



Via Orzinuovi 73
25125 Brescia
Telefono: +39 0303581239
Fax: +39 0302659974
Posta elettronica: info@gullivernet.com
Web: www.gullivernet.com

MDC Data Structure via Scripts

Technical guide

Ver. 1

Versions

<i>Version</i>	<i>Notes</i>
Ver. 1 17/06/2016	First writing.

Summary

[Overview](#)

[How to get data](#)

[Screen types and data](#)

[Free capture](#)

[Single selection](#)

[Multiple selection](#)

[Text](#)

[Multi-line text](#)

[Integer numeric](#)

[Decimal numeric](#)

[Date](#)

[Time](#)

[Date and time](#)

[Multi-capture](#)

[Signature](#)

[Drawing](#)

[Picture](#)

[Attachment](#)

[Payment](#)

[Capture from list](#)

[Text grid](#)

[Integer numeric grid](#)

[Decimal numeric grid](#)

[Capture from table \(Lookup\)](#)

[Single selection](#)

[Single selection with search](#)

[Multiple selection](#)

[Multiple selection with search](#)

[Text grid](#)

[Multi-line grid](#)

[Integer numeric grid](#)

[Decimal numeric grid](#)

[Date grid](#)

[Time grid](#)

[Multi-capture grid](#)

[Multi-capture grid \(compact view\)](#)

[CalendarPayment](#)

[Summary](#)

Overview

The following document explains how the user can manage data on MDC client through the scripts functionality.

How to get data

All the data inserted by the user with a MDC App are collected in different objects that differs by the screen type, but one property of these object is the same for all the screen: the value. The value is an object of type `JSFormValue`: it is composed by the field `value` that contains the data inserted by the user. All these data are structured in different ways and differs from screen to screen. The user can retrieve this object by using the script `JSMDc.getForm()` as in the following sample code:

```
var frmJson = JSMDc.getForm(currentIdd);
var frm = JSON.parse(frmJson);

var value = frm.value.value;
```

Some screen (for example multiple selection, multi-capture grid) have also a parameter that contains additional data saved into the property `listofvaluesext`. This object is an array of `JSFormValueExt`.

Below an example with the iteration on this array:

```
var listofValueExt = frm.listofvaluesext;
for(i = 0; i < listofValueExt.length; i++)
{
    var valueExt = listofValueExt[i];
}
```

Screen types and data

As we said before, every screen saves the data in different ways. Below is the list of the screen with a sample of acquisition and the detail of the objects `JSFormValue` and `JSFormValueExt` both in table and JSON format.

Free capture

Single selection

The image shows a mobile application window with a title bar containing a menu icon, the text 'APP', and navigation arrows. Below the title bar, the screen is titled 'Free capture - Single selection'. It displays a list of five radio button options: '#1 choice', '#2 choice' (which is selected), '#3 choice', '#4 choice', and '#5 choice'.

The selected choice is saved in the field `value` with the following schema:

`idi1|idi2|idi3|...|idin`

The order of the elements is the same as that of visualization and the value is only present for the selected option (idi) while the other options are blank.

value
2

JSFormValue

```
{
  "value": "|2|||",
  "dtttime": "20150709103450"
}
```

Multiple selection

APP

Free capture - Multiple Selection

☐ # 1 choice

☒ #2 choice

☐ #3 choice

☒ #4 choice

☐ #5 choice

The selected choice is saved in the field `value` with the following schema:

`idi1|idi2|idi3|...|idin`

The order of the elements is the same as that of visualization and the value is only present for the selected option (idi) while the other options are blank.

value
2 4

JSFormValue

```
{
  "value": "|2||4|",
  "dttime": "20150709103450"
}
```

Text

APP

Free capture - Text

This is a short message.

The entered text is saved in the field `value`.

value
This is a short message.

JSFormValue

```
{
  "value": "This is a short message.",
  "dttime": "20150709103450"
}
```

Multi-line text

Free capture - Multi-line text

This is a long message, this is a long message, this is a long message, this is a long message, this is a long message.

The entered text is saved in the field `value`.

value
This is a long message, this is a long message, this is a long message, this is a long message, this is a long message.

JSFormValue

```
{
  "value": "This is a long message, this is a long message, this is a long message,
this is a long message, this is a long message, this is a long message.",
  "dtttime": "20150709103450"
}
```

Integer numeric

APP

Free capture - Integer numeric

35

The entered text is saved in the field value.

value
35

JSFormValue

```
{
  "value": "35",
  "dttime": "20150709103450"
}
```


Decimal numeric

The image shows a mobile application interface. At the top, there is a header bar with a hamburger menu icon on the left, the text "APP" in the center, and left and right arrow icons on the right. Below the header, the text "Free capture - Decimal numeric" is displayed. Underneath this text is a text input field containing the value "5,2".

The entered text is saved in the field `value`.

value
5,2

JSFormValue

```
{  
  "value": "5,2",  
  "dttime": "20150709103450"  
}
```

Date

APP

Free capture - Date

11	^	March	^	2014	^
12		April		2015	
13	v	June	v	2016	v

The entered text is saved in the field value.

value
12/04/2015

JSFormValue

```
{
  "value": "12/04/2015",
  "dtttime": "20150709103450"
}
```

Time

APP

Free capture - Time

11	^	34	^
12		35	
13	v	36	v

The entered text is saved in the field value.

value
12:35

JSFormValue

```
{
  "value": "12:35",
  "dttime": "20150709103450"
}
```

Date and time

Free capture - Date and time

11	^	March	^	2014	^
12		April		2015	
13	v	June	v	2016	v

11	^	34	^
12		35	
13	v	36	v

The entered text is saved in the field `value`.

value
12/04/2015 12.35

JSFormValue

```
{
  "value": "12/04/2015|12.35",
  "dtttime": "20150709103450"
}
```

Multi-capture

App

Free capture - Multi capture

Integer

3

Decimal

3,5

String

Hello

Date

11

12

13

March

April

June

2014

2015

2016

Time (hours)

11

12

13

34

35

36

Lookup

Action 001

Photo

Items

☒ # 1 choice

☒ # 2 choice

☐ # 3 choice

☐ # 4 choice

☐ # 5 choice

The entered data are saved in the field `value`.

value
3 3,5 Hello 12/04/2015 12:35 ACTION:001 /path/to/photo/filename1.jpg;/path/to/photo/filename2.jpg 1 2 /TGK=11618-11680TGK/

Below is the schema:

value_idacq1|value_idacq2|value_idacq3|...|value_idacqn|idi1|idi2|idi3|...|idin

The order of the elements is the same as that of visualization and the value is present only for the selected options (*idi*) and for the entered fields (*idacq*). The other fields are blank. The char “|” is used to separate every field even if the field has no value.

The key of the record has the following format:

/TGK=idq-iddTGK/

JSFormValue

```
{
  "value":
  "3|3,5|Hello|12/04/2015|12:35|ACTION:001|/path/to/photo/filename1.jpg;/path/to/photo/file
  name2.jpg|1|2|||/TGK=11618-11680TGK/",
  "dttime": "20150709103450"
}
```

In order to read the data, you can use the object *JSFormValueExt*: here is a summary schema for every type of data:

type	progr	key	val01	val02
integer	1	- 11618-11680 - a - 01 - 1	3	
decimal	2	- 11618-11680 - a - 01 - 2	3,5	
string	3	- 11618-11680 - a - 01 - 3	Hello	
date	4	- 11618-11680 - a - 01 - 4	12/04/2015	
time	5	- 11618-11680 - a - 01 - 5	12:35	
lookup	6	- 11618-11680 - a - 01 - 6	ACTION	001
photo	7	- 11618-11680 - a - 01 - 7	/path/to/photo/filename1.jpg	
photo	8	- 11618-11680 - a - 01 - 7	/path/to/photo/filename2.jpg	
item	9	- 11618-11680 - i - 02 - 1		
item	10	- 11618-11680 - i - 02 - 2		

The field *key* contains the key of the selected record. The key can be composed by different parts depending on the type of record: acquisition table or option group.

In the case of “option group”, *key* has a value composed by the key of the record and the key of the selected item (*idi*), while *val01* has no value. Below the sample code for the *key*:

```
||-|idq-idd|-|i|-|<idri>|-|<idi>|
```

In the case of “acquisition table”, *key* has a value composed by the key of the record and the key of the record of the acquisition table (*idacq*). *val01* contains the value inserted by the user.

```
||-|idq-idd|-|a|-|<idgracq>|-|<idacq>|
```

Below the *JSFormValueExt* objects for the above sample:

JSFormValueExt***Integer***

```
{
```

```

"key": "||-|11618-11680|-|a|-|01|-|1|",
"tabgen": {
  "key": "11618-11680",
  "tabname": "",
  "val01": "",
  ...
  "val20": ""
},
"val01": "3",
...
"progressivo": 1
}

```

Decimal

```

{
  "key": "||-|11618-11680|-|a|-|01|-|2|",
  "tabgen": {...},
  "val01": "3,5",
  ...
  "progressivo": 2
}

```

String

```

{
  "key": "||-|11618-11680|-|a|-|01|-|3|",
  "tabgen": {...},
  "val01": "Hello",
  ...
  "progressivo": 3
}

```

Date

```

{
  "key": "||-|11618-11680|-|a|-|01|-|4|",
  "tabgen": {...},
  "val01": "09/07/2015",
  ...
  "progressivo": 4
}

```

Time (hours)

```

{
  "key": "||-|11618-11680|-|a|-|01|-|5|",
  "tabgen": {...},
  "val01": "10:32",
  ...
  "progressivo": 5
}

```

```
}
```

Lookup

```
{
  "key": "||-|11618-11680|-|a|-|01|-|6|",
  "tabgen": {...},
  "val01": "ANA_CLIENTI",
  "val02": "10-323675",
  ...
  "progressivo": 6
}
```

Photo

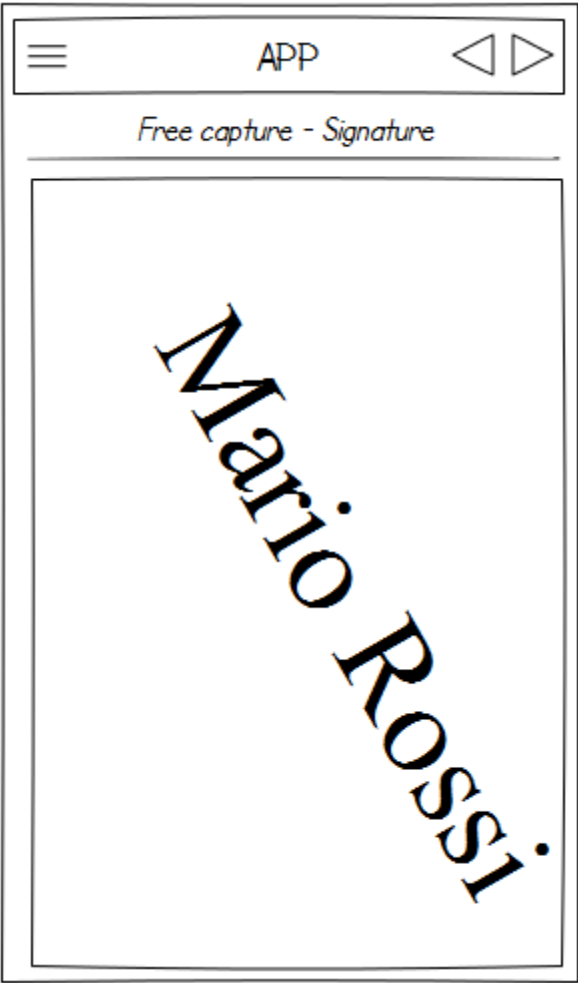
N.B.: there is one JSFormValueExt for every photo.

```
{
  "key": "||-|11618-11680|-|a|-|01|-|7|",
  "tabgen": {...},
  "val01": "/path/to/photo/filename1.jpg",
  ...
  "progressivo": 7
}
```

Items

```
{
  "key": "||-|11618-11680|-|i|-|02|-|2|",
  "tabgen": {...},
  "val01": "",
  ...
  "progressivo": 9
}
```


Signature



The path of the image relating to the signature is saved into the field *value*.

value
/path/to/signature/filename.jpeg

JSFormValue

```
{
  "dttime": "20150708102224",
  "value": "/path/to/signature/filename.jpeg"
}
```

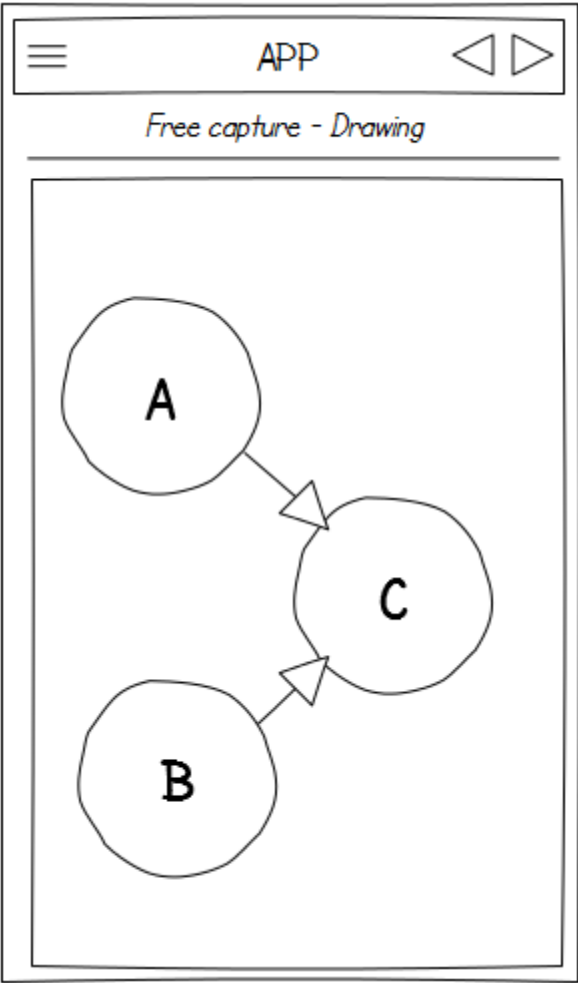
The path can be retrieved by reading the `JSFormValueExt`:

progressivo	key	val01
1		/path/to/signature/filename.jpeg

JSFormValueExt

```
{
  "key": "",
  "val01": "/path/to/signature/filename.jpeg",
  ...
  "progressivo": 1
}
```

Drawing



The path of the image relating to the drawing is saved into the field *value*.

value
/path/to/drawing/filename.jpeg

JSFormValue

```
{
  "value": "/path/to/drawing/filename.jpeg",
  "dtttime": "20150708102224"
}
```

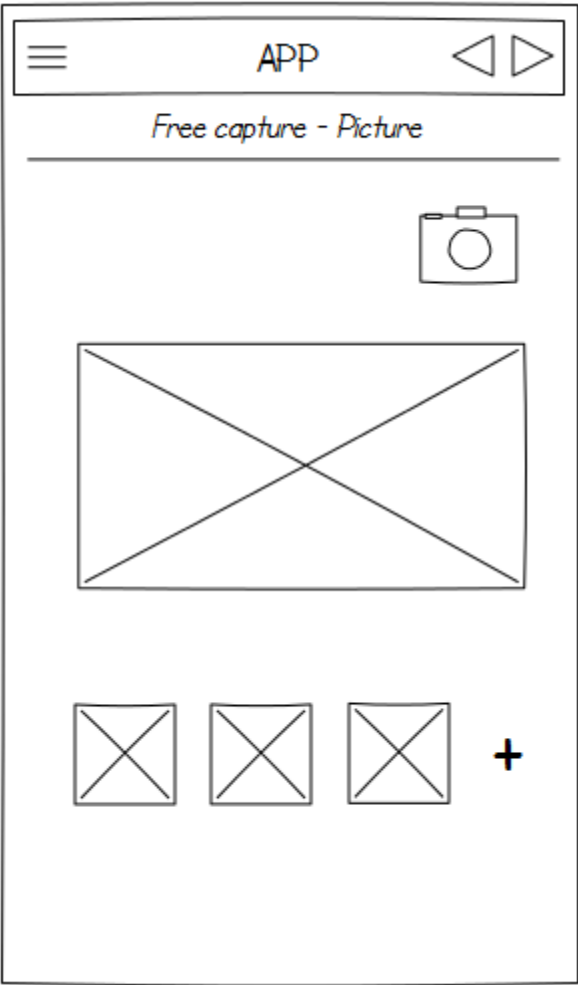
The path can be retrieved by reading the `JSFormValueExt`:

progressivo	key	val01
1		/path/to/drawing/filename.jpeg

JSFormValueExt

```
{  
  "key": "",  
  "val01": "/path/to/drawing/filename.jpeg",  
  ...  
  "progressivo": 1  
}
```

Picture



The path of the photo is saved into the field *value*. If more photos are present the paths are concatenated by the char “|”:

path_photo1|path_photo2|...|path_photon

value
/path/to/photo/filename1.jpeg /path/to/photo/filename2.jpeg

JSFormValue

```
{
  "value": "/path/to/photo/filename1.jpeg|/path/to/photo/filename2.jpeg",
  "dttime": "20150708102224"
}
```

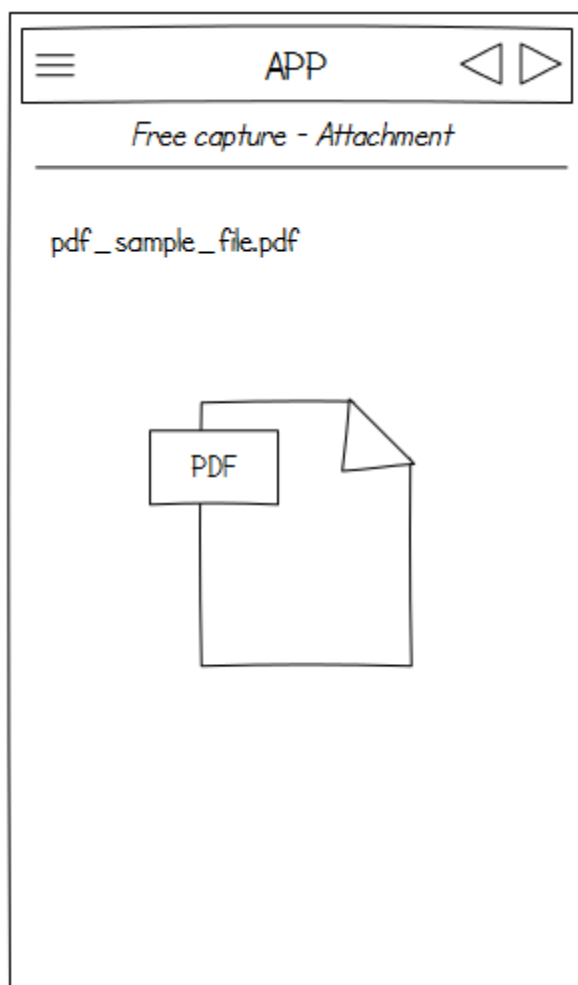
The path can be retrieved by reading the `JSFormValueExt`:

progressivo	key	val01
1		/path/to/photo/filename1.jpeg
2		/path/to/photo/filename2.jpeg

JSFormValueExt

```
{
  "key": "",
  "val01": "/path/to/photo/filename1.jpeg",
  ...
  "progressivo": 1
}
{
  "key": "",
  "val01": "/path/to/photo/filename2.jpeg",
  ...
  "progressivo": 2
}
```

Attachment



The path of the photo is saved into the field *value*.

value
/path/to/file/pdf_sample_file.pdf

JSFormValue

```
{  
  "value": "/path/to/file/pdf_sample_file.pdf",  
  "dtttime": "20150708102224"  
}
```

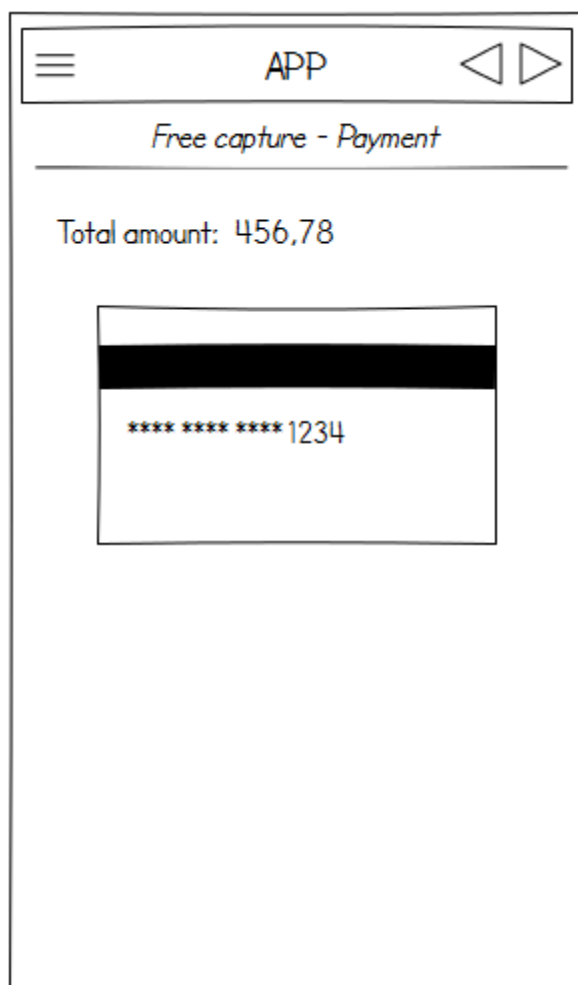
The path can be retrieved by reading the `JSFormValueExt`:

progressivo	key	val01
1		/path/to/file/pdf sample file.pdf

JSFormValueExt

```
{  
  "key": "",  
  "val01": "/path/to/file/pdf_sample_file.pdf",  
  ...  
  "progressivo": 1  
}
```


Payment



You can check if the payment succeeded by checking the field *value*.

Android

value
190

iOS

value
0

WP: payment not available

JSFormValue

```
{
  "value": "0",
  "dtttime": "20150708102224"
}
```

The object JSFormValueExt contains the payment details (N.B.: the values differs depending on the platform):

Android

progr	key	val01	val02
1		result	190
2		com.setefi.pos.external.pay.cvm	2
3		com.setefi.pos.external.pay.pan	*****6314
4		com.setefi.pos.external.pay.acquirer_id	4
5		com.setefi.pos.external.pay.exp	****
6		com.setefi.pos.external.pay.termid	97701158
7		com.setefi.pos.external.pay.operation_number	47
8		com.setefi.pos.external.pay.authoritation_number	862960
9		com.setefi.pos.external.pay.timestamp	2007151430
10		com.setefi.pos.external.pay.stan	6
11		com.setefi.pos.external.pay.amount	01.00.00
12		com.setefi.pos.external.pay.function	500
13		com.setefi.pos.external.pay.trans_type	CLI
14		com.setefi.pos.external.pay.merchant_id	303186300140013

iOS

progr	key	val01	val02
1		acquirer_id	000000000004
2		exp	
3		amount	1,00
4		pan	*****6314
5		stan	000004
6		timestamp	1507151448
7		authorization number	432022
8		cvm	0
9		operation number	000045
10		returnCode	0
11		trans_type	CLI
12		merchant_id	303186300140013
13		termid	97701158

WP (payment not available)

The object `JSFormValueExt` can assume different values depending on the platform: *key* is always blank, *val01* contains the name of the payment parameter, *val02* the value of the payment parameter. The following is an example:

JSFormValueExt

```
{
  "key": "",
  "val01": "acquirer_id",
  "val02": "000000000004",
  ...
  "progressivo": 1
}
```

Capture from list

Text grid

The entered values are saved into the field *value* with this schema:

idri-idi_answer_1{value_answer_1}|idi_answer_2{value_answer_2}|..|idi_answer_n{value_answer_n}

The order of the items is the same as that of visualization and the value is present only for the selected options (idi) while the other options are blank.

value
3-1{alfa} 3{beta} 4{gamma}

JSFormValue

```
{
  "value": "3-1{alfa}||3{beta}|4{gamma}|",
  "dttime": "20150709103450"
}
```

}

Integer numeric grid

The values entered by a user are saved into the field *value* with this schema:

idri-idi_answer_1{value_answer_1}|idi_answer_2{value_answer_2}|..|idi_answer_n{value_answer_n}

The order of the items is the same as that of visualization and the value is present only for the selected options (idi) while the other options are blank.

value
3-1{43} 3{57}

JSFormValue

```
{
  "value": "3-1{43}||3{57}||",
  "dttime": "20150709103450"
}
```



}

Decimal numeric grid

The screenshot shows a mobile application window with a title bar containing a menu icon, the text 'APP', and navigation arrows. Below the title bar is a subtitle 'Capture from list - Decimal numeric grid'. The main content area contains five vertically stacked input fields, each preceded by a label: '#1 choice', '#2 choice', '#3 choice', '#4 choice', and '#5 choice'. The input fields are rectangular. The second field contains the text '2,1', the third contains '3,4', and the fifth contains '5,7'. The first and fourth fields are empty.

The values entered by a user are saved into the field *value* with this schema:

idri-idi_answer_1{value_answer_1}|idi_answer_2{value_answer_2}|..|idi_answer_n{value_answer_n}

The order of the items is the same as that of visualization and the value is present only for the selected options (idi) while the other options are blank.

value
3- 2{2,1} 3{3,4} 5{5,7}

JSFormValue

```
{
  "value": "3-|2{2,1}|3{3,4}||5{5,7}",
  "dttime": "20150709103450"
}
```

Capture from table (Lookup)

Single selection

The key of the selected item is saved into the field *value*.

value
002

JSFormValue

```
{
  "value": "002",
  "dtttime": "20150709103450"
}
```


The value can be retrieved from the object `JSFormValueExt`:

progressivo	key	val01	val02
1	CUSTOMER - 002		

key contains the key of the record of a specific table and it is composed by the following structure:

|<tablename>|-|<reckey_record>|

val01 is a blank field.

JSFormValueExt

```
{
  "key": "|CUSTOMER|-|002|",
  "tabgen": {
    "key": "002",
    "tablename": "CUSTOMER",
    "val01": "",
    ...
    "val20": ""
  },
  "val01": "",
  ...
  "progressivo": 1
}
```

Single selection with search

The key of the selected item is saved into the field *value*.

value
002

JSFormValue

```
{
  "value": "002",
  "dtttime": "20150709103450"
}
```

The value can be retrieved from the object `JSFormValueExt`:

progressivo	key	val01	val02
1	CUSTOMER - 002		

key contains the key of the record of a specific table and it is composed by the following structure:

|<tablename>|-|<reckey_record>|

val01 is a blank field.

JSFormValueExt

```
{
  "key": "|CUSTOMER|-|002|",
  "tabgen": {
    "key": "002",
    "tablename": "CUSTOMER",
    "val01": "",
    ...
    "val20": ""
  },
  "val01": "",
  ...
  "progressivo": 1
}
```

Multiple selection

The screenshot shows a mobile application window with a title bar containing a menu icon, the text 'APP', and navigation arrows. Below the title bar, the text 'Capture from Table' and 'Multiple selection' is displayed. A list of five items follows, each consisting of a square icon with an 'X' and a checkbox with a label:

- ☒ Customer 001
- ☒ Customer 002
- ☐ Customer 003
- ☐ Customer 004
- ☐ Customer 005

The key of the selected items are saved into the field *value*.

value
001 002

JSFormValue

```
{
  "value": "001|002",
  "dtttime": "20150709103450"
}
```

The value can be retrieved from the object `JSFormValueExt`:

progressivo	key	val01	val02
1	CUSTOMER - 001	val01	
2	CUSTOMER - 002		

key contains the key of the record of a specific table and it is composed by the following structure:

|<tablename>|-|<reckey_record>|

val01 is a blank field.

JSFormValueExt

```
{
  "key": "|CUSTOMER|-|001|",
  "tabgen": {
    "key": "001",
    "tablename": "CUSTOMER",
    "val01": "",
    ...
    "val20": ""
  },
  "val01": "",
  ...
  "progressivo": 1
}
{
  "key": "|CUSTOMER|-|002|",
  "tabgen": {
    "key": "002",
    "tablename": "CUSTOMER",
    "val01": "",
    ...
    "val20": ""
  },
  "val01": "",
  ...
  "progressivo": 2
}
```

Multiple selection with search

The key of the selected item is saved into the field *value*.

value
001 002

JSFormValue

```
{
  "value": "001|002",
  "dtttime": "20150709103450"
}
```

The value can be retrieved from the object `JSFormValueExt`:

progressivo	key	val01	val02
1	CUSTOMER - 001		
2	CUSTOMER - 002		

key contains the key of the record of a specific table and it is composed by the following structure:

|<tablename>|-|<reckey_record>|

val01 is a blank field.

JSFormValueExt

```
{
  "key": "|CUSTOMER|-|001|",
  "tabgen": {
    "key": "001",
    "tablename": "CUSTOMER",
    "val01": "",
    ...
    "val20": ""
  },
  "val01": "",
  ...
  "progressivo": 1
}
{
  "key": "|CUSTOMER|-|002|",
  "tabgen": {
    "key": "002",
    "tablename": "CUSTOMER",
    "val01": "",
    ...
    "val20": ""
  },
  "val01": "",
  ...
  "progressivo": 2
}
```

Text grid

APP

Capture from Table
Text grid

<input checked="" type="checkbox"/>	Customer 001	Text 1
<input checked="" type="checkbox"/>	Customer 002	Text 2
<input type="checkbox"/>	Customer 003	
<input checked="" type="checkbox"/>	Customer 004	Text 4
<input type="checkbox"/>	Customer 005	

The field *value* contains the value of the fields filled by the user, with the following structure:

value1|value2|value3|...|valuen

The order of the items is the same as that of visualization and the value is present only for the valorized items.

value
Text 1 Text 2 Text 4

JSFormValue

```
{
  "value": "Testo1|Testo2||Testo4|",
  "dtttime": "20150708102028"
}
```

The values can be retrieved from the object JSFormValueExt:

progressivo	key	val01
1	CUSTOMER - 001	Text 1
2	CUSTOMER - 002	Text 2
3	CUSTOMER - 004	Text 4

key contains the key of the record of a specific table and it is composed by the following structure:

|<tablename>|-|<reckey_record>|

val01 contains the value entered by the user for the record.

JSFormValueExt

```
{
  "key": "|CUSTOMER|-|001|",
  "tabgen": {
    "tablename": "CUSTOMER",
    "key" : "001",
    ...
  }
  "progressivo": 1,
  "val01": "Text 1"
}
{
  "key": "|CUSTOMER|-|002|",
  "tabgen": {...},
  "progressivo": 2,
  "val01": "Text 2"
}
{
  "key": "|CUSTOMER|-|004|",
  "tabgen": {...},
  "progressivo": 3,
  "val01": "Text 4"
}
```

Multi-line grid

The field *value* contains the value of the fields filled by the user, with the following structure:

`value1|value2|value3|...|valuen`

The order of the items is the same as that of visualization and the value is present only for the valorized items.

value
This is a long text 1. This is a long text 2. This is a long text 4.

JSFormValue

```
{
  "value": "TestoMoltoLungo1|TestoMoltoLungo2||TestoMoltoLungo4|",
  "dttime": "20150708102028"
}
```

The values can be retrieved from the object `JSFormValueExt`:

progressivo	key	val01
1	CUSTOMER - 001	This is a long text 1.
2	CUSTOMER - 002	This is a long text 2.
3	CUSTOMER - 004	This is a long text 4.

key contains the key of the record of a specific table and it is composed by the following structure:

|<tablename>|-<reckey_record>|

val01 contains the value entered by the user for the record.

JSFormValueExt

```
{
  "key": "|CUSTOMER|-|001|",
  "tabgen": {
    "tablename": "CUSTOMER",
    "key" : "001",
    ...
  },
  "val01": "This is a long text 1.",
  ...
  "progressivo": 1
}
{
  "key": "|CUSTOMER|-|002|",
  "tabgen": {...},
  "val01": "This is a long text 2.",
  ...
  "progressivo": 2
}
{
  "key": "|CUSTOMER|-|004|",
  "tabgen": {...},

  "val01": "This is a long text 4.",
  ...
  "progressivo": 3
}
```

Integer numeric grid

The interface displays a list of customers with the following data:

Customer	Value
Customer 001	12
Customer 002	45
Customer 003	
Customer 004	37
Customer 005	

The field *value* contains the value of the fields filled by the user, with the following structure:

value1|value2|value3|...|valuen

The order of the items is the same as that of visualization and the value is present only for the valorized items.

value
12 45 37

JSFormValue

```
{
  "value": "12|45||37|",
  "dttime": "20150708102028"
}
```

The values can be retrieved from the object `JSFormValueExt`:

progressivo	key	val01
1	CUSTOMER - 001	12
2	CUSTOMER - 002	45
3	CUSTOMER - 004	37

key contains the key of the record of a specific table and it is composed by the following structure:

|<tablename>|-|<reckey_record>|

val01 contains the value inserted by the user for the record.

JSFormValueExt

```
{
  "key": "|CUSTOMER|-|001|",
  "tabgen": {
    "tablename": "CUSTOMER",
    "key" : "001",
    ...
  }
  "val01": "12",
  ...
  "progressivo": 1
}
```

```
{
  "key": "|CUSTOMER|-|002|",
  "tabgen": {...},
  "val01": "45",
  ...
  "progressivo": 2
}
```

```
{
  "key": "|CUSTOMER|-|004|",
  "tabgen": {...},
  "val01": "37",
  ...
  "progressivo": 3
}
```

Decimal numeric grid

Customer	Value
Customer 001	1,2
Customer 002	4,5
Customer 003	
Customer 004	3,7
Customer 005	

The field *value* contains the value of the fields filled by the user, with the following structure:

value1|value2|value3|...|valuen

The order of the items is the same as that of visualization and the value is present only for the valorized items.

value
1,2 4,5 3,7

JSFormValue

```
{
  "value": "1,2|4,5||3,7|",
  "dttime": "20150708102028"
}
```

The values can be retrieved from the object `JSFormValueExt`:

progressivo	keyval	val01
1	CUSTOMER - 001	1,2
2	CUSTOMER - 002	4,5
3	CUSTOMER - 004	3,7

key contains the key of the record of a specific table and it is composed by the following structure:

|<tablename>|-|<reckey_record>|

val01 contains the value inserted by the user for the record.

JSFormValueExt

```
{
  "key": "|CUSTOMER|-|001|",
  "tabgen": {
    "tabname": "CUSTOMER",
    "key" : "001",
    ...
  }
  "progressivo": 1,
  ...
  "val01": "1,2"
}
{
  "key": "|CUSTOMER|-|002|",
  "tabgen": {...},
  "progressivo": 2,
  ...
  "val01": "4,5"
}
{
  "key": "|CUSTOMER|-|004|",
  "tabgen": {...},
  "progressivo": 3,
  ...
  "val01": "3,7"
}
```

Date grid

APP

Capture from Table
Date grid

Customer 001

12	June	2014
13	July	2015
14	August	2016

Customer 002

13	June	2014
14	July	2015
15	August	2016

Customer 003

...
-----	-----	-----

Customer 004

14	June	2014
15	July	2015
16	August	2016

Customer 005

...
-----	-----	-----

The field *value* contains the value of the fields filled by the user, with the following structure:

value1|value2|value3|...|valuen

The order of the items is the same as that of visualization and the value is present only for the valorized items.

value
13/07/2015 14/07/2015 15/07/2015

JSFormValue

```
{
  "value": "Testo1|Testo2||Testo4|",
  "dttime": "20150708102028"
}
```

The values can be retrieved from the object `JSFormValueExt`:

progressivo	keyval	val01
1	CUSTOMER - 001	13/07/2015
2	CUSTOMER - 002	14/07/2015
3	CUSTOMER - 004	15/07/2015

key contains the key of the record of a specific table and it is composed by the following structure:

|<tablename>|-|<reckey_record>|

val01 contains the value inserted by the user for the record.

JSFormValueExt

```
{
  "key": "|CUSTOMER|-|001|",
  "tabgen": {
    "tablename": "CUSTOMER",
    "key" : "001",
    ...
  }
  "val01": "13/07/2015",
  ...
  "progressivo": 1
}
{
  "key": "|CUSTOMER|-|002|",
  "tabgen": {...},
  "val01": "14/07/2015",
  ...
  "progressivo": 2
}
{
  "key": "|CUSTOMER|-|004|",
  "tabgen": {...},
  "val01": "15/07/2015",
  ...
  "progressivo": 3
}
```

Time grid

APP

Capture from Table
Time grid

Customer 001

10	^	30	^
11		31	
12	v	32	v

Customer 002

10	^	31	^
11		32	
12	v	33	v

Customer 003

...	↔	...	↔	...	↔
-----	---	-----	---	-----	---

Customer 004

10	^	32	^
11		33	
12	v	34	v

Customer 005

...	↔	...	↔	...	↔
-----	---	-----	---	-----	---

The field *value* contains the value of the fields filled by the user, with the following structure:

value1|value2|value3|...|valuen

The order of the items is the same as that of visualization and the value is present only for the valorized items.

value
11:31 11:32 11:34

JSFormValue

```
{
  "value": "11:31|11:32||11:34|",
  "dttime": "20150708102028"
}
```

The values can be retrieved from the object `JSFormValueExt`:

progressivo	keyval	val01
1	CUSTOMER - 001	11:31
2	CUSTOMER - 002	11:32
3	CUSTOMER - 004	11:34

key contains the key of the record of a specific table and it is composed by the following structure:

|<tablename>|-<reckey_record>|

val01 contains the value inserted by the user for the record.

JSFormValueExt

```
{
  "key": "|CUSTOMER|-|001|",
  "tabgen": {
    "tablename": "CUSTOMER",
    "key" : "001",
    ...
  }
  "val01": "11:31",
  ...
  "progressivo": 1
}

{
  "key": "|CUSTOMER|-|002|",
  "tabgen": {...},
  "val01": "11:32",
  ...
  "progressivo": 2
}

{
  "key": "|CUSTOMER|-|004|",
  "tabgen": {...},
  "val01": "11:34",
  ...
  "progressivo": 3
}
```


The field *value* contains all the data inserted by the user:

value
3 3,5 Hello 09/07/2015 10:32 ACTION:001 /path/to/photo/filename1.jpg;/path/to/photo/filename2.jpg 2 3 /TGK=001TGK//0/2 1,5 Good morning 13/07/2015 14:50 ACTION:003 /path/to/photo/filename3.jpg/ 1 4 /TGK=002TGK/

Following is the schema of the field value:

value_idacq1|value_idacq2|valore_idacq3|...|value_idacqn|idi1|idi2|idi3|...|idin

The order of the elements is the same as that of visualization and the value is only present for the selected option (*idi*) and for the inserted fields (*idacq*). The other fields are blank. The char “|” is used to separate every field even if the field has no value.

The key of the record has the following format:

/TGK=reckeyTGK/

The following sequence of characters is the separator for lookup records:

/0/

JSFormValue

```
{
  "value":
  "3|3,5|Hello|12/04/2015|12:35|ACTION:001|/path/to/photo/filename1.jpg;/path/to/photo/filename2.jpg||2|3||/TGK=001TGK//0/2|1,5|Good morning|13/07/2015|14:50|ACTION:003|/path/to/photo/filename3.jpg/|1|||4|/TGK=002TGK/",
  "dtttime": "20150709103450"
}
```

In order to read the data, you can use the object *JSFormValueExt*: here is a summary schema for every type of data:

type	progr.	key	val01	val02
integer	1	CUSTOMER - 001 - a - 22 - 1	3	
decimal	2	CUSTOMER - 001 - a - 22 - 2	3,5	
string	3	CUSTOMER - 001 - a - 22 - 3	Hello	
date	4	CUSTOMER - 001 - a - 22 - 4	12/04/2015	
time	5	CUSTOMER - 001 - q - 22 - 5	12:35	
lookup	6	CUSTOMER - 001 - a - 22 - 6	ACTION	001
photo	7	CUSTOMER - 001 - a - 22 - 7	/path/to/photo/filename1.jpg	
photo	8	CUSTOMER - 001 - a - 22 - 7	/path/to/photo/filename2.jpg	

item	9	CUSTOMER - 001 - i - 23 - 1		
item	10	CUSTOMER - 001 - i - 23 - 2		
integer	11	CUSTOMER - 002 - a - 22 - 1	2	
decimal	12	CUSTOMER - 002 - a - 22 - 2	1,5	
string	13	CUSTOMER - 002 - a - 22 - 3	Good morning	
date	14	CUSTOMER - 002 - a - 22 - 4	13/07/2015	
time	15	CUSTOMER - 002 - a - 22 - 5	14:50	
lookup	16	CUSTOMER - 002 - a - 22 - 6	ACTION	003
photo	17	CUSTOMER - 002 - a - 22 - 7	/path/to/photo/filename3.jpg	
item	18	CUSTOMER - 002 - i - 23 - 1		
item	19	CUSTOMER - 002 - i - 23 - 4		

The field *key* contains the key of the selected record. The key can be composed by different parts depending on the type of record: acquisition table or option group.

In the case of “option group”, *key* has a value composed by the key of the record and the key of the selected item (*idi*), while *val01* has no value. Below the sample code for the *key*:

```
|<tablename>|-|<reckey_record>|-|i|-|<idri>|-|<idi>|
```

In the case of “acquisition table”, *key* has a value composed by the key of the record and the key of the record of the acquisition table (*idacq*). *val01* contains the value inserted by the user.

```
|<tablename>|-|<reckey_record>|-|a|-|<idgracq>|-|<idacq>|
```

Below the `JSFormValueExt` objects for the above sample:

JSFormValueExt

Integer

```
{
  "key": "|CUSTOMER|-|001|-|a|-|22|-|1|",
  "tabgen": {
    "key": "001",
    "tablename": "CUSTOMER",
    "val01": "",
    ...
    "val20": ""
  },
  "val01": "3",
  ...
  "progressivo": 1
}
```

Decimal

```
{
```

```

    "key": "|CUSTOMER|-|001|-|a|-|22|-|2|",
    "tabgen": {...},
    "val01": "3,5",
    ...
    "progressivo": 2
}

```

String

```

{
    "key": "|CUSTOMER|-|001|-|a|-|22|-|3|",
    "tabgen": {...},
    "val01": "Hello",
    ...
    "progressivo": 3
}

```

Date

```

{
    "key": "|CUSTOMER|-|001|-|a|-|22|-|4|",
    "tabgen": {...},
    "val01": "12/05/2015",
    ...
    "progressivo": 4
}

```

Time (hours)

```

{
    "key": "|CUSTOMER|-|001|-|a|-|22|-|5|",
    "tabgen": {...},
    "val01": "12:35",
    ...
    "progressivo": 5
}

```

Lookup

```

{
    "key": "|CUSTOMER|-|001|-|a|-|22|-|6|",
    "tabgen": {...},
    "val01": "ACTION",
    "val02": "001",
    ...
    "progressivo": 6
}

```

Photo

N.B.: there is a `JSFormValueExt` for every photo.

```
{  
  "key": "|CUSTOMER|-|001|-|a|-|22|-|7|",  
  "tabgen": {...},  
  "val01": "/path/to/photo/filename1.jpg",  
  ...  
  "progressivo": 7  
}
```

Items

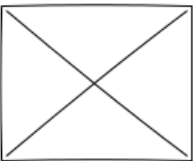

```
{  
  "key": "|CUSTOMER|-|001|-|i|-|23|-|1|",  
  "tabgen": {...},  
  "val01": "",  
  ...  
  "progressivo": 10  
}
```


Time (hours)

11	⬆	34	⬆
12		35	
13	⬇	36	⬇

Lookup

Photo

☐ ☐ +

Items

- ☒ # 1 choice
- ☒ # 2 choice
- ☐ # 3 choice
- ☐ # 4 choice
- ☐ # 5 choice



Time (hours)


13	⬆	49	⬆
14		50	
15	⬇	51	⬇

Lookup

▼

Photo


+

Items

- ☒ #1 choice
- ☐ #2 choice
- ☐ #3 choice
- ☒ #4 choice
- ☐ #5 choice

The field *value* contains all the data inserted by the user:

value
3 3,5 Hello 09/07/2015 10:32 ACTION:001 /path/to/photo/filename1.jpg;/path/to/photo/filename2.jpg 2 3 /TGK=001TGK//0/2 1,5 Good morning 13/07/2015 14:50 ACTION:003 /path/to/photo/filename3.jpg/ 1 4 /TGK=002TGK/

Following is the schema of the field value:

value_idacq1|value_idacq2|valore_idacq3|...|value_idacqn|idi1|idi2|idi3|...|idin

The order of the elements is the same as that of visualization and the value is only present for the selected option (*idi*) and for the inserted fields (*idacq*). The other fields are blank. The char “|” is used to separate every field even if the field has no value.

The key of the record has the following format:

/TGK=reckeyTGK/

The following sequence of characters is the separator for lookup records:

/0/

JSFormValue

```
{
  "value":
  "3|3,5|Hello|12/04/2015|12:35|ACTION:001|/path/to/photo/filename1.jpg;/path/to/photo/filename2.jpg||2|3||/TGK=001TGK//0/2|1,5|Good morning|13/07/2015|14:50|ACTION:003|/path/to/photo/filename3.jpg/|1|||4|/TGK=002TGK/",
  "dttime": "20150709103450"
}
```

In order to read the data, you can use the object *JSFormValueExt*: here is a summary schema for every type of data:

tipo	progr.	key	val01	val02
intero	1	CUSTOMER - 001 - a - 22 - 1	3	
decimale	2	CUSTOMER - 001 - a - 22 - 2	3,5	
stringa	3	CUSTOMER - 001 - a - 22 - 3	Hello	
data	4	CUSTOMER - 001 - a - 22 - 4	12/04/2015	
ora	5	CUSTOMER - 001 - q - 22 - 5	12:35	
lookup	6	CUSTOMER - 001 - a - 22 - 6	ACTION	001
foto	7	CUSTOMER - 001 - a - 22 - 7	/path/to/photo/filename1.jpg	
foto	8	CUSTOMER - 001 - a - 22 - 7	/path/to/photo/filename2.jpg	

item	9	CUSTOMER - 001 - i - 23 - 1		
item	10	CUSTOMER - 001 - i - 23 - 2		
intero	11	CUSTOMER - 002 - a - 22 - 1	2	
decimale	12	CUSTOMER - 002 - a - 22 - 2	1,5	
stringa	13	CUSTOMER - 002 - a - 22 - 3	Good morning	
data	14	CUSTOMER - 002 - a - 22 - 4	13/07/2015	
ora	15	CUSTOMER - 002 - a - 22 - 5	14:50	
lookup	16	CUSTOMER - 002 - a - 22 - 6	ACTION	003
foto	17	CUSTOMER - 002 - a - 22 - 7	/path/to/photo/filename3.jpg	
item	18	CUSTOMER - 002 - i - 23 - 1		
item	19	CUSTOMER - 002 - i - 23 - 4		

The field *key* contains the key of the selected record. The key can be composed by different parts depending on the type of record: acquisition table or option group.

In the case of “option group”, *key* has a value composed by the key of the record and the key of the selected item (*idi*), while *val01* has no value. Below the sample code for the *key*:

```
|<tablename>|-|<reckey_record>|-|i|-|<idri>|-|<idi>|
```

In the case of “acquisition table”, *key* has a value composed by the key of the record and the key of the record of the acquisition table (*idacq*). *val01* contains the value inserted by the user.

```
|<tablename>|-|<reckey_record>|-|a|-|<idgracq>|-|<idacq>|
```

Below the `JSFormValueExt` objects for the above sample:

JSFormValueExt

Integer

```
{
  "key": "|CUSTOMER|-|001|-|a|-|22|-|1|",
  "tabgen": {
    "key": "001",
    "tablename": "CUSTOMER",
    "val01": "",
    ...
    "val20": ""
  },
  "val01": "3",
  ...
  "progressivo": 1
}
```

Decimal

```
{
  "key": "|CUSTOMER|-|001|-|a|-|22|-|2|",
  "tabgen": {...},
  "val01": "3,5",
  ...
  "progressivo": 2
}
```

String

```
{
  "key": "|CUSTOMER|-|001|-|a|-|22|-|3|",
  "tabgen": {...},
  "val01": "Hello",
  ...
  "progressivo": 3
}
```

Date

```
{
  "key": "|CUSTOMER|-|001|-|a|-|22|-|4|",
  "tabgen": {...},
  "val01": "12/05/2015",
  ...
  "progressivo": 4
}
```

Time (hours)

```
{
  "key": "|CUSTOMER|-|001|-|a|-|22|-|5|",
  "tabgen": {...},
  "val01": "12:35",
  ...
  "progressivo": 5
}
```

Lookup

```
{
  "key": "|CUSTOMER|-|001|-|a|-|22|-|6|",
  "tabgen": {...},
  "val01": "ACTION",
  "val02": "001",
  ...
  "progressivo": 6
}
```

Photo

N.B.: there is a `JSFormValueExt` for every photo.

```
{
  "key": "|CUSTOMER|-|001|-|a|-|22|-|7|",
  "tabgen": {...},
  "val01": "/path/to/photo/filename1.jpg",
  ...
  "progressivo": 7
}
```

Items

```
{
  "key": "|CUSTOMER|-|001|-|i|-|23|-|1|",
  "tabgen": {...},
  "val01": "",
  ...
  "progressivo": 10
}
```

Calendar

APP

Capture from Table - Calendar

July 2015

	1	2	3	4	5
6	7	8	9	10	11
12	13	14	15	16	17
18	19	20	21	22	23
24	25	26	27	28	29
30	31				

Thu 16th July 2015

Customer 001 Meeting

☒ #3 choice

☒ #4 choice

Meeting with CTO and engineering technicians.

from 10 a.m. to 1 p.m.

The field *value* contains the key of the selected event.

value
12345

JSFormValue

```
{
  "value": "12345",
  "dtttime": "20150715144701"
}
```

JSFormValueExt

In order to read the data, you can use the object *JSFormValueExt*: there are different data depending by the progressive number.

In particular the record with **progressive=1** depends by the selected type event.

If the event has been defined by the client (free event) the structure is:

progr	key	val01	val02	val03	val04	val05	val06	val07
1	12345			Customer 001 Meeting	Meeting with CTO and engineering technicians.	1436914800000	1436918400000	#669900

The fields *val01* and *val02* have no value; the field *val03* contains the title field of the event, *val04* contains the note field, *val05* contains date and time of the start of the event, *val06* contains date and time of the end of the event (milliseconds); *val07* contains the event color.

```
{
  "key": "12345",
  "val01": "",
  "val02": "",
  "val03": "Customer 001 Meeting",
  "val04": "Meeting with CTO and engineering technicians.",
  "val05": "1436914800000",
  "val06": "1436918400000",
  "val07": "#669900",
  ...
  "progressivo": 1
}
```

If the event has been defined by the server the structure is:

progr	key	val01	val02	val03	val04	val05	val06	val07
1	12345	EVENTS	12345		Meeting with CTO and engineering technicians.	1436914800000	1436918400000	#669900

The fields *val01* e *val02* respectively contains the name of the table (*tablename*) and the key of the record (*reckey*) for the event previously generated on the server. Field *val03* has no value and the other fields contains data like with the free event.

```
{
  "val01": "EVENTS",
  "val02": "1436918400000",
  "val03": "",
  "val04": "Meeting with CTO and engineering technicians.",
  "val05": "1436914800000",
  "val06": "1436918400000",
  "val07": "#669900",
  ...
  "progressivo": 1,
  "key": "12345"
}
```

The record with **progressive=2** contains the data of the selected record of table with the following structure:

progr	key	val01	val02	val03	val04	val05	val06	val07
2	12345	CUSTOMER - 001						

Field *val01* contains the name of the table (*tablename*) and the key (*reckey*) of the selected record.

```
{
  "key": "12345",
  "val01": "|CUSTOMER|-|001|",
  ...
  "progressivo": 2
}
```

Finally, if an option group is present, a record with **progressive=3** is generated and it has the following structure:

progr	key	val01	val02	val03	val04	val05	val06	val07
3	12345	3 4	12					

Field *val01* contains the list of selected items (*idi*) and field *val02* contains the id of the option group (*idri*).

```
{
  "key": "1436963929561",
  "val01": "3|4",
  "val02": "12",
  ...
  "progressivo": 3
}
```


Payment

APP

Capture from Table - Payment

Customer 001

Total amount: 456,78

**** * 1234

You can check if the payment succeeded by checking the value of the field *value*.

Android

value
190

iOS

value
0

WP: payment not available

JSFormValue

```
{
  "value": "0",
  "dtttime": "20150708102224"
}
```

The object JSFormValueExt contains the payment details (N.B.: the values differs depending on the platform). The field key contains the name of the table (tablename) and the key of the record (reckey_record) like in the following sample code:

|<tablename>|-|<reckey_record>|

Android

progr	key	val01	val02
1	CUSTOMER - 001	result	190
2	CUSTOMER - 001	com.setefi.pos.external.pay.cvm	2
3	CUSTOMER - 001	com.setefi.pos.external.pay.pan	*****6314
4	CUSTOMER - 001	com.setefi.pos.external.pay.acquirer_id	4
5	CUSTOMER - 001	com.setefi.pos.external.pay.exp	****
6	CUSTOMER - 001	com.setefi.pos.external.pay.termid	97701158
7	CUSTOMER - 001	com.setefi.pos.external.pay.operation_number	47
8	CUSTOMER - 001	com.setefi.pos.external.pay.authoritation_number	862960
9	CUSTOMER - 001	com.setefi.pos.external.pay.timestamp	2007151430
10	CUSTOMER - 001	com.setefi.pos.external.pay.stan	6
11	CUSTOMER - 001	com.setefi.pos.external.pay.amount	01.00.00
12	CUSTOMER - 001	com.setefi.pos.external.pay.function	500
13	CUSTOMER - 001	com.setefi.pos.external.pay.trans_type	CLI
14	CUSTOMER - 001	com.setefi.pos.external.pay.merchant_id	303186300140013

iOS

progr	key	val01	val02
1	CUSTOMER - 001	acquirer_id	00000000004
2	CUSTOMER - 001	exp	
3	CUSTOMER - 001	amount	1,00
4	CUSTOMER - 001	pan	*****6314
5	CUSTOMER - 001	stan	000004
6	CUSTOMER - 001	timestamp	1507151448
7	CUSTOMER - 001	authorization_number	432022
8	CUSTOMER - 001	cvm	0
9	CUSTOMER - 001	operation_number	000045
10	CUSTOMER - 001	returnCode	0
11	CUSTOMER - 001	trans_type	CLI
12	CUSTOMER - 001	merchant_id	303186300140013

13	CUSTOMER - 001	termid	97701158
----	----------------	--------	----------

WP: payment not available

The object `JSFormValueExt` can assume different values depending on the platform: *key* is always blank, *val01* contains the name of the payment parameter, *val02* the value of the payment parameter. The field *tabgen* contains the data for the lookup.

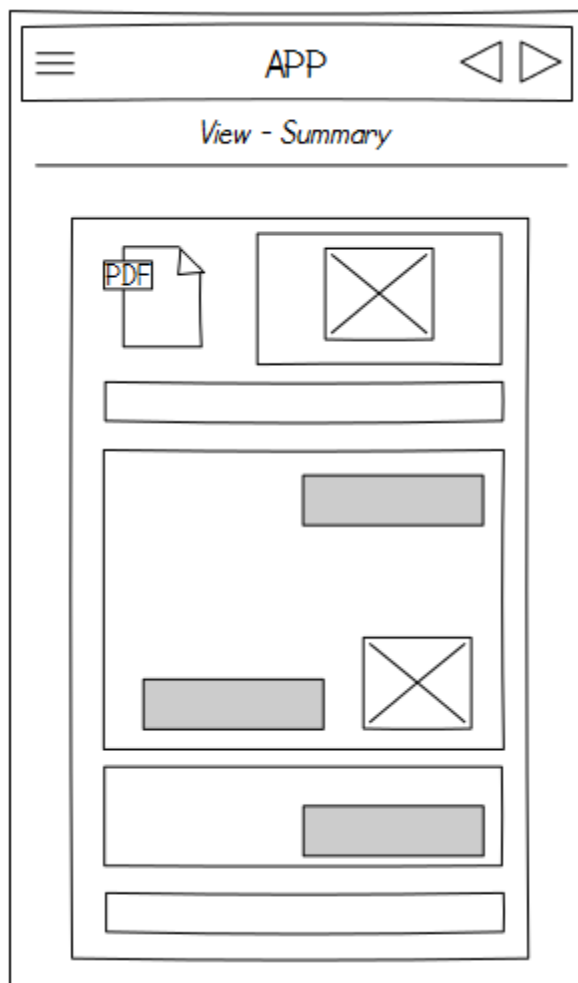
The following is an example:

JSFormValueExt

```
{
  "key": "",
  "tabgen": {
    "tabname": "PAGAMENTO",
    "key": "001",
    ...
  },
  "val01": "acquirer_id",
  "val02": "000000000004",
  ...
  "progressivo": 1
}
```

View

Summary



The field *value* contains the path of the summary PDF:

value
/path/to/file/filename.pdf

JSFormValue

```
{
  "value": "/path/to/file/filename.pdf",
  "dtttime": "20150716102544"
}
```

In order to get the value of the path you can use the object `JSFormValueExt`:

progressivo	key	val01
1		/path/to/file/filename.pdf

JSFormValueExt

```
{  
  "key": "",  
  "val01": "/path/to/file/filename.pdf",  
  ...  
  "progressivo": 1  
}
```