



# Webservice specification

Document Revision	Date	Page
4	22/02/2018	1 / 43

# CarPark 3 API



# Webservice specification

Document Revision	Date	Page
4	22/02/2018	2 / 43

## 4 Index

4	Index .....	2
5	Introduction .....	4
6	Error handling .....	4
7	Webservice interface description .....	5
4	login .....	5
5	getGroups .....	5
8	.....	5
4	getGroupById.....	6
9	.....	6
4	getBays.....	7
10	.....	7
11	.....	7
4	getBaysById.....	7
12	.....	7
13	.....	8
4	getCategories.....	8
14	.....	8
4	getEventCategories .....	9
15	.....	9
4	getSequences.....	9
16	.....	9
4	getMZC.....	10
17	.....	10
18	.....	10
4	getIndicators .....	10
19	.....	10
20	.....	11
4	getSxweb .....	11
21	.....	11
4	getRtBaysStatus .....	12
22	.....	12
4	getRtBaysStatusById .....	12
23	.....	12
4	getRtBaysStatusByGroup .....	13
24	.....	13
25	.....	13



# Webservice specification

Document Revision	Date	Page
4	22/02/2018	3 / 43

- 4 getRtParkOccupancy..... 13
- 26..... 14
- 4 getRtGroupOccupancy..... 14
- 27..... 14
- 28..... 15
- 4 getRtCatParkOccupancy ..... 15
- 29..... 15
- 4 getRtCatGroupOccupancy ..... 15
- 30..... 15
- 4 getRtDiagnostic..... 16
- 31..... 16
- 4 getRtMZCOccupied ..... 16
- 32..... 17
- 4 getActiveAlarms..... 17
- 33..... 17
- 4 getAlarmsByTime..... 18
- 34..... 18
- 4 getParkOccupancy ..... 18
- 5 getGroupsOccupancy..... 19
- 6 changeBaysCategory..... 20
- 35..... 20
- 4 playSequence..... 20
- 36..... 20
- 4 setMzcValue..... 21
- 37..... 21
- 4 resetMzc ..... 22
- 38..... 22
- 4 changeIndicatorCategory..... 22
- 39..... 22
- 4 changeBaysEventCategory..... 23
- 40..... 23
- 4 acknowledgeAlarms..... 24
- 41..... 24
- 42 Data structures ..... 25
  - 42.4.1 Code errors ..... 29
- 43 WSDL reference ..... 32



# Webservice specification

Document Revision	Date	Page
4	22/02/2018	4 / 43

## 5 Introduction

This document describes the Car Park 3 Webservice.

The web service is available at the following URL:

<https://<ipaddress>/CP3WebInterface>

The protocol used to communicate between the CP3 and the user is based on SOAP over HTTP. The Webservice uses SOAP version 1.2 (<http://www.w3.org/TR/soap/>) with WSDL.

Used data structures are reported in chapter 7.

In order to use the webservices it is necessary to create an user account using CP-Y web interface enabling for this user webservice read/write permissions. In order to use the change category method it is useful enable more than one category.

The WSDL refers to the "localhost" address. The client software has to replace it with the real ip address.

## 6 Error handling

Errors occurring as a result of invalid processing of the request are returned in the response data. Unexpected errors which may occur as a result of any other error condition will be returned as SOAP faults with a message describing the error. The related HTTP status code will be 500 in all error cases.



# Webservice specification

Document Revision	Date	Page
4	22/02/2018	5 / 43

## 7 Webservice interface description

### 4 login

Authenticates the user with the system and obtains the token. All further methods use this token as parameter. After 30 minutes the token expires and it is necessary to login again.

#### Request

Method
<code>login</code>

Params	Type
username	string
password	string (MD5 encoded)

#### Response

Status	Response
Success	<code>token (string)</code>
Error	faultcode: 401; faultstring: "Invalid user"
Error	faultcode: 403; faultstring: "Unauthorized user"

## 5 getGroups

Returns the list of the configured groups.

## 8

#### Request

Method
<code>getGroups</code>

Params	Type
token	string



# Webservice specification

Document Revision	Date	Page
4	22/02/2018	6 / 43

## Response

Status	Response
Success	<code>groupList (GroupsList)</code>
Error	faultcode: 401; faultstring: "Invalid token"
Error	faultcode: 401; faultstring: "Expired token"
Error	faultcode: 403; faultstring: "Unauthorized user"

## 4 getGroupById

Returns the group with the given id.

## 9

### Request

Method
<code>getGroupById</code>

Params	Type
token	string
idgroup	int

### Response

Status	Response
Success	<code>group (Group)</code>
Error	faultcode: 401; faultstring: "Invalid token"
Error	faultcode: 401; faultstring: "Expired token"
Error	faultcode: 403; faultstring: "Unauthorized user"

## 4 getBays

Returns the list of the configured bays.

### 10

#### Request

Method
getBays

Params	Type
token	string

#### Response

Status	Response
Success	baysList ( <b>BaysList</b> )
Error	faultcode: 401; faultstring: "Invalid token"
Error	faultcode: 401; faultstring: "Expired token"
Error	faultcode: 403; faultstring: "Unauthorized user"

### 11

## 4 getBaysById

Returns the list of bays with the given ids.

### 12

#### Request

Method
getBaysById

Params	Type
token	string
idbays	IntList



# Webservice specification

Document Revision	Date	Page
4	22/02/2018	8 / 43

## Response

Status	Response
Success	baysList ( <b>BaysList</b> )
Error	faultcode: 401; faultstring: "Invalid token"
Error	faultcode: 401; faultstring: "Expired token"
Error	faultcode: 403; faultstring: "Unauthorized user"

13

## 4 getCategories

Returns the list of the configured bay categories.

14

## Request

Method
getCategories

Params	Type
token	string

## Response

Status	Response
Success	categoriesList ( <b>CategoriesList</b> )
Error	faultcode: 401; faultstring: "Invalid token"
Error	faultcode: 401; faultstring: "Expired token"
Error	faultcode: 403; faultstring: "Unauthorized user"



Document Revision	Date	Page
4	22/02/2018	9 / 43

#### 4 getEventCategories

Returns the list of the configured bay event categories.

### 15

#### Request

Method
<code>getEventCategories</code>

Params	Type
token	<code>string</code>

#### Response

Status	Response
<code>Success</code>	<code>categoriesList (CategoriesList)</code>
<code>Error</code>	faultcode: 401; faultstring: "Invalid token"
<code>Error</code>	faultcode: 401; faultstring: "Expired token"
<code>Error</code>	faultcode: 403; faultstring: "Unauthorized user"

#### 4 getSequences

Returns the list of the configured sequences.

### 16

#### Request

Method
<code>getSequences</code>

Params	Type
token	<code>string</code>

#### Response



# Webservice specification

Document Revision	Date	Page
4	22/02/2018	10 / 43

Status	Response
Success	<code>sequencesList</code> ( <code>SequencesList</code> )
Error	faultcode: 401; faultstring: "Invalid token"
Error	faultcode: 401; faultstring: "Expired token"
Error	faultcode: 403; faultstring: "Unauthorized user"

## 4 getMZC

Returns the list of the configured MZC.

17

### Request

Method
<code>getMZC</code>

Params	Type
token	<code>string</code>

### Response

Status	Response
Success	<code>MZCList</code> ( <code>MZCList</code> )
Error	faultcode: 401; faultstring: "Invalid token"
Error	faultcode: 401; faultstring: "Expired token"
Error	faultcode: 403; faultstring: "Unauthorized user"

18

## 4 getIndicators

Returns the list of the configured indicators.

19

### Request

Document Revision	Date	Page
4	22/02/2018	11 / 43

**Method**`getIndicators`

Params	Type
token	string

**Response**

Status	Response
Success	<code>indicatorsList</code> ( <code>IndicatorsList</code> )
Error	faultcode: 401; faultstring: "Invalid token"
Error	faultcode: 401; faultstring: "Expired token"
Error	faultcode: 403; faultstring: "Unauthorized user"

**20****4 getSxweb**

Returns the list of the configured sxweb.

**21****Request****Method**`getSxweb`

Params	Type
token	string

**Response**

Status	Response
Success	<code>sxwebList</code> ( <code>SxwebList</code> )
Error	faultcode: 401; faultstring: "Invalid token"



# Webservice specification

Document Revision	Date	Page
4	22/02/2018	12 / 43

<b>Error</b>	faultcode: 401; faultstring: "Expired token"
<b>Error</b>	faultcode: 403; faultstring: "Unauthorized user"

## 4 getRtBaysStatus

Returns the list of the realtime statuses of the configured bays.

### 22

#### Request

Method
<code>getRtBaysStatus</code>

Params	Type
token	string

#### Response

Status	Response
Success	<code>rtBaysStatusList</code> ( <code>RtBaysStatusList</code> )
Error	faultcode: 401; faultstring: "Invalid token"
Error	faultcode: 401; faultstring: "Expired token"
Error	faultcode: 403; faultstring: "Unauthorized user"

## 4 getRtBaysStatusById

Returns the list of the realtime statuses of the bays with the given ids.

### 23

#### Request

Method
<code>getRtBaysStatusById</code>

Params	Type
--------	------



# Webservice specification

Document Revision	Date	Page
4	22/02/2018	13 / 43

token	string
idbays	IntList

## Response

Status	Response
Success	rtBayStatus (RtBaysStatusList)
Error	faultcode: 401; faultstring: "Invalid token"
Error	faultcode: 401; faultstring: "Expired token"
Error	faultcode: 403; faultstring: "Unauthorized user"

## 4 getRtBaysStatusByGroup

Returns the list of the realtime statuses of the bays in the given group.

### 24

## Request

Method
getRtBaysStatusByGroup

Params	Type
token	string

## Response

Status	Response
Success	rtGroupBayStatus (RtBaysStatusList)
Error	faultcode: 401; faultstring: "Invalid token"
Error	faultcode: 401; faultstring: "Expired token"
Error	faultcode: 403; faultstring: "Unauthorized user"

### 25

## 4 getRtParkOccupancy

Returns the realtime occupancy data of the whole car park.

## 26

### Request

Method
<code>getRtParkOccupancy</code>

Params	Type
token	string

### Response

Status	Response
Success	<code>rtParkOccupancy</code> ( <code>RtGroupOccupancy</code> )
Error	faultcode: 401; faultstring: "Invalid token"
Error	faultcode: 401; faultstring: "Expired token"
Error	faultcode: 403; faultstring: "Unauthorized user"

## 4 getRtGroupOccupancy

Returns the realtime occupancy data of the specified groups.

## 27

### Request

Method
<code>getRtGroupOccupancy</code>

Params	Type
token	string
idgroups	IntList

### Response

Status	Response
Success	<code>rtGroupOccupancyList</code> ( <code>RtGroupOccupancyList</code> )
Error	faultcode: 401; faultstring: "Invalid token"



# Webservice specification

Document Revision	Date	Page
4	22/02/2018	15 / 43

Error	faultcode: 401; faultstring: "Expired token"
Error	faultcode: 403; faultstring: "Unauthorized user"

28

## 4 getRtCatParkOccupancy

Returns the realtime occupancy data of the whole carpark split by category.

29

### Request

Method
getRtCatParkOccupancy

Params	Type
token	string

### Response

Status	Response
Success	rtCatParkOccupancy (RtCatGroupOccupancy)
Error	faultcode: 401; faultstring: "Invalid token"
Error	faultcode: 401; faultstring: "Expired token"
Error	faultcode: 403; faultstring: "Unauthorized user"

## 4 getRtCatGroupOccupancy

Returns the realtime occupancy data of the specified groups split by category.

30

### Request

Method
getRtCatGroupOccupancy

Params	Type
--------	------



# Webservice specification

Document Revision	Date	Page
4	22/02/2018	16 / 43

token	string
idgroups	IntList

## Response

Status	Response
Success	rtCatGroupOccupancyList (RtCatGroupOccupancy)
Error	faultcode: 401; faultstring: "Invalid token"
Error	faultcode: 401; faultstring: "Expired token"
Error	faultcode: 403; faultstring: "Unauthorized user"

## 4 getRtDiagnostic

Returns the realtime diagnostic for all car park objects (bay, display, indicator, sxweb...).

31

## Request

Method
getRtDiagnostic

Params	Type
token	string

## Response

Status	Response
Success	rtDiagnosticList (RtDiagnostic)
Error	faultcode: 401; faultstring: "Invalid token"
Error	faultcode: 401; faultstring: "Expired token"
Error	faultcode: 403; faultstring: "Unauthorized user"

## 4 getRtMZCOccupied

Returns the realtime value of occupied spaces for all master zone counter areas in the park.





# Webservice specification

Document Revision	Date	Page
4	22/02/2018	17 / 43

## 32

### Request

Method
<code>getRtMZCOccupied</code>

Params	Type
token	string

### Response

Status	Response
Success	<code>rtMZCOccupiedList</code> ( <code>RtMZCOccupied</code> )
Error	faultcode: 401; faultstring: "Invalid token"
Error	faultcode: 401; faultstring: "Expired token"
Error	faultcode: 403; faultstring: "Unauthorized user"

## 4 getActiveAlarms

Returns all alarms still active in the park. An alarm is active when end time (alarm off) is not setted.

## 33

### Request

Method
<code>getActiveAlarms</code>

Params	Type
token	string

### Response

Status	Response
Success	<code>activeAlarmsList</code> ( <code>AlarmsList</code> )
Error	faultcode: 401; faultstring: "Invalid token"
Error	faultcode: 401; faultstring: "Expired token"



# Webservice specification

Document Revision	Date	Page
4	22/02/2018	18 / 43

<b>Error</b>	faultcode: 403; faultstring: "Unauthorized user"
--------------	--

## 4 getAlarmsByTime

Returns all alarms in the park between two timestamp (startTime – endTime ). Alarms could be active or inactive

34

### Request

Method
getAlarmsByTime

Params	Type
token	string
startTime	timestamp (UTC)
endTime	timestamp (UTC)

### Response

Status	Response
Success	alarmsList (AlarmsList)
Error	faultcode: 401; faultstring: "Invalid token"
Error	faultcode: 401; faultstring: "Expired token"
Error	faultcode: 403; faultstring: "Unauthorized user"

## 4 getParkOccupancy

Returns occupancy status of the park (number of vacant/occupied spaces split by category, errors) every 15 minutes between two timestamp (startTime – endTime ).

### Request

Method
getParkOccupancy

Params	Type
--------	------



# Webservice specification

Document Revision	Date	Page
4	22/02/2018	19 / 43

token	string
startTime	timestamp (UTC)
endTime	timestamp (UTC)

## Response

Status	Response
Success	parkOccupancyList (GroupOccupancyList)
Error	faultcode: 401; faultstring: "Invalid token"
Error	faultcode: 401; faultstring: "Expired token"
Error	faultcode: 403; faultstring: "Unauthorized user"

## 5 getGroupsOccupancy

Returns occupancy status of a group or a list of groups (number of vacant/occupied spaces split by category, errors) every 15 minutes between two timestamp (startTime – endTime).

### Request

Method
getGroupsOccupancy

Params	Type
token	string
startTime	timestamp (UTC)
endTime	timestamp (UTC)
idgroups	IntList

### Response

Status	Response
Success	parkOccupancyList (GroupOccupancyList)
Error	faultcode: 401; faultstring: "Invalid token"
Error	faultcode: 401; faultstring: "Expired token"
Error	faultcode: 403; faultstring: "Unauthorized user"



# Webservice specification

Document Revision	Date	Page
4	22/02/2018	20 / 43

## 6 changeBaysCategory

Changes the category of the specified bays.

35

### Request

Method
<code>changeBaysCategory</code>

Params	Type
token	string
idcategory	int
idbays	IntList

### Response

Status	Response
Success	<code>chCatResp</code> ( <code>changeCategoryResponse</code> )
Error	faultcode: 401; faultstring: "Invalid token"
Error	faultcode: 401; faultstring: "Expired token"
Error	faultcode: 403; faultstring: "Unauthorized user"
Error	faultcode: 410; faultstring: "Disabled category"
Error	faultcode: 411; faultstring: "Invalid category"

## 4 playSequence

Play a sequence

36

### Request

Method
<code>playSequence</code>



# Webservice specification

Document Revision	Date	Page
4	22/02/2018	21 / 43

Params	Type
token	string
idsequence	int

## Response

Status	Response
Success	chCatResp (changeCategoryResponse)
Error	faultcode: 401; faultstring: "Invalid token"
Error	faultcode: 401; faultstring: "Expired token"
Error	faultcode: 403; faultstring: "Unauthorized user"
Error	faultcode: 411; faultstring: "Invalid sequence"

## 4 setMzcValue

Set the number of vacant spaces for a master zone counter

37

## Request

Method
setMzcValue

Params	Type
token	string
idmzc	int
value	int

## Response

Status	Response
Success	result (int)
Error	faultcode: 401; faultstring: "Invalid token"



# Webservice specification

Document Revision	Date	Page
4	22/02/2018	22 / 43

Error	faultcode: 401; faultstring: "Expired token"
Error	faultcode: 403; faultstring: "Unauthorized user"
Error	faultcode: 410; faultstring: "Exceeded mzc max value"
Error	faultcode: 411; faultstring: "Invalid mzc"

## 4 resetMzc

Reset master zone counter: set vacant spaces to maximum zone spaces.

38

### Request

Method
resetMzc

Params	Type
token	string
idmzc	int

### Response

Status	Response
Success	result (int)
Error	faultcode: 401; faultstring: "Invalid token"
Error	faultcode: 401; faultstring: "Expired token"
Error	faultcode: 403; faultstring: "Unauthorized user"
Error	faultcode: 411; faultstring: "Invalid mzc"

## 4 changeIndicatorCategory

Change category for an indicator and all other indicators and bay linked to that indicator

39

### Request

Method
--------



# Webservice specification

Document Revision	Date	Page
4	22/02/2018	23 / 43

changeIndicatorCategory

Params	Type
token	string
idcategory	int
idindicator	int

## Response

Status	Response
Success	chIndCatResp (changeCatIndResponse)
Error	faultcode: 401; faultstring: "Invalid token"
Error	faultcode: 401; faultstring: "Expired token"
Error	faultcode: 403; faultstring: "Unauthorized user"
Error	faultcode: 410; faultstring: "Disabled category"
Error	faultcode: 411; faultstring: "Invalid category"

## 4 changeBaysEventCategory

Changes the event category of the specified bays.

40

## Request

Method
changeBaysEventCategory

Params	Type
token	string
idcategory	int
idbays	IntList

## Response



# Webservice specification

Document Revision	Date	Page
4	22/02/2018	24 / 43

Status	Response
Success	chEvCatResp ( <b>changeCategoryResponse</b> )
Error	faultcode: 401; faultstring: "Invalid token"
Error	faultcode: 401; faultstring: "Expired token"
Error	faultcode: 403; faultstring: "Unauthorized user"
Error	faultcode: 410; faultstring: "Disabled event category"
Error	faultcode: 411; faultstring: "Invalid event category"

## 4 acknowledgeAlarms

Acknowledge an alarms

41

### Request

Method
acknowledge

Params	Type
token	string
idalarms	IntList

### Response

Status	Response
Success	result ( <b>int</b> )
Error	faultcode: 401; faultstring: "Invalid token"
Error	faultcode: 401; faultstring: "Expired token"
Error	faultcode: 403; faultstring: "Unauthorized user"





# Webservice specification

Document Revision	Date	Page
4	22/02/2018	25 / 43

## 42 Data structures

**GroupsList:** *array of Group*

**Group:** *object*

Describes an arbitrary aggregation of any of the following elements:

- bays
- real groups (Lines, Lanes, Zones, etc.)

Fields:

<b>pupdate</b>	<i>int</i>	UTC timestamp of the last configuration update
<b>idgroup</b>	<i>int</i>	group id
<b>name</b>	<i>string</i>	group name
<b>idparent</b>	<i>int</i>	parent group id
<b>type</b>	<i>int</i>	group type (0: normal, 1: line, 2: lane, 3: indicator, 255: virtual)
<b>idzonecounter</b>	<i>int</i>	master zone counter (MZC) id

**BaysList:** *array of Bay*

**Bay:** *object*

Describes a carpark bay.

Fields:

<b>pupdate</b>	<i>int</i>	UTC timestamp of the last configuration update
<b>idbay</b>	<i>int</i>	bay id
<b>name</b>	<i>string</i>	bay name
<b>idcategory</b>	<i>int</i>	bay category id
<b>ideventcategory</b>	<i>int</i>	bay event category id
<b>idsxweb</b>	<i>int</i>	id of the SxWEB that owns the bay
<b>lanenum</b>	<i>int</i>	lane number (L)
<b>linenum</b>	<i>int</i>	line number (L)
<b>positionnum</b>	<i>int</i>	position number (P)

**CategoriesList:** *array of Category*

**Category:** *object*

Describes a bay category.

Fields:

<b>pupdate</b>	<i>int</i>	UTC timestamp of the last configuration update
<b>idcategory</b>	<i>int</i>	category id (1 - 7)
<b>name</b>	<i>string</i>	category name
<b>color</b>	<i>string</i>	category color (RGB code)
<b>blink</b>	<i>int</i>	blink type (0: off, 1: active 0.5s, 2: active 1s, 3: active 2s)
<b>enable</b>	<i>int</i>	category enable (0: disabled, 1: enabled)

**SequencesList:** *array of Sequences*

**Sequences:** *object*

Describes a carpark sequence.

Fields:



# Webservice specification

Document Revision	Date	Page
4	22/02/2018	26 / 43

<b>idsequence</b>	<i>int</i>	sequences id
<b>idsxweb</b>	<i>int</i>	<i>id of sxweb</i> where sequence is configured
<b>idgroup</b>	<i>int</i>	<i>id of group</i> where sequence is located
<b>name</b>	<i>string</i>	sequences name

**MZCList:** *array of MZC*

**MZC:** *object*

Describes a carpark master zone counter.

Fields:

<b>idmzc</b>	<i>int</i>	sequences id
<b>idsxweb</b>	<i>int</i>	<i>id of sxweb</i> where MZC is configured
<b>name</b>	<i>string</i>	sequence name
<b>totalspaces</b>	<i>int</i>	number of total spaces

**IndicatorsList:** *array of Indicator*

**Indicator:** *object*

Describes a carpark master zone counter.

Fields:

<b>pupdate</b>	<i>int</i>	UTC timestamp of the last status configuration update
<b>idindicator</b>	<i>int</i>	indicator id
<b>idsxweb</b>	<i>int</i>	<i>id of sxweb</i> where MZC is configured
<b>name</b>	<i>string</i>	sequence name
<b>idgroup</b>	<i>int</i>	id of groups where indicator is located
<b>idcategory</b>	<i>int</i>	indicator category id
<b>valuesstatus</b>	<i>int</i>	status of bays that are linked to indicator (in this case is always vacant [1])

**SxwebList:** *array of Sxweb*

**Sxweb:** *object*

Describes a carpark master zone counter.

Fields:

<b>idsxweb</b>	<i>int</i>	<i>id of sxweb</i>
<b>name</b>	<i>string</i>	<i>sxweb</i> name
<b>mac</b>	<i>string</i>	<i>sxweb</i> mac address
<b>ipaddress</b>	<i>string</i>	<i>sxweb</i> ip address
<b>port</b>	<i>string</i>	<i>sxweb</i> port

**RtBaysStatusList:** *array of RtBayStatus*

**RtBayStatus:** *object*

Describes the realtime status of a carpark bay.

Fields:

<b>idbay</b>	<i>int</i>	bay id
<b>timestamp</b>	<i>int</i>	UTC timestamp of the last realtime status update
<b>status</b>	<i>int</i>	realtime bay status (0: occupied, 1: vacant, -2: error)
<b>idcategory</b>	<i>int</i>	bay category id

**RtGroupOccupancyList:** *array of RtGroupOccupancy*



# Webservice specification

Document Revision	Date	Page
4	22/02/2018	27 / 43

## **RtGroupOccupancy:** *object*

Describes the realtime group occupancy.

Fields:

<b>idgroup</b>	<i>int</i>	group id
<b>vacant</b>	<i>int</i>	vacant bays number
<b>occupied</b>	<i>int</i>	occupied bays number
<b>error</b>	<i>int</i>	number of bays in error state

## **RtCatGroupOccupancyList:** *array of RtCatGroupOccupancy*

### **RtCatGroupOccupancy:** *object*

Describes the realtime group occupancy split by category.

Fields:

<b>idgroup</b>	<i>int</i>	group id
<b>c1vacant</b>	<i>int</i>	category 1 vacant bays number
<b>c1occupied</b>	<i>int</i>	category 1 occupied bays number
<b>c1error</b>	<i>int</i>	category 1 number of bays in error state
<b>c2vacant</b>	<i>int</i>	category 2 vacant bays number
<b>c2occupied</b>	<i>int</i>	category 2 occupied bays number
<b>c2error</b>	<i>int</i>	category 2 number of bays in error state
<b>c3vacant</b>	<i>int</i>	category 3 vacant bays number
<b>c3occupied</b>	<i>int</i>	category 3 occupied bays number
<b>c3error</b>	<i>int</i>	category 3 number of bays in error state
<b>c4vacant</b>	<i>int</i>	category 4 vacant bays number
<b>c4occupied</b>	<i>int</i>	category 4 occupied bays number
<b>c4error</b>	<i>int</i>	category 4 number of bays in error state
<b>c5vacant</b>	<i>int</i>	category 5 vacant bays number
<b>c5occupied</b>	<i>int</i>	category 5 occupied bays number
<b>c5error</b>	<i>int</i>	category 5 number of bays in error state
<b>c6vacant</b>	<i>int</i>	category 6 vacant bays number
<b>c6occupied</b>	<i>int</i>	category 6 occupied bays number
<b>c6error</b>	<i>int</i>	category 6 number of bays in error state
<b>c7vacant</b>	<i>int</i>	category 7 vacant bays number
<b>c7occupied</b>	<i>int</i>	category 7 occupied bays number
<b>c7error</b>	<i>int</i>	category 7 number of bays in error state

## **RtDiagnosticList** *array of RtDiagnostic*

### **RtDiagnostic:** *object*

Describes the realtime diagnostic for all objects in the park.

Fields:

<b>pupdate</b>	<i>int</i>	UTC timestamp of the last status configuration update
<b>id</b>	<i>int</i>	row id
<b>source</b>	<i>int</i>	object enumeration
<b>idobj</b>	<i>int</i>	object id
<b>timestamp</b>	<i>int</i>	UTC timestamp of the last realtime status update
<b>status</b>	<i>int</i>	realtime bay status (0: no diagnostic , -1: diagnostic)
<b>code</b>	<i>bitfield</i>	diagnostic bitfield



# Webservice specification

Document Revision	Date	Page
4	22/02/2018	28 / 43

**RtMZCOccupiedList:** *array of RtMZCOccupied*

**RtMZCOccupied:** *object*

Describes the realtime status of a carpark master zone counter.

Fields:

<b>pupdate</b>	<i>int</i>	UTC timestamp of the last status configuration update
<b>idmzc</b>	<i>int</i>	master zone counter id
<b>timestamp</b>	<i>int</i>	UTC timestamp of the last realtime status update
<b>occupied</b>	<i>int</i>	number of occupied mzc spaces
<b>total</b>	<i>int</i>	number of mzc total spaces

**AlarmsList:** *array of Alarm*

**Alarm:** *object*

Describes the realtime status of a carpark master zone counter.

Fields:

<b>id</b>	<i>int</i>	<i>alarm id</i>
<b>source</b>	<i>int</i>	type of object (1: sxweb, 2: bay, 3: display, 6: indicator, 101: bay alarm, 102: group alarm)
<b>idobj</b>	<i>int</i>	id object
<b>alarmon</b>	<i>int</i>	UTC timestamp when alarm open
<b>alarmoff</b>	<i>int</i>	UTC timestamp when alarm close
<b>code</b>	<i>bitfield</i>	alarm code ( <b>code errors</b> )
<b>ack</b>	<i>int</i>	UTC timestamp acknowledgement
<b>iduser</b>	<i>int</i>	id of user that acknowledge alarm
<b>username</b>	<i>string</i>	name of user that acknowledge alarm

**GroupOccupancyList:** *array of GroupOccupancy*

**GroupOccupancy:** *object*

Describes the realtime group occupancy split by category.

Fields:

<b>timestamp</b>	<i>int</i>	UTC timestamp when sample is saved
<b>idgroup</b>	<i>int</i>	group id
<b>c1vacant</b>	<i>int</i>	category 1 vacant bays number
<b>c1occupied</b>	<i>int</i>	category 1 occupied bays number
<b>c2vacant</b>	<i>int</i>	category 2 vacant bays number
<b>c2occupied</b>	<i>int</i>	category 2 occupied bays number
<b>c3vacant</b>	<i>int</i>	category 3 vacant bays number
<b>c3occupied</b>	<i>int</i>	category 3 occupied bays number
<b>c4vacant</b>	<i>int</i>	category 4 vacant bays number
<b>c4occupied</b>	<i>int</i>	category 4 occupied bays number
<b>c5vacant</b>	<i>int</i>	category 5 vacant bays number
<b>c5occupied</b>	<i>int</i>	category 5 occupied bays number
<b>c6vacant</b>	<i>int</i>	category 6 vacant bays number
<b>c6occupied</b>	<i>int</i>	category 6 occupied bays number
<b>c7vacant</b>	<i>int</i>	category 7 vacant bays number
<b>c7occupied</b>	<i>int</i>	category 7 occupied bays number
<b>errors</b>	<i>int</i>	number of bays in error state



# Webservice specification

Document Revision	Date	Page
4	22/02/2018	29 / 43

## **changeCategoryResponse:** *object*

Response to a changeBaysCategory request.

Fields:

- result** *int* return value (0: ok, -1: internal application error, -2: sbweb connection error)
- bays** *IntList* list of bays on which the category change has succeeded
- errorBays** *IntList* list of bays on which the category change has failed

## **changeCatIndResponse:** *object*

Response to a changeIndicatorCategory request.

Fields:

- result** *int* return value (0: ok, -1: internal application error, -2: sbweb connection error)
- indicator** *IntList* list of indicators on which the category change has succeeded
- errorIndicator** *IntList* list of indicators on which the category change has failed

## **IntList:** *array of int*

### **RGB code:**

- Red: "#FF0000"
- Green: "#00FF00"
- Blue: "#0000FF"
- Orange: "#FF9900"
- Pink: "#FF66FF"
- Yellow: "#FFFF00"
- Cyan: "#00CCFF"
- White: "#FFFFFF"
- Purple: "#660066"
- OFF: "#000000"

## 42.4.1 Code errors

Error message	Bit	Number Value
shortcircuit_hw_protection	0	1
shortcircuit_sw_protection	1	2
load_not_present	2	4



# Webservice specification

Document Revision	Date	Page
4	22/02/2018	30 / 43

dali_lamp_failure	3	8
dali_network_shorted	4	16
critical_current_overload	5	32
inverted_dupline_modules	6	64
battery_low	7	128
battery_not_present	8	256
sensor_dirty	9	512
hardware_error	10	1024
presence	11	2048
voltage	12	4096
quality_index	13	8192
wrong_connection	14	16384
short_circuit	15	32768
overload_warning	16	65536
overload_protection	17	131072
peak_over_supply	18	262144
voltage_bus_repeated	19	524288
current_bus_repeated	20	1048576
pd_voltage_threshold	21	2097152
over_temperature	22	4194304
over_voltage	23	8388608
over_peak	24	16777216
battery_level	25	33554432
configuration_ok	26	67108864
current	27	134217728
error_counter	28	268435456
pull_down_level	29	536870912
quality_index_submaster	30	1073741824
errors	31	2147483648
null_cp_address	32	4294967296
startup	33	8589934592
cross_talk	34	17179869184
unstable	35	34359738368
pow_voltage_drop	36	68719476736
d_voltage_drop	37	137438953472



# Webservice specification

Document Revision	Date	Page
4	22/02/2018	31 / 43

base_holder_fault	38	274877906944
local_button_fault	39	549755813888
calibration_error	40	1099511627776
remote_calibration	41	2199023255552
local_calibration	42	4398046511104
dali_ballast_no_answer	43	8796093022208
display_disconnected	44	17592186044416
display_not_correct	45	35184372088832
module_power_up	46	70368744177664
sensor_error	47	140737488355328
NOT USED	50	-
NOT USED	51	-
NOT USED	52	-
NOT USED	53	-
NOT USED	54	-
NOT USED	55	-
NOT USED	56	-
NOT USED	57	-
NOT USED	58	-
NOT USED	59	-
NOT USED	60	-
NOT USED	61	-
NOT USED	62	-
module_not_connected	63	9223372036854775808

Document Revision	Date	Page
4	22/02/2018	32 / 43

## 43 WSDL reference

The WSDL reference is available by connecting to the CPY webserver interface at the following IP address:

<http://ipaddress/CP3Service/public/CP3WebInterface.wsdl>

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://www.example.org/service/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="CP3WebInterface" targetNamespace="http://www.example.org/service/">

  <wsdl:types>
    <xsd:schema elementFormDefault="qualified" targetNamespace="http://www.example.org/service/">

      <xsd:complexType name="IntList">
        <xsd:sequence>
          <xsd:element minOccurs="0" maxOccurs="unbounded" name="List" type="xsd:int"/>
        </xsd:sequence>
      </xsd:complexType>

      <xsd:complexType name="Group">
        <xsd:all>
          <xsd:element name="pupdate" type="xsd:int"/>
          <xsd:element name="idgroup" type="xsd:int"/>
          <xsd:element name="name" type="xsd:string"/>
          <xsd:element name="idparent" type="xsd:int"/>
          <xsd:element name="type" type="xsd:int"/>
          <xsd:element name="idzonecounter" type="xsd:int"/>
        </xsd:all>
      </xsd:complexType>

      <xsd:complexType name="GroupsList">
        <xsd:sequence>
          <xsd:element minOccurs="0" maxOccurs="unbounded" name="Group" nillable="true" type="tns:Group"/>
        </xsd:sequence>
      </xsd:complexType>

      <xsd:complexType name="Bay">
        <xsd:all>
          <xsd:element name="pupdate" type="xsd:int"/>
          <xsd:element name="idbay" type="xsd:int"/>
          <xsd:element name="name" type="xsd:string"/>
          <xsd:element name="type" type="xsd:int"/>
          <xsd:element name="idcategory" type="xsd:int"/>
          <xsd:element name="ideventcategory" type="xsd:int"/>
          <xsd:element name="idsxweb" type="xsd:int"/>
          <xsd:element name="lanenum" type="xsd:int"/>
          <xsd:element name="linenum" type="xsd:int"/>
          <xsd:element name="positionnum" type="xsd:int"/>
        </xsd:all>
      </xsd:complexType>

      <xsd:complexType name="BaysList">
        <xsd:sequence>
          <xsd:element minOccurs="0" maxOccurs="unbounded" name="Bay" nillable="true" type="tns:Bay"/>
        </xsd:sequence>
      </xsd:complexType>

      <xsd:complexType name="Category">
        <xsd:all>
          <xsd:element name="pupdate" type="xsd:int"/>
          <xsd:element name="idcategory" type="xsd:int"/>
          <xsd:element name="name" type="xsd:string"/>
          <xsd:element name="color" type="xsd:string"/>
          <xsd:element name="blink" type="xsd:int"/>
          <xsd:element name="enable" type="xsd:int"/>
        </xsd:all>
      </xsd:complexType>

      <xsd:complexType name="CategoriesList">
        <xsd:sequence>
          <xsd:element minOccurs="0" maxOccurs="unbounded" name="Category" nillable="true" type="tns:Category"/>
        </xsd:sequence>
      </xsd:complexType>

      <xsd:complexType name="Sequence">
        <xsd:all>
          <xsd:element name="idsequence" type="xsd:int"/>
          <xsd:element name="idsxweb" type="xsd:string"/>
          <xsd:element name="idgroup" type="xsd:string"/>
          <xsd:element name="name" type="xsd:string"/>
        </xsd:all>
      </xsd:complexType>
    </xsd:schema>
  </wsdl:types>

```



```

</xsd:all>
</xsd:complexType>
<xsd:complexType name="SequencesList">
  <xsd:sequence>
    <xsd:element minOccurs="0" maxOccurs="unbounded" name="Sequence" nillable="true" type="tns:Sequence"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="MZC">
  <xsd:all>
    <xsd:element name="idmzc" type="xsd:int"/>
    <xsd:element name="idsxweb" type="xsd:int"/>
    <xsd:element name="name" type="xsd:string"/>
    <xsd:element name="totalspaces" type="xsd:int"/>
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="MZCList">
  <xsd:sequence>
    <xsd:element minOccurs="0" maxOccurs="unbounded" name="MZC" nillable="true" type="tns:MZC"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Indicator">
  <xsd:all>
    <xsd:element name="pupdate" type="xsd:int"/>
    <xsd:element name="idindicator" type="xsd:int"/>
    <xsd:element name="idsxweb" type="xsd:int"/>
    <xsd:element name="name" type="xsd:string"/>
    <xsd:element name="idgroup" type="xsd:int"/>
    <xsd:element name="idcategory" type="xsd:int"/>
    <xsd:element name="valuestatus" type="xsd:int"/>
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="IndicatorsList">
  <xsd:sequence>
    <xsd:element minOccurs="0" maxOccurs="unbounded" name="Indicator" nillable="true" type="tns:Indicator"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Sxweb">
  <xsd:all>
    <xsd:element name="idsxweb" type="xsd:int"/>
    <xsd:element name="name" type="xsd:string"/>
    <xsd:element name="mac" type="xsd:string"/>
    <xsd:element name="ipaddress" type="xsd:string"/>
    <xsd:element name="port" type="xsd:string"/>
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="SxwebList">
  <xsd:sequence>
    <xsd:element minOccurs="0" maxOccurs="unbounded" name="Sxweb" nillable="true" type="tns:Sxweb"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="RtBayStatus">
  <xsd:all>
    <xsd:element name="idbay" type="xsd:int"/>
    <xsd:element name="timestamp" type="xsd:int"/>
    <xsd:element name="status" type="xsd:int"/>
    <xsd:element name="idcategory" type="xsd:int"/>
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="RtBaysStatusList">
  <xsd:sequence>
    <xsd:element minOccurs="0" maxOccurs="unbounded" name="RtBayStatus" nillable="true" type="tns:RtBayStatus"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="RtGroupOccupancyList">
  <xsd:sequence>
    <xsd:element minOccurs="0" maxOccurs="unbounded" name="RtGroupOccupancy" nillable="true" type="tns:RtGroupOccupancy"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="RtGroupOccupancy">
  <xsd:all>
    <xsd:element name="idgroup" type="xsd:int"/>
    <xsd:element name="vacant" type="xsd:int"/>
    <xsd:element name="occupied" type="xsd:int"/>
    <xsd:element name="error" type="xsd:int"/>
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="RtCatGroupOccupancy">
  <xsd:all>

```

```

<xsd:element name="idgroup" type="xsd:int"/>
<xsd:element name="c1vacant" type="xsd:int"/>
<xsd:element name="c1occupied" type="xsd:int"/>
<xsd:element name="c1error" type="xsd:int"/>
<xsd:element name="c2vacant" type="xsd:int"/>
<xsd:element name="c2occupied" type="xsd:int"/>
<xsd:element name="c2error" type="xsd:int"/>
<xsd:element name="c3vacant" type="xsd:int"/>
<xsd:element name="c3occupied" type="xsd:int"/>
<xsd:element name="c3error" type="xsd:int"/>
<xsd:element name="c4vacant" type="xsd:int"/>
<xsd:element name="c4occupied" type="xsd:int"/>
<xsd:element name="c4error" type="xsd:int"/>
<xsd:element name="c5vacant" type="xsd:int"/>
<xsd:element name="c5occupied" type="xsd:int"/>
<xsd:element name="c5error" type="xsd:int"/>
<xsd:element name="c6vacant" type="xsd:int"/>
<xsd:element name="c6occupied" type="xsd:int"/>
<xsd:element name="c6error" type="xsd:int"/>
<xsd:element name="c7vacant" type="xsd:int"/>
<xsd:element name="c7occupied" type="xsd:int"/>
<xsd:element name="c7error" type="xsd:int"/>
</xsd:all>
</xsd:complexType>
<xsd:complexType name="RtCatGroupOccupancyList">
  <xsd:sequence>
    <xsd:element minOccurs="0" maxOccurs="unbounded" name="RtCatGroupOccupancy" nillable="true"
      type="tns:RtCatGroupOccupancy"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="RtDiagnostic">
  <xsd:all>
    <xsd:element name="pupdate" type="xsd:int"/>
    <xsd:element name="id" type="xsd:int"/>
    <xsd:element name="source" type="xsd:int"/>
    <xsd:element name="idobj" type="xsd:int"/>
    <xsd:element name="timestamp" type="xsd:int"/>
    <xsd:element name="status" type="xsd:int"/>
    <xsd:element name="code" type="xsd:int"/>
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="RtDiagnosticList">
  <xsd:sequence>
    <xsd:element minOccurs="0" maxOccurs="unbounded" name="RtDiagnostic" nillable="true" type="tns:RtDiagnostic"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="RtMZCOccupied">
  <xsd:all>
    <xsd:element name="pupdate" type="xsd:int"/>
    <xsd:element name="idmzc" type="xsd:int"/>
    <xsd:element name="timestamp" type="xsd:int"/>
    <xsd:element name="occupied" type="xsd:int"/>
    <xsd:element name="total" type="xsd:int"/>
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="RtMZCOccupiedList">
  <xsd:sequence>
    <xsd:element minOccurs="0" maxOccurs="unbounded" name="RtMZCOccupied" nillable="true"
      type="tns:RtMZCOccupied"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Alarm">
  <xsd:all>
    <xsd:element name="id" type="xsd:int"/>
    <xsd:element name="source" type="xsd:int"/>
    <xsd:element name="idobj" type="xsd:int"/>
    <xsd:element name="alarmon" type="xsd:int"/>
    <xsd:element name="alarmoff" type="xsd:int"/>
    <xsd:element name="code" type="xsd:int"/>
    <xsd:element name="ack" type="xsd:int"/>
    <xsd:element name="iduser" type="xsd:int"/>
    <xsd:element name="username" type="xsd:string"/>
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="AlarmsList">
  <xsd:sequence>
    <xsd:element minOccurs="0" maxOccurs="unbounded" name="Alarm" nillable="true" type="tns:Alarm"/>
  </xsd:sequence>
</xsd:complexType>

```

Document Revision	Date	Page
4	22/02/2018	35 / 43

```

<xsd:complexType name="GroupOccupancy">
  <xsd:all>
    <xsd:element name="timestamp" type="xsd:int"/>
    <xsd:element name="idgroup" type="xsd:int"/>
    <xsd:element name="c1vacant" type="xsd:int"/>
    <xsd:element name="c1occupied" type="xsd:int"/>
    <xsd:element name="c2vacant" type="xsd:int"/>
    <xsd:element name="c2occupied" type="xsd:int"/>
    <xsd:element name="c3vacant" type="xsd:int"/>
    <xsd:element name="c3occupied" type="xsd:int"/>
    <xsd:element name="c4vacant" type="xsd:int"/>
    <xsd:element name="c4occupied" type="xsd:int"/>
    <xsd:element name="c5vacant" type="xsd:int"/>
    <xsd:element name="c5occupied" type="xsd:int"/>
    <xsd:element name="c6vacant" type="xsd:int"/>
    <xsd:element name="c6occupied" type="xsd:int"/>
    <xsd:element name="c7vacant" type="xsd:int"/>
    <xsd:element name="c7occupied" type="xsd:int"/>
    <xsd:element name="errors" type="xsd:int"/>
  </xsd:all>
</xsd:complexType>

<xsd:complexType name="GroupOccupancyList">
  <xsd:sequence>
    <xsd:element minOccurs="0" maxOccurs="unbounded" name="GroupOccupancy" nillable="true"
type="tns:GroupOccupancy"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="changeCategoryResponse">
  <xsd:all>
    <xsd:element name="result" type="xsd:int"/>
    <xsd:element name="bays" type="tns:IntList"/>
    <xsd:element name="errorBays" type="tns:IntList"/>
  </xsd:all>
</xsd:complexType>

<xsd:complexType name="changeCatIndResponse">
  <xsd:all>
    <xsd:element name="result" type="xsd:int"/>
    <xsd:element name="indicator" type="tns:IntList"/>
    <xsd:element name="errorIndicator" type="tns:IntList"/>
  </xsd:all>
</xsd:complexType>

</xsd:schema>
</wsdl:types>

<wsdl:message name="loginRequest">
  <wsdl:part name="username" type="xsd:string"/>
  <wsdl:part name="password" type="xsd:string"/>
</wsdl:message>
<wsdl:message name="loginResponse">
  <wsdl:part name="token" type="xsd:string"/>
</wsdl:message>

<wsdl:message name="getGroupsRequest">
  <wsdl:part name="token" type="xsd:string"/>
</wsdl:message>
<wsdl:message name="getGroupsResponse">
  <wsdl:part name="groupList" type="tns:GroupsList"/>
</wsdl:message>
<wsdl:message name="getGroupByldRequest">
  <wsdl:part name="token" type="xsd:string"/>
  <wsdl:part name="idgroup" type="xsd:int"/>
</wsdl:message>
<wsdl:message name="getGroupByldResponse">
  <wsdl:part name="group" type="tns:Group"/>
</wsdl:message>

<wsdl:message name="getBaysRequest">
  <wsdl:part name="token" type="xsd:string"/>
</wsdl:message>
<wsdl:message name="getBaysResponse">
  <wsdl:part name="baysList" type="tns:BaysList"/>
</wsdl:message>
<wsdl:message name="getBaysByldRequest">
  <wsdl:part name="token" type="xsd:string"/>
  <wsdl:part name="idbays" type="tns:IntList"/>
</wsdl:message>
<wsdl:message name="getBaysByldResponse">
  <wsdl:part name="baysList" type="tns:BaysList"/>
</wsdl:message>

```

Document Revision	Date	Page
4	22/02/2018	36 / 43

```

<wsdl:message name="getCategoriesRequest">
  <wsdl:part name="token" type="xsd:string"/>
</wsdl:message>
<wsdl:message name="getCategoriesResponse">
  <wsdl:part name="categoriesList" type="tns:CategoriesList"/>
</wsdl:message>
<wsdl:message name="getEventCategoriesRequest">
  <wsdl:part name="token" type="xsd:string"/>
</wsdl:message>
<wsdl:message name="getEventCategoriesResponse">
  <wsdl:part name="categoriesList" type="tns:CategoriesList"/>
</wsdl:message>

<wsdl:message name="getSequencesRequest">
  <wsdl:part name="token" type="xsd:string"/>
</wsdl:message>
<wsdl:message name="getSequencesResponse">
  <wsdl:part name="sequencesList" type="tns:SequencesList"/>
</wsdl:message>
<wsdl:message name="getMZCRequest">
  <wsdl:part name="token" type="xsd:string"/>
</wsdl:message>
<wsdl:message name="getMZCResponse">
  <wsdl:part name="mzclList" type="tns:MZCList"/>
</wsdl:message>
<wsdl:message name="getIndicatorsRequest">
  <wsdl:part name="token" type="xsd:string"/>
</wsdl:message>
<wsdl:message name="getIndicatorsResponse">
  <wsdl:part name="indicatorsList" type="tns:IndicatorsList"/>
</wsdl:message>
<wsdl:message name="getSxwebRequest">
  <wsdl:part name="token" type="xsd:string"/>
</wsdl:message>
<wsdl:message name="getSxwebResponse">
  <wsdl:part name="sxwebList" type="tns:SxwebList"/>
</wsdl:message>

<wsdl:message name="getRtBaysStatusRequest">
  <wsdl:part name="token" type="xsd:string"/>
</wsdl:message>
<wsdl:message name="getRtBaysStatusResponse">
  <wsdl:part name="rtBaysStatusList" type="tns:RtBaysStatusList"/>
</wsdl:message>
<wsdl:message name="getRtBaysStatusByldRequest">
  <wsdl:part name="token" type="xsd:string"/>
  <wsdl:part name="idbays" type="tns:IntList"/>
</wsdl:message>
<wsdl:message name="getRtBaysStatusByldResponse">
  <wsdl:part name="rtBayStatus" type="tns:RtBaysStatusList"/>
</wsdl:message>
<wsdl:message name="getRtBaysStatusByGroupRequest">
  <wsdl:part name="token" type="xsd:string"/>
  <wsdl:part name="idgroup" type="xsd:int"/>
</wsdl:message>
<wsdl:message name="getRtBaysStatusByGroupResponse">
  <wsdl:part name="rtGroupBaysStatus" type="tns:RtBaysStatusList"/>
</wsdl:message>

<wsdl:message name="getRtParkOccupancyRequest">
  <wsdl:part name="token" type="xsd:string"/>
</wsdl:message>
<wsdl:message name="getRtParkOccupancyResponse">
  <wsdl:part name="rtParkOccupancy" type="tns:RtGroupOccupancy"/>
</wsdl:message>
<wsdl:message name="getRtGroupOccupancyRequest">
  <wsdl:part name="token" type="xsd:string"/>
  <wsdl:part name="idgroups" type="tns:IntList"/>
</wsdl:message>
<wsdl:message name="getRtGroupOccupancyResponse">
  <wsdl:part name="rtGroupOccupancyList" type="tns:RtGroupOccupancyList"/>
</wsdl:message>
<wsdl:message name="getRtCatParkOccupancyRequest">
  <wsdl:part name="token" type="xsd:string"/>
</wsdl:message>
<wsdl:message name="getRtCatParkOccupancyResponse">
  <wsdl:part name="rtCatParkOccupancy" type="tns:RtCatGroupOccupancy"/>
</wsdl:message>
<wsdl:message name="getRtCatGroupOccupancyRequest">
  <wsdl:part name="token" type="xsd:string"/>

```

Document Revision	Date	Page
4	22/02/2018	37 / 43

```

    <wsdl:part name="idgroups" type="tns:IntList"/>
</wsdl:message>
<wsdl:message name="getRtCatGroupOccupancyResponse">
    <wsdl:part name="rtCatGroupOccupancyList" type="tns:RtCatGroupOccupancyList"/>
</wsdl:message>
<wsdl:message name="getRtDiagnosticRequest">
    <wsdl:part name="token" type="xsd:string"/>
</wsdl:message>
<wsdl:message name="getRtDiagnosticResponse">
    <wsdl:part name="rtDiagnosticList" type="tns:RtDiagnosticList"/>
</wsdl:message>

<wsdl:message name="getRtMZCOccupiedRequest">
    <wsdl:part name="token" type="xsd:string"/>
</wsdl:message>
<wsdl:message name="getRtMZCOccupiedResponse">
    <wsdl:part name="rtMZCOccupiedList" type="tns:RtMZCOccupiedList"/>
</wsdl:message>

<wsdl:message name="getActiveAlarmsRequest">
    <wsdl:part name="token" type="xsd:string"/>
</wsdl:message>
<wsdl:message name="getActiveAlarmsResponse">
    <wsdl:part name="activeAlarmsList" type="tns:AlarmsList"/>
</wsdl:message>
<wsdl:message name="getAlarmsByTimeRequest">
    <wsdl:part name="token" type="xsd:string"/>
    <wsdl:part name="starTime" type="xsd:int"/>
    <wsdl:part name="endTime" type="xsd:int"/>
</wsdl:message>
<wsdl:message name="getAlarmsByTimeResponse">
    <wsdl:part name="alarmsList" type="tns:AlarmsList"/>
</wsdl:message>

<wsdl:message name="getParkOccupancyRequest">
    <wsdl:part name="token" type="xsd:string"/>
    <wsdl:part name="starTime" type="xsd:int"/>
    <wsdl:part name="endTime" type="xsd:int"/>
</wsdl:message>
<wsdl:message name="getParkOccupancyResponse">
    <wsdl:part name="parkOccupancyList" type="tns:GroupOccupancyList"/>
</wsdl:message>
<wsdl:message name="getGroupsOccupancyRequest">
    <wsdl:part name="token" type="xsd:string"/>
    <wsdl:part name="starTime" type="xsd:int"/>
    <wsdl:part name="endTime" type="xsd:int"/>
    <wsdl:part name="idgroups" type="tns:IntList"/>
</wsdl:message>
<wsdl:message name="getGroupsOccupancyResponse">
    <wsdl:part name="groupOccupancyList" type="tns:GroupOccupancyList"/>
</wsdl:message>

<wsdl:message name="changeBaysCategoryRequest">
    <wsdl:part name="token" type="xsd:string"/>
    <wsdl:part name="idcategory" type="xsd:int"/>
    <wsdl:part name="idbays" type="tns:IntList"/>
</wsdl:message>
<wsdl:message name="changeBaysCategoryResponse">
    <wsdl:part name="chCatResp" type="tns:changeCategoryResponse"/>
</wsdl:message>

<wsdl:message name="playSequenceRequest">
    <wsdl:part name="token" type="xsd:string"/>
    <wsdl:part name="idsequence" type="xsd:int"/>
</wsdl:message>
<wsdl:message name="playSequenceResponse">
    <wsdl:part name="result" type="xsd:int"/>
</wsdl:message>
<wsdl:message name="setMzcValueRequest">
    <wsdl:part name="token" type="xsd:string"/>
    <wsdl:part name="idmzc" type="xsd:int"/>
    <wsdl:part name="value" type="xsd:int"/>
</wsdl:message>
<wsdl:message name="setMzcValueResponse">
    <wsdl:part name="result" type="xsd:int"/>
</wsdl:message>
<wsdl:message name="resetMzcRequest">
    <wsdl:part name="token" type="xsd:string"/>
    <wsdl:part name="idmzc" type="xsd:int"/>
</wsdl:message>
<wsdl:message name="resetMzcResponse">
    <wsdl:part name="result" type="xsd:int"/>

```

Document Revision	Date	Page
4	22/02/2018	38 / 43

```

</wsdl:message>
<wsdl:message name="changeIndicatorCategoryRequest">
  <wsdl:part name="token" type="xsd:string"/>
  <wsdl:part name="idcategory" type="xsd:int"/>
  <wsdl:part name="idindicator" type="xsd:int"/>
</wsdl:message>
<wsdl:message name="changeIndicatorCategoryResponse">
  <wsdl:part name="chIndCatResp" type="tns:changeCatIndResponse"/>
</wsdl:message>

<wsdl:message name="changeBaysEventCategoryRequest">
  <wsdl:part name="token" type="xsd:string"/>
  <wsdl:part name="idcategory" type="xsd:int"/>
  <wsdl:part name="idbays" type="tns:IntList"/>
</wsdl:message>
<wsdl:message name="changeBaysEventCategoryResponse">
  <wsdl:part name="chEvCatResp" type="tns:changeCategoryResponse"/>
</wsdl:message>

<wsdl:message name="acknowledgeAlarmsRequest">
  <wsdl:part name="token" type="xsd:string"/>
  <wsdl:part name="idalarms" type="tns:IntList"/>
</wsdl:message>
<wsdl:message name="acknowledgeAlarmsResponse">
  <wsdl:part name="result" type="xsd:int"/>
</wsdl:message>

<wsdl:portType name="CP3WebInterface">

  <wsdl:operation name="login">
    <wsdl:input message="tns:loginRequest"/>
    <wsdl:output message="tns:loginResponse"/>
  </wsdl:operation>

  <wsdl:operation name="getGroups">
    <wsdl:input message="tns:getGroupsRequest"/>
    <wsdl:output message="tns:getGroupsResponse"/>
  </wsdl:operation>
  <wsdl:operation name="getGroupByld">
    <wsdl:input message="tns:getGroupByldRequest"/>
    <wsdl:output message="tns:getGroupByldResponse"/>
  </wsdl:operation>

  <wsdl:operation name="getBays">
    <wsdl:input message="tns:getBaysRequest"/>
    <wsdl:output message="tns:getBaysResponse"/>
  </wsdl:operation>
  <wsdl:operation name="getBaysByld">
    <wsdl:input message="tns:getBaysByldRequest"/>
    <wsdl:output message="tns:getBaysByldResponse"/>
  </wsdl:operation>

  <wsdl:operation name="getCategories">
    <wsdl:input message="tns:getCategoriesRequest"/>
    <wsdl:output message="tns:getCategoriesResponse"/>
  </wsdl:operation>
  <wsdl:operation name="getEventCategories">
    <wsdl:input message="tns:getEventCategoriesRequest"/>
    <wsdl:output message="tns:getEventCategoriesResponse"/>
  </wsdl:operation>

  <wsdl:operation name="getSequences">
    <wsdl:input message="tns:getSequencesRequest"/>
    <wsdl:output message="tns:getSequencesResponse"/>
  </wsdl:operation>
  <wsdl:operation name="getMZC">
    <wsdl:input message="tns:getMZCRequest"/>
    <wsdl:output message="tns:getMZCResponse"/>
  </wsdl:operation>
  <wsdl:operation name="getIndicators">
    <wsdl:input message="tns:getIndicatorsRequest"/>
    <wsdl:output message="tns:getIndicatorsResponse"/>
  </wsdl:operation>
  <wsdl:operation name="getSxweb">
    <wsdl:input message="tns:getSxwebRequest"/>
    <wsdl:output message="tns:getSxwebResponse"/>
  </wsdl:operation>

  <wsdl:operation name="getRtBaysStatus">
    <wsdl:input message="tns:getRtBaysStatusRequest"/>
    <wsdl:output message="tns:getRtBaysStatusResponse"/>
  </wsdl:operation>

```

Document Revision	Date	Page
4	22/02/2018	39 / 43

```

</wsdl:operation>
<wsdl:operation name="getRtBaysStatusById">
  <wsdl:input message="tns:getRtBaysStatusByIdRequest"/>
  <wsdl:output message="tns:getRtBaysStatusByIdResponse"/>
</wsdl:operation>
<wsdl:operation name="getRtBaysStatusByGroup">
  <wsdl:input message="tns:getRtBaysStatusByGroupRequest"/>
  <wsdl:output message="tns:getRtBaysStatusByGroupResponse"/>
</wsdl:operation>

<wsdl:operation name="getRtParkOccupancy">
  <wsdl:input message="tns:getRtParkOccupancyRequest"/>
  <wsdl:output message="tns:getRtParkOccupancyResponse"/>
</wsdl:operation>
<wsdl:operation name="getRtGroupOccupancy">
  <wsdl:input message="tns:getRtGroupOccupancyRequest"/>
  <wsdl:output message="tns:getRtGroupOccupancyResponse"/>
</wsdl:operation>
<wsdl:operation name="getRtCatParkOccupancy">
  <wsdl:input message="tns:getRtCatParkOccupancyRequest"/>
  <wsdl:output message="tns:getRtCatParkOccupancyResponse"/>
</wsdl:operation>
<wsdl:operation name="getRtCatGroupOccupancy">
  <wsdl:input message="tns:getRtCatGroupOccupancyRequest"/>
  <wsdl:output message="tns:getRtCatGroupOccupancyResponse"/>
</wsdl:operation>
<wsdl:operation name="getRtDiagnostic">
  <wsdl:input message="tns:getRtDiagnosticRequest"/>
  <wsdl:output message="tns:getRtDiagnosticResponse"/>
</wsdl:operation>
<wsdl:operation name="getRtMZCOccupied">
  <wsdl:input message="tns:getRtMZCOccupiedRequest"/>
  <wsdl:output message="tns:getRtMZCOccupiedResponse"/>
</wsdl:operation>

<wsdl:operation name="getActiveAlarms">
  <wsdl:input message="tns:getActiveAlarmsRequest"/>
  <wsdl:output message="tns:getActiveAlarmsResponse"/>
</wsdl:operation>
<wsdl:operation name="getAlarmsByTime">
  <wsdl:input message="tns:getAlarmsByTimeRequest"/>
  <wsdl:output message="tns:getAlarmsByTimeResponse"/>
</wsdl:operation>

<wsdl:operation name="getParkOccupancy">
  <wsdl:input message="tns:getParkOccupancyRequest"/>
  <wsdl:output message="tns:getParkOccupancyResponse"/>
</wsdl:operation>
<wsdl:operation name="getGroupsOccupancy">
  <wsdl:input message="tns:getGroupsOccupancyRequest"/>
  <wsdl:output message="tns:getGroupsOccupancyResponse"/>
</wsdl:operation>

<wsdl:operation name="changeBaysCategory">
  <wsdl:input message="tns:changeBaysCategoryRequest"/>
  <wsdl:output message="tns:changeBaysCategoryResponse"/>
</wsdl:operation>
<wsdl:operation name="playSequence">
  <wsdl:input message="tns:playSequenceRequest"/>
  <wsdl:output message="tns:playSequenceResponse"/>
</wsdl:operation>
<wsdl:operation name="setMzcValue">
  <wsdl:input message="tns:setMzcValueRequest"/>
  <wsdl:output message="tns:setMzcValueResponse"/>
</wsdl:operation>
<wsdl:operation name="resetMzc">
  <wsdl:input message="tns:resetMzcRequest"/>
  <wsdl:output message="tns:resetMzcResponse"/>
</wsdl:operation>
<wsdl:operation name="changeIndicatorCategory">
  <wsdl:input message="tns:changeIndicatorCategoryRequest"/>
  <wsdl:output message="tns:changeIndicatorCategoryResponse"/>
</wsdl:operation>
<wsdl:operation name="changeBaysEventCategory">
  <wsdl:input message="tns:changeBaysEventCategoryRequest"/>
  <wsdl:output message="tns:changeBaysEventCategoryResponse"/>
</wsdl:operation>
<wsdl:operation name="acknowledgeAlarms">
  <wsdl:input message="tns:acknowledgeAlarmsRequest"/>
  <wsdl:output message="tns:acknowledgeAlarmsResponse"/>
</wsdl:operation>

```

Document Revision	Date	Page
4	22/02/2018	40 / 43

</wsdl:portType>

```

<wsdl:binding name="serviceSOAP" type="tns:CP3WebInterface">
  <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>

  <wsdl:operation name="login">
    <soap:operation soapAction="http://localhost/CP3Service/login"/>
    <wsdl:input>
      <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
    </wsdl:output>
  </wsdl:operation>

  <wsdl:operation name="getGroups">
    <soap:operation soapAction="http://localhost/CP3Service/getGroups"/>
    <wsdl:input>
      <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
    </wsdl:output>
  </wsdl:operation>

  <wsdl:operation name="getGroupById">
    <soap:operation soapAction="http://localhost/CP3Service/getGroupById"/>
    <wsdl:input>
      <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
    </wsdl:output>
  </wsdl:operation>

  <wsdl:operation name="getBays">
    <soap:operation soapAction="http://localhost/CP3Service/getBays"/>
    <wsdl:input>
      <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
    </wsdl:output>
  </wsdl:operation>

  <wsdl:operation name="getBaysById">
    <soap:operation soapAction="http://localhost/CP3Service/getBaysById"/>
    <wsdl:input>
      <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
    </wsdl:output>
  </wsdl:operation>

  <wsdl:operation name="getCategories">
    <soap:operation soapAction="http://localhost/CP3Service/getCategories"/>
    <wsdl:input>
      <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
    </wsdl:output>
  </wsdl:operation>

  <wsdl:operation name="getEventCategories">
    <soap:operation soapAction="http://localhost/CP3Service/getEventCategories"/>
    <wsdl:input>
      <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
    </wsdl:output>
  </wsdl:operation>

  <wsdl:operation name="getSequences">
    <soap:operation soapAction="http://localhost/CP3Service/getSequences"/>
    <wsdl:input>
      <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
    </wsdl:output>
  </wsdl:operation>

```



Document Revision	Date	Page
4	22/02/2018	41 / 43

```

<wsdl:operation name="getMZC">
  <soap:operation soapAction="http://localhost/CP3Service/getMZC"/>
  <wsdl:input>
    <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="getIndicators">
  <soap:operation soapAction="http://localhost/CP3Service/getIndicators"/>
  <wsdl:input>
    <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="getSxweb">
  <soap:operation soapAction="http://localhost/CP3Service/getSxweb"/>
  <wsdl:input>
    <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="getRtBaysStatus">
  <soap:operation soapAction="http://localhost/CP3Service/getRtBaysStatus"/>
  <wsdl:input>
    <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="getRtBaysStatusById">
  <soap:operation soapAction="http://localhost/CP3Service/getRtBaysStatusById"/>
  <wsdl:input>
    <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="getRtBaysStatusByGroup">
  <soap:operation soapAction="http://localhost/CP3Service/getRtBaysStatusByGroup"/>
  <wsdl:input>
    <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="getRtParkOccupancy">
  <soap:operation soapAction="http://localhost/CP3Service/getRtParkOccupancy"/>
  <wsdl:input>
    <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="getRtGroupOccupancy">
  <soap:operation soapAction="http://localhost/CP3Service/getRtGroupOccupancy"/>
  <wsdl:input>
    <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="getRtCatParkOccupancy">
  <soap:operation soapAction="http://localhost/CP3Service/getRtParkOccupancy"/>
  <wsdl:input>
    <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
  </wsdl:output>
</wsdl:operation>

```

Document Revision	Date	Page
4	22/02/2018	42 / 43

```

</wsdl:operation>
  <wsdl:operation name="getRtCatGroupOccupancy">
    <soap:operation soapAction="http://localhost/CP3Service/getCatGroupOccupancy"/>
    <wsdl:input>
      <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
    </wsdl:output>
  </wsdl:operation>

<wsdl:operation name="getRtDiagnostic">
  <soap:operation soapAction="http://localhost/CP3Service/getRtDiagnostic"/>
  <wsdl:input>
    <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="getRtMZCOccupied">
  <soap:operation soapAction="http://localhost/CP3Service/getRtMZCOccupied"/>
  <wsdl:input>
    <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="getActiveAlarms">
  <soap:operation soapAction="http://localhost/CP3Service/getActiveAlarms"/>
  <wsdl:input>
    <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="getAlarmsByTime">
  <soap:operation soapAction="http://localhost/CP3Service/getAlarmsByTime"/>
  <wsdl:input>
    <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="getParkOccupancy">
  <soap:operation soapAction="http://localhost/CP3Service/getParkOccupancy"/>
  <wsdl:input>
    <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="getGroupsOccupancy">
  <soap:operation soapAction="http://localhost/CP3Service/getGroupsOccupancy"/>
  <wsdl:input>
    <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="changeBaysCategory">
  <soap:operation soapAction="http://localhost/CP3Service/changeBaysCategory"/>
  <wsdl:input>
    <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="playSequence">
  <soap:operation soapAction="http://localhost/CP3Service/playSequence"/>
  <wsdl:input>
    <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>

```

Document Revision	Date	Page
4	22/02/2018	43 / 43

```

        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="setMzcValue">
        <soap:operation soapAction="http://localhost/CP3Service/setMzcValue"/>
        <wsdl:input>
            <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="resetMzc">
        <soap:operation soapAction="http://localhost/CP3Service/resetMzc"/>
        <wsdl:input>
            <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="changeIndicatorCategory">
        <soap:operation soapAction="http://localhost/CP3Service/changeIndicatorCategory"/>
        <wsdl:input>
            <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
        </wsdl:output>
    </wsdl:operation>

    <wsdl:operation name="changeBaysEventCategory">
        <soap:operation soapAction="http://localhost/CP3Service/changeBaysEventCategory"/>
        <wsdl:input>
            <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
        </wsdl:output>
    </wsdl:operation>

    <wsdl:operation name="acknowledgeAlarms">
        <soap:operation soapAction="http://localhost/CP3Service/acknowledgeAlarms"/>
        <wsdl:input>
            <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" namespace="http://localhost/CP3Service/public/CP3WebInterface"/>
        </wsdl:output>
    </wsdl:operation>

</wsdl:binding>

<wsdl:service name="CP3WebInterface">
    <wsdl:port binding="tns:serviceSOAP" name="serviceSOAP">
        <soap:address location="https://localhost/CP3Service/public/CP3WebInterface.php"/>
    </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```