name	Type	Description
		No values are necessary for action “get”, and in this case the message element will be populated with “N/A”. Valid values for action “set” are listed below
message	string	This may either be a status modifier for the value or a notification message from the device. message is an optional field, but either the value or the message field must be present in a Value object

Get "values" (from device) example

Receives the last (1) meter values from a sensor. Property can be the same values that is used regarding telemetry

Request: { messageid: "123", deviceid: "246", action: "get", property: "instantaneousdemand",

values: [{ message: "N/A" }] }

Response: { messageid: "123", deviceid: "246", property: "instantaneousdemand",

unit: "W", values: [{ starttime: "2016-05-09T14:32:52Z", value: "66.99" }] }

Get "ping" (from local controller) example

Checks if the local controller is in operation (available and answers)

Request: { messageid: "123", deviceid: "-", action: "get", property: "ping",

values: [{ message: "N/A" }] }

Response: { messageid: "123", deviceid: "- ", property: "ping",

values: [{ starttime: "2016-05-09T14:32:52Z", message: "pong" }] }

Get softwareVersion (from local controller) example

Gets information on which software versions that are installed on the local controller. The message field will be filled with a comma separated list with versions information.

Request: { messageid: "123", deviceid: "-", action: "get", property: "softwareversion",

values: [{ message: "N/A" }] }

Response: { messageid: "123", deviceid: "-", property: "softwareversion",

values:[{starttime:"2016-05-09T14:32:52Z",message:

"gateway_server_application.release.1.0.4, gateway_os.release.1.0.2.ubifs" }] }

Get trustCenter (from local controller) example

Gets infromasjon on which devices wthat are included in the local controller trust center. The message filed will be filled with a comma separated list with device information (deviceId, deviceTYpe is joined)

Request: { messageid: "123", deviceid: "-", action: "get", property: "trustcenter",

values: [{ message: "N/A" }] }

Response: { messageid: "123", deviceid: "-", property: "trustcenter",

values: [{ starttime: "2016-05-09T14:32:52Z", message: "0015BC001B02172D DP_EMI_LED_IR YES, 0015BC002F000315 DP_SMPLGM_10 NO" }] }

Get log (from gateway) example

Gets the last 50 lines from the local controller log. The message filed will be filled with a comma separated list with log lines.

Request: { messageid: "123", deviceid: "-", action: "get", property: "log",

values: [{ message: "N/A" }] }

Response: { messageid: "123", deviceid: "-", property: "log",

values: [{starttime: "2016-05-09T14:32:52Z", message: "[INFO] 13.06.2016 07:23:25.959 statemachine.devices.AbstractDevice putEvent: Sending [REPORT] event to state machine, [INFO] 13.06.2016 07:23:26.100 connection.ClientConnManager handleRx: Got telegram with source address 123, [INFO] 13.06.2016 07:23:26.132 connection.ClientConnManager handleRx: Forwarding GenericDataIn msg to device: [123]" }] }

Set "values" / command (to device)

Sends configuration values and commands to a device

Request: { messageid: "123", deviceid: "246", action: "set", property: "switchposition",

values: [{ starttime: "2016-05-09T14:32:52Z", value: 1 }] }

Response: { messageid: "123", deviceid: "246", property: "switchposition",

values: [{ starttime: "2016-05-09T14:32:52Z", value: 1 }] }

Same / similar message to telemetry end point

If start time is not specified, the value is set time. If the time of setting one of the values has passed, the value will be set to real-time (implies as quickly as possible). Endtime will not be used to prevent ambiguity.

Set reboot (to local controller)

One is able to send a reboot command to a local controller when there is a need for a restart.

Request: { messageid: "123", deviceid: "-", action: "set", property: "reboot",

values: [{ starttime: "2016-05-09T14:32:52Z", message: "N/A" }] }

Response: { messageid: "123", deviceid: "-", property: "switchposition",

values: [{ starttime: "2016-05-09T14:32:52Z", message: "N/A" }]

3 Scada Web Services

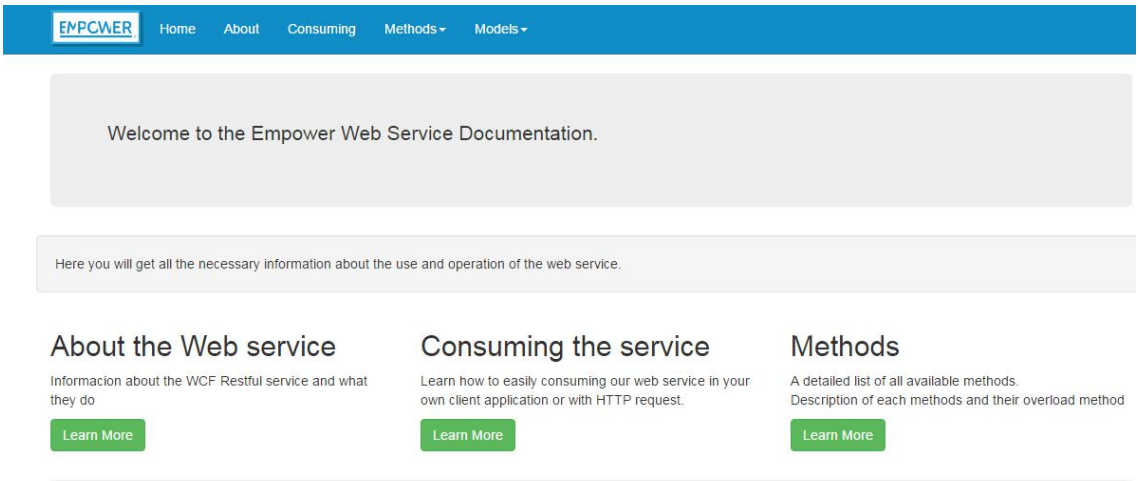
3.1 Communication

For communication between SESP and SCADA it has developed some webservices. Here you will find some developments that has been made in order to facilitate integration between these systems.

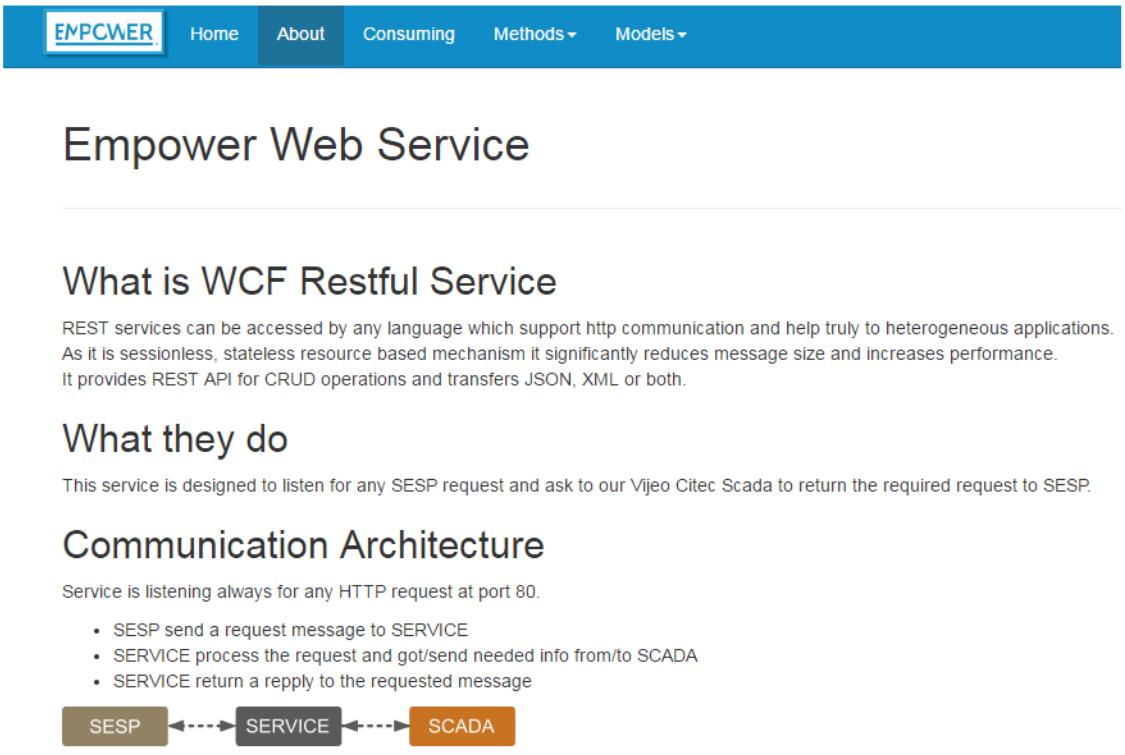
It has been created a Webpage for accessing in case of any question or doubt about the integration. http://195.77.82.74

This could be used as an user manual for developing webservices for connecting with Scada Platform.

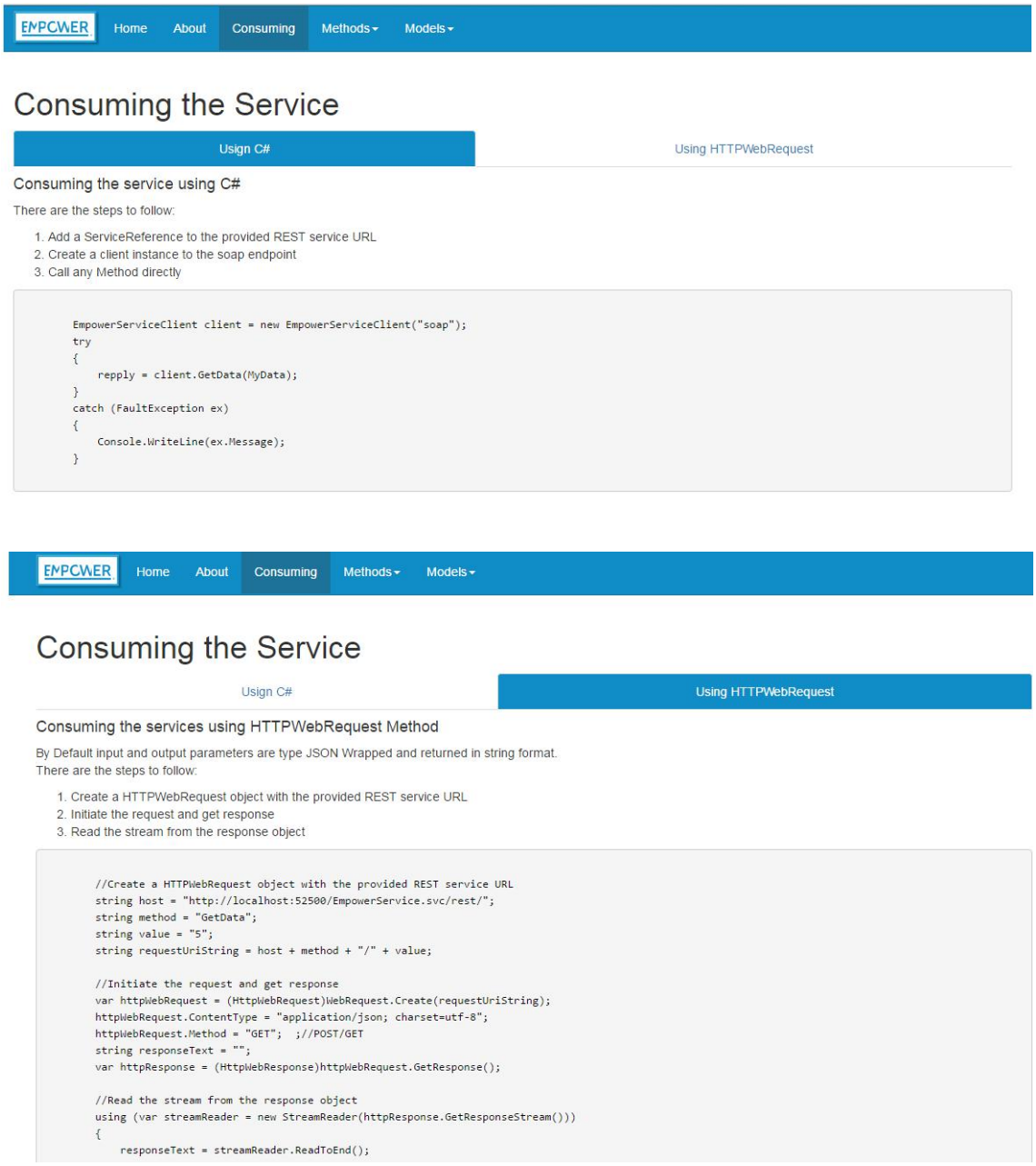
Below it´s shown some screenshots of the Web Page.



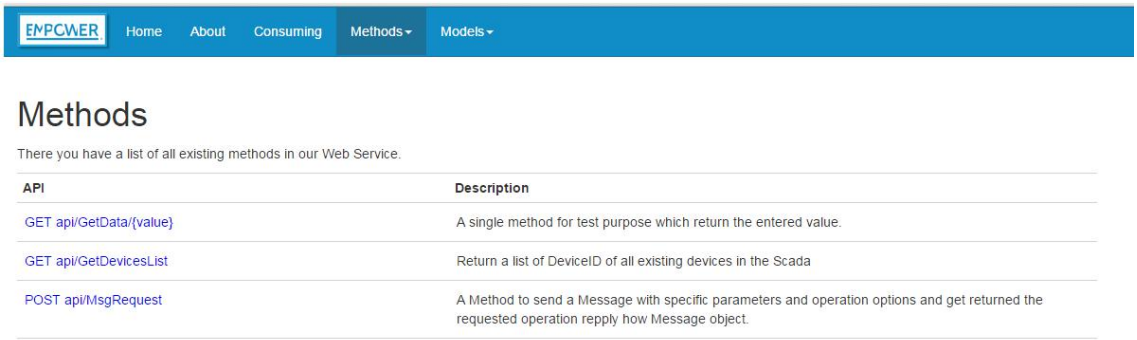
Home Page



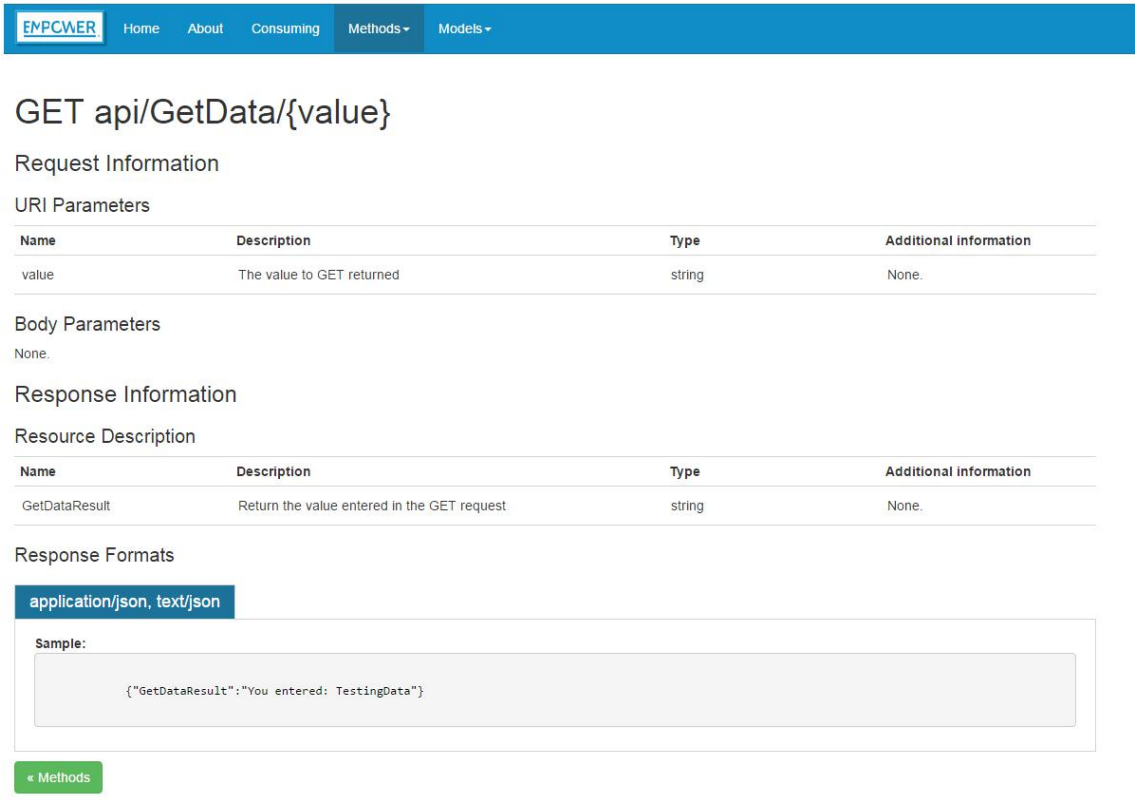
Introduction to the web service



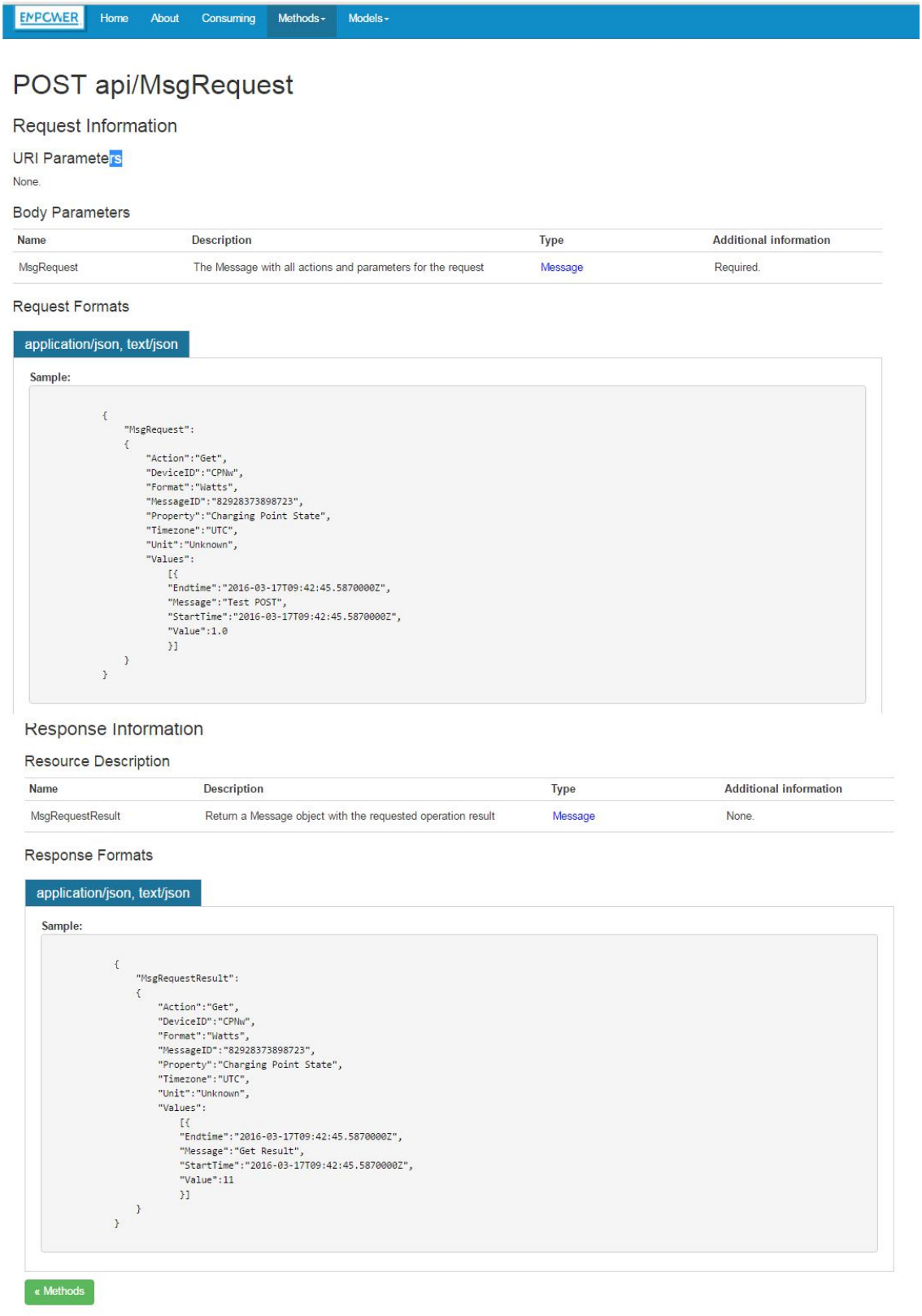
Consuming Service (Http and Json message)



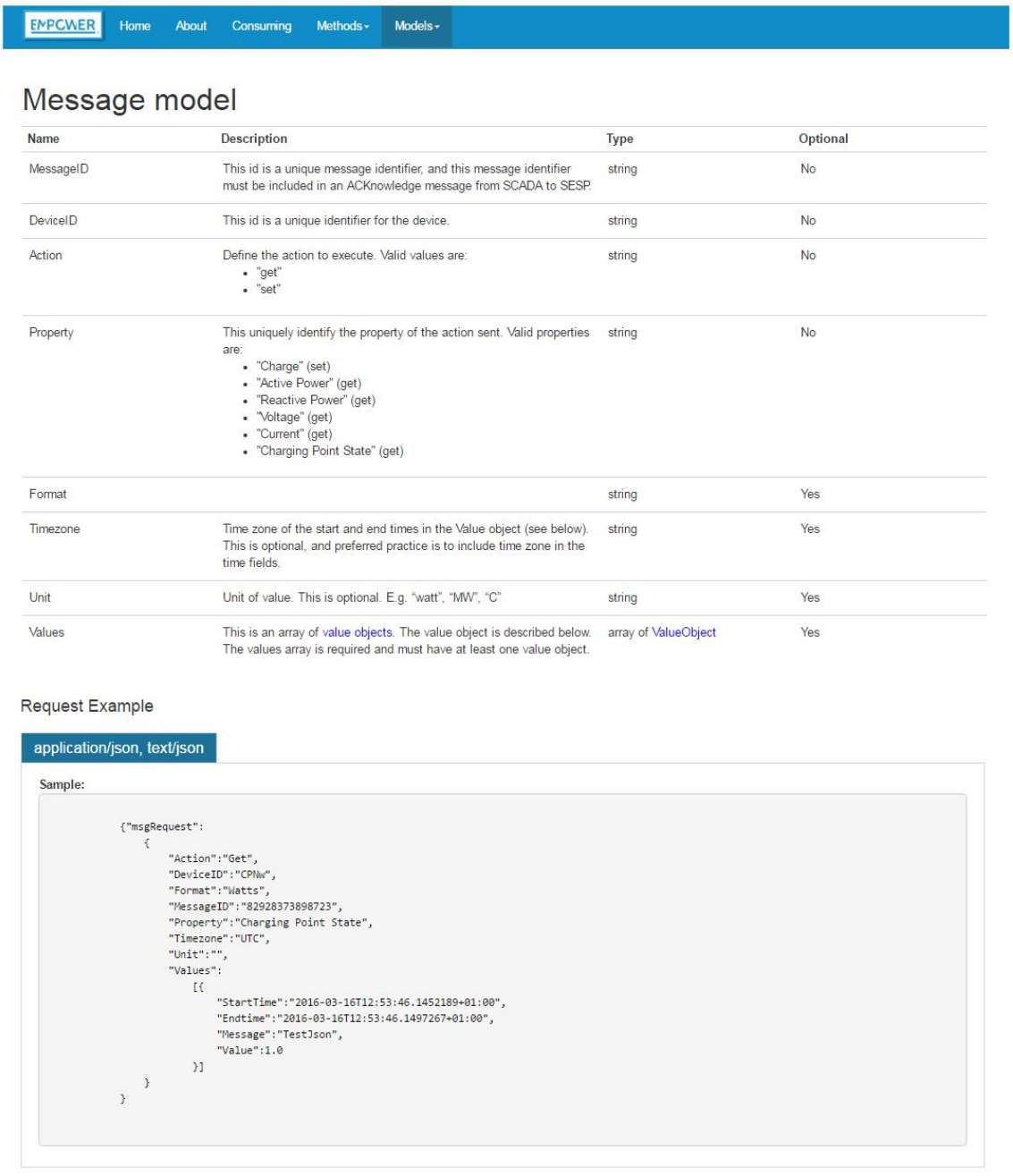
Web Methods



Get Data Method



Request Message and Response



Message Model

ValueObject model

Name	Description	Type	Optional
Starttime	Start time for when to perform the action. If omitted, the action will be executed immediately. The time is given as an ISO 8601 date time formatted string, including time zone: E.g. "2015-04-30T14:32:52Z" If no time zone is given UTC is assumed.	string datetime	No
Endtime	The end time for when to end the interval for a control signal. The time is given as an ISO 8601 date time formatted string, including time zone: E.g. 2015-04-30T14:32:52Z If no time zone is given UTC is assumed.	string datetime	No
Value	The value. Note that "." is always used as decimal separator for JSON messages. value is an optional field, but either the value or the message field must be present in a Value object. Valid values for action "set" together with property "Charge" are: • 1 (Charge On) • 0 (Charge Off) Valid values for action "get" together with the properties "Active Power", "Reactive Power", "Voltage", "Current", "Charging Point State" are: • 1 (Historical Value) • 0 (Realtime Value) We specify the start time and end time for which interval to get values from historical values.	float	Yes
Message	This may either be a status modifier for the value or a notification message from the device. Message is an optional field, but either the value or the message field must be present in a Value object.	string	Yes

Value objet model