

EPSON

111-38-URM-009

KDS API

User Manual

EPSON	KDS API User Manual Page 1 of 26	111-38-URM-009 R2.30
--------------	--	-------------------------

Table of Contents

1. Introduction.....	3
2. API Interface.....	3
3. API and POS Parser.....	4
4. Typical Order Receipt.....	4
4.1. JSON Schema	5
4.1.1 JSON Schema detailed description	5
4.2. Sample JSON payload	8
4.3. Response errorcode list.....	9
5. Order Types.....	10
5.1. “new” Order POS Request	10
5.2. “new” Order KDS Response	11
5.3. “append” Order KDS Response	11
5.4. “append” Order KDS response	11
5.5. “void” Order POS request.....	12
5.6. “void” Order KDS response	12
5.7. “status” Order POS request.....	12
5.8. “status” Order KDS Response	14
6. Callback Registration	17
6.1. “callback” POS Request	17
6.2. “callback” KDS Response.....	18
7. Callback Notifications	19
7.1. “onenter” Callback	19
7.2. “onbump” Callback.....	22
7.3. “onpriority” Callback.....	22
7.4. “onrush” Callback.....	23
7.5. “onunbump” Callback.....	23
7.6. “onrecall” Callback	23
7.7. POS/Server Response	23
8. Status of Callback.....	24
8.1. POS Request.....	24
8.2. KDS Response	24
9. Callback behavior in multistation mode	25

1. Introduction

The Epson KDS traditionally parsed incoming “print” jobs from a Point of Sale System (POS) to an Epson POS printer. Each POS system that the Epson KDS supports requires a “parser” module to understand the format of the print job. This parser module is configured using the KDS Configuration Utility and the format must be followed from then on.

This method has advantages in that it is very easy to have a working KDS system out of the box for POS systems that are already supported. There is no integration with the POS required. For a POS that is not supported it is not difficult for Epson to create a new parser module – typically within 1-2 weeks.

Some disadvantages are that the POS may not have the flexibility to deliver all types of orders the Epson KDS is capable of handling - i.e. Void orders or order Appends. Also, as the data is a print job there is no mechanism to provide feedback to the POS about the status of any given order.

This document outlines a specification for an “API” or Application Programming Interface to the Epson KDS system. The API provides an alternative direct way of communicating with the Epson KDS that not only allows data to be sent to the Epson KDS but also for KDS data to be returned. This allows for better control and feedback from the Epson KDS system.

2. API Interface

Data is exchanged between the POS and KDS using a JSON format which is described later in this document. The communication between POS and KDS is done using HTTP port 80:

<http://192.168.192.168/cgi-bin/kdsapi/service.cgi>

HTTP POST (MAIN DELIVERY OPTION) – JSON data will be the body of HTTP POST request and response, using

`Content-Type: application/json.`

EPSON	KDS API User Manual Page 3 of 26	111-38-URM-009 R2.30
--------------	--	-------------------------

3. API and POS Parser

Traditionally the Epson KDS allows a parser to be configured in order for the KDS to understand the particular POS data. A POS parser can still be selected and used in conjunction with the KDS API to provide flexibility for customers who wish to have multiple inputs to the POS system.

The POS parser would still listen to the traditional port inputs -> Serial, USB, or Ethernet. The KDS API service is listening on port 80.

A POS parser needs to be selected when configuring the KDS. If a traditional POS parser is not required there will be a "KDS API" parser that can be selected that will allow the system to be configured.

4. Typical Order Receipt

A typical order receipt (printout) will look like the following:

```
Server William
Table 7      CK# 0012
-----
1  Steak Burger    3
1  Fries           3
    No salt
1  Coke           3
    Regular
    No ice
```

It can be broken down into the following basic fields:

- Server Name
- Table Number
- Check Number
- Item Qty
- Item Description
- Seat Number
- Modifier(s)

The Epson KDS is capable of showing other miscellaneous information within the order such as course info, order types, customer name, customer phone number, customer email etc. This is described elsewhere within this specification.

EPSON	KDS API User Manual Page 4 of 26	111-38-URM-009 R2.30
--------------	--	-------------------------

4.1. JSON Schema

The interface to the KDS API will be based on the following JSON packet schema:

```
{
  "seq" : number,
  "type" : "string",
  "server" : "string",
  "table" : "string",
  "check" : "string",
  "cust_name" : "string",
  "cust_phone" : "string",
  "cust_email" : "string",
  "itemlist" :
  [
    {
      "itemid" : number,
      "item" : "string",
      "header" : "string",
      "label" : "string",
      "qty" : number,
      "seat" : number,
      "modifierlist" :
      [
        {
          "modifier" : "string",
          "colour" : "string"
        }
      ]
    }
  ]
}
```

4.1.1 JSON Schema detailed description

- "seq" – JSON payload sequence number. Should be incremented by 1 for every API call to the Epson KDS. Range 0 – 1048575. Once the upper limit is reached it should roll back to 0.
- "type" – Order type. Can be "new", "void", "append", "status", "callback". Each order type is described later in this document.
- "server" – Server name as it would be displayed in the KDS order header. It could be replaced with other text such as "Pickup", or "Delivery". While 0-40 ASCII characters are accepted the server name should be kept small so as to be displayed fully in the order header.

EPSON	KDS API User Manual Page 5 of 26	111-38-URM-009 R2.30
--------------	--	-------------------------

- "table" – Table number as displayed in the KDS order header. 0-20 ASCII characters are accepted however the table number should be kept small so as to be displayed fully in the order header.
- "check" – Check number for the order as displayed in the KDS order header. This check number should be unique for every new order entered into the system. 0-20 ASCII characters are accepted however the check number should be kept small so as to be displayed fully in the order header.
- "cust_name" – Optional field indicating customer name. 0-40 ASCII characters are accepted.
- "cust_phone" – Optional field indicating customer phone number. Must be in one of the following formats:
 (555)555-5555
 555-555-5555
 5555555555
- "cust_email" – Optional field indicating customer e-mail address. 0-40 ASCII characters are accepted.

The "cust_name", "cust_phone", "cust_email" are all optional and can be included or not in any combination.

Only 1 of the following 3 fields (*item*, *header*, *label*) should be sent:

- "item" – Description of the item as displayed on the Epson KDS. Eg. "Hamburger". 0-40 ASCII characters.
- "header" – Any information that should be displayed at the top of the order (below the KDS Order Header). Eg. "Dine-In, Take-Out. There can be multiple header fields but each should correspond to its own itemid. 0-40 ASCII characters.
- "label" – Any miscellaneous information to be displayed within the order. Eg. "Course 1". 0-40 ASCII characters.
- "itemid" – The unique identifier assigned to an item by the POS. The itemid needs to be unique within an order and is used by the KDS to easily associate to an item. The best practice is for the first item in the list to have an id of 1 and then increase for every item. Numbers 1-99 are supported.

EPSON	KDS API User Manual Page 6 of 26	111-38-URM-009 R2.30
--------------	--	-------------------------

The following fields (qty and seat) are optional:

- "qty" – Quantity of the item as displayed on the KDS. Numbers 1-99 are supported.
- "seat" – Seat number for the respective item. This field is optional. If it is not there then no seat # will be displayed. Numbers 1-99 are supported.

If "qty" field is sent and "seat" is not then the Epson KDS should be configured to show Quantity.

If "seat" field is sent and "qty" is not then the Epson KDS should be configured to show Seat.

If both "qty" and "seat" fields are sent then the Epson KDS should be configured to show Seat. The Quantity will be prepended to the Item Description.

If neither "qty" or "seat" fields are sent then the Epson KDS will only show the item description regardless of the configuration.

- "modifier" – Description of the modifier as displayed on the Epson KDS. 0-40 ASCII characters.
- "colour" – can be "normal" (default) or "alert" to show the text in the alert color.

EPSON	KDS API User Manual Page 7 of 26	111-38-URM-009 R2.30
--------------	--	-------------------------

4.2. Sample JSON payload

```
{
  "seq" : 1,
  "type" : "new",
  "server" : "William",
  "table" : "7",
  "check" : "12",
  "cust_name" : "Robert",
  "cust_phone" : "5555555555",
  "cust_email" : "Robert@Robert.com",
  "itemlist" :
  [
    {
      "itemid" : 1,
      "qty" : 1,
      "seat" : "3",
      "item" : "Steak Burger"
    },
    {
      "itemid" : 2,
      "qty" : 1,
      "seat" : "3",
      "item" : "Fries",
      "modifierlist" :
      [
        {
          "modifier" : "No salt",
          "colour" : "alert"
        }
      ]
    },
    {
      "itemid" : 3,
      "qty" : 1,
      "seat" : "3",
      "item" : "Coke",
      "modifierlist" :
      [
        {
          "modifier" : "Regular"
        },
        {
          "modifier" : "No ice"
        }
      ]
    }
  ]
}
```

EPSON	KDS API User Manual Page 8 of 26	111-38-URM-009 R2.30
--------------	--	-------------------------

4.3. Response errorcode list

The KDS will send back a response indicating success or failure. If success,

```
{  
  "errorcode" : 0,  
  "description" : "success"  
}
```

If failure, an errorcode in the below list will be returned:

- 1 – json syntax error
- 2 – callback entry does not exist
- 3 – no callbacks are registered
- 4 – order does not exist
- 5 – invalid 'callbackid'
- 6 – invalid callback 'trigger'
- 7 – invalid callback 'url'
- 8 – reached maximum order number
- 9 – invalid json parameter
- 10 – missing json parameter
- 101 – data format error
- 104 – timeout, no response from kds

5. Order Types

5.1. "new" Order POS Request

Request for a brand new order to be sent to the KDS:

```
{
  "seq" : 1,
  "type" : "new",
  "server" : "William",
  "table" : "7",
  "check" : "12",
  "cust_name" : "Robert",
  "cust_phone" : "5555555555",
  "cust_email" : "Robert@Robert.com",
  "itemlist" :
  [
    {
      "itemid" : 1,      Unique positive integer number within the order context
      "header" : "Dine-In"      Will be displayed only once on screen
    },
    {
      "itemid" : 2,
      "header" : "Address: 185 Renfrew Dr."
    },
    {
      "itemid" : 3,
      "label" : "--- Course 1 ---"  ## will be displayed as a special not
                                   selectable item on the screen
                                   multiple times
    },
    {
      "itemid" : 4,
      "qty" : 1,
      "seat" : "3",
      "item" : "Steak Burger"
    },
    {
      "itemid" : 5,
      "qty" : 1,
      "seat" : "3",
      "item" : "Fries",
      "modifierlist" :
      [
        { "modifier" : "No salt", "colour" : "alert" }
      ]
    },
    {
      "itemid" : 6,
      "label" : "--- Course 2 ---"
    },
    {
      "itemid" : 7,
      "qty" : 1,
      "seat" : "3",
      "item" : "Coke",
      "modifierlist" :
      [
        { "modifier" : "Regular" },
        { "modifier" : "No ice" }
      ]
    }
  ]
}
```

EPSON	KDS API User Manual Page 10 of 26	111-38-URM-009 R2.30
--------------	---	-------------------------

5.2. “new” Order KDS Response

KDS will send back a response indicating success or failure. Response errorcodes are listed in Section 4.3.

5.3. “append” Order KDS Response

Request for an addition to be sent to an already existing order. The check number should match a previous order:

```
{
  "seq" : 2,
  "type" : "append",
  "server" : "William",
  "table" : "7",
  "check" : "12"
  "itemlist" :
  [
    {
      "itemid" : 8,      Unique positive number within an order
      "qty" : 1,
      "seat" : "2",
      "item" : "Classic Burger",
      "modifierlist" :
      [
        { "modifier" : "No pickle" }
      ]
    }
  ]
}
```

The difference between “new” and “append” order is that appended orders will show the quantity column in a different background color on the KDS display to indicate the order was appended.

5.4. “append” Order KDS response

KDS will send back a response indicating success or failure. Response errorcodes are listed in Section 4.3.

EPSON	KDS API User Manual Page 11 of 26	111-38-URM-009 R2.30
--------------	---	-------------------------

5.5. “void” Order POS request

Voiding the order will be facilitated with ‘itemid’ field to un-ambiguously find the exact item within the order.

When voiding “Fries” and “Coke” from above order, POS will send

```
{
  "seq" : 3,
  "type" : "void",
  "server" : "William",
  "table" : "7",
  "check" : "12",
  "itemlist" :
  [
    { "itemid" : 5 },      Fries
    { "itemid" : 6 }      Coke
  ]
}
```

To void the whole order, POS will send request without item list.

```
{
  "seq" : 3,
  "type" : "void",
  "server" : "William",
  "table" : "7",
  "check" : "12"
}
```

5.6. “void” Order KDS response

KDS will send back a response indicating success or failure. Response errorcodes are listed in Section 4.3.

5.7. “status” Order POS request

Status of particular order(s) can be obtained from the KDS system by sending a status order request.

EPSON	KDS API User Manual Page 12 of 26	111-38-URM-009 R2.30
--------------	---	-------------------------

5.7.1 Single Order

For a single order,

```
{
  "seq" : 4,
  "type" : "status",
  "statusof" : "ordersonly",
  "orderlist" :
  [
    {
      "table" : "7",
      "check" : "12"
    }
  ]
}
```

5.7.2 Multiple Orders

For multiple orders, orderlist will contain multiple check numbers:

```
{
  "seq" : 5,
  "type" : "status",
  "statusof" : "ordersonly",
  "orderlist" :
  [
    {
      "table" : "Table 7",
      "check" : "12"
    },
    {
      "table" : "Table 8",
      "check" : "14"
    },
    {
      "table" : "Table 9",
      "check" : "15"
    }
  ]
}
```

If orderlist is missing, then it implies all active orders and they will all be returned.

EPSON	KDS API User Manual Page 13 of 26	111-38-URM-009 R2.30
--------------	---	-------------------------

The parameter `statusof` can be:

- `"ordersonly"` – for the status of orders, and the counts of active, bumped, voided items.
- `"ordersanditems"` – same as `"ordersonly"` (ie. order status and item counts), plus the status of all their items.

5.8. "status" Order KDS Response

There are some extra fields in the Status Response:

`state` of order can be:

- `"active"` – still cooking. This means, at least one item is `"active"`.
- `"bumped"` – the order has been completely bumped.
- `"voided"` – the order has been completely voided

Similarly, the `state` of item can be:

- `"active"` – still cooking
- `"bumped"` – the item has been bumped
- `"voided"` – the item has been voided

For `"active"` orders only, `timeout` of order will be one of

- `"normal"` – normal timer
- `"priority"` – the timer reached Priority Time
- `"rush"` – the timer reached Rush Time

If the order is `"active"`, then `elapsedtime` is the number of second since the order was created, and it would have different value when the next POS request is made. If the order is `"bumped"` or `"voided"`, then the `elapsedtime` is the number of second it took for the order to be bumped or voided, and it would remain the same for subsequent POS requests.

EPSON	KDS API User Manual Page 14 of 26	111-38-URM-009 R2.30
--------------	---	-------------------------

There will be 4 item counts:

- "activeitemcount" – the number of active items
- "bumpeditemcount" – the number of bumped items
- "voideditemcount" – the number of voided items
- "totalitemcount" – the number of items in total, which is sum of above counts.

5.8.1 "ordersonly" Status Response

If the request was "ordersonly", then KDS will return order status and item counts.

```
{
  "errorcode" : 0,
  "description" : "success",
  "orderlist" :
  [
    {
      "check" : "15",
      "table" : "Table 9",
      "server" : "William",
      "cust_name" : "Robert",
      "cust_phone" : "5555555555",
      "cust_email" : "Robert@Robert.com",
      "state" : "active",
      "elapsedtime" : 120,
      "timeout" : "normal",
      "activeitemcount" : 2,
      "bumpeditemcount" : 1,
      "voideditemcount" : 1,
      "totalitemcount" : 4
    }
  ]
}
```

EPSON	KDS API User Manual Page 15 of 26	111-38-URM-009 R2.30
--------------	---	-------------------------

5.8.2 “ordersanditems” Status Response

If the request was “ordersanditems”, then KDS will return order status, item counts, and item status.

```
{
  "errorcode" : 0,
  "description" : "success",
  "orderlist" :
  [
    {
      "check" : "15",
      "table" : "Table 9",
      "server" : "William",
      "cust_name" : "Robert",
      "cust_phone" : "5555555555",
      "cust_email" : "Robert@Robert.com",
      "state" : "active",
      "elapsedtime" : 120,
      "timeout" : "normal",

      "activeitemcount" : 2,
      "bumpeditemcount" : 1,
      "voideditemcount" : 1,
      "totalitemcount" : 4

      "itemlist" :
      [
        {
          "itemid" : 2,
          "state" : "active",
          "qty" : 1,
          "seat" : "2",
          "item" : "Classic Burger",
          "modifierlist" :
          [
            { "modifier" : "No pickle" }
          ]
        },
        {
          "itemid" : 3,
          "state" : "bumped"
          "qty" : 1,
          "seat" : "3",
          "item" : "Fries",
        },
        {
          "itemid" : 4,
          "state" : "voided"
          "qty" : 1,
          "seat" : "4",
          "item" : "Coke",
        },
        {
          "itemid" : 5,
          "state" : "active"
          "qty" : 1,
          "seat" : "1",
          "item" : "Classic Burger",
        }
      ]
    }
  ]
}
```

EPSON	KDS API User Manual Page 16 of 26	111-38-URM-009 R2.30
--------------	---	-------------------------

5.8.3 Status Error Response

KDS will send back a response indicating success or failure. Response errorcodes are listed in Section 4.3.

6. Callback Registration

Callbacks can be registered to have the KDS send information to the server for various events:

6.1. “callback” POS Request

```
{
  "seq" : 11,
  "type" : "callback",
  "callbackid" : 1,
  "action" : "set",
  "url" :
  "http://192.168.1.100:8080/kdsapi/callback_listener.cgi",
  "trigger" : "onbump"
}
```

where

- url can be standard HTTP POST, like
 - "http://192.168.1.100:8080/kdsapi/callback_listener.cgi"
- callbackid is unique identifier assigned to callback entry by POS. This identifier will be used to delete the entry later. Should start at 1 and increment by 1 for each new call back registration.
- action can be
 - "set" – to create or overwrite a callback entry,
 - "clear" – to delete the entry,
- trigger is the condition that causes KDS to call back. It can be
 - "onenter" – KDS will return both check, table and itemid(s) when an order is entered.
 - "onbump" – KDS will return both check, table and itemid(s) when a bump takes place.
 - "onpriority" – when the timer of order reaches Priority Time, KDS will return check, table.

EPSON	KDS API User Manual Page 17 of 26	111-38-URM-009 R2.30
--------------	---	-------------------------

- "onrush" – when the timer of order reaches Rush Time, KDS will return check, table.
- "onunbump" – KDS will return both check, table and itemid when an item is unbumped.
- "onrecall" – KDS will return both check, table when an order recall takes place.

6.2. "callback" KDS Response

KDS will send back a response indicating success or failure. Response errorcodes are listed in Section 4.3.

EPSON	KDS API User Manual Page 18 of 26	111-38-URM-009 R2.30
--------------	---	-------------------------

7. Callback Notifications

7.1. “onenter” Callback

The “new”, “append”, “void” requests will all generate “onenter” callback, with type “new”, “append”, “void” separately. If the trigger is “onenter” then the KDS will send the original print job to the callback URL in the following JSON format. Optional “cust_name”, “cust_phone” and “cust_email” will be included if these fields were included in “new” order request.

- **Callback for “new”**

```
{
  "type" : "new",
  "check" : "0025",
  "table" : "Table 15",
  "server" : "Mark",
  "cust_name" : "Robert",
  "cust_phone" : "5555555555",
  "cust_email" : "Robert@Robert.com",

  "itemlist" :
  [
    {
      "itemid" : 1,
      "qty" : 1,
      "seat" : 3,
      "item" : "Meat Lover",
      "modifierlist" :
      [
        {
          "modifier" : "> Olives",
          "color" : "alert"
        },
        {
          "modifier" : "> NO Mushrooms"
        }
      ]
    },
    {
      "itemid" : 2,
      "label" : "- Also sent to:Station 1"
    },
    {
      "itemid" : 3,
      "label" : "- Also sent to:Pizza Oven"
    }
  ]
}
```

EPSON	KDS API User Manual Page 19 of 26	111-38-URM-009 R2.30
--------------	---	-------------------------

- **Callback for “append”**

```
{
  "type" : "append",
  "check" : "0025",
  "table" : "Table 15",
  "server" : "Mark",
  "cust_name" : "Robert",
  "cust_phone" : "5555555555",
  "cust_email" : "Robert@Robert.com",
  "itemlist" :
  [
    {
      "itemid" : 8,
      "qty" : 1,
      "seat" : "2",
      "item" : "Classic Burger",
      "modifierlist" :
      [
        { "modifier" : "No pickle" }
      ]
    }
  ]
}
```

- **Callback for “void”**

Callback generated by void item 1 of above example will be:

```
{
  "type" : "void",
  "check" : "0025",
  "table" : "Table 15",
  "server" : "Mark",
  "cust_name" : "Robert",
  "cust_phone" : "5555555555",
  "cust_email" : "Robert@Robert.com",
  "itemlist" :
  [
    {
      "itemid" : 1,
      "qty" : 1,
      "seat" : 3,
      "item" : "Meat Lover",
      "modifierlist" :
      [
        {
          "modifier" : "> Olives",
          "color" : "alert"
        },
        {
          "modifier" : "> NO Mushrooms"
        }
      ]
    }
  ]
}
```

Callback generated by void the whole order will not contain any item:

```
{
  "type" : "void",
  "check" : "0025",
  "table" : "Table 15",
  "server" : "Mark",
  "cust_name" : "Robert",
  "cust_phone" : "5555555555",
  "cust_email" : Robert@Robert.com
}
```

7.2. “onbump” Callback

If the trigger is “onbump” and an order is bumped, then KDS will send `check`, `table` of the bumped order. Optional “cust_name”, “cust_phone” and “cust_email” will be included if these fields were included in “new” order request.

```
{
  "type" : "callback",
  "callbackid" : 1,
  "trigger" : "onbump",
  "check" : "12",
  "table" : "Table 7"
  "cust_name" : "Robert",
  "cust_phone" : "5555555555",
  "cust_email" : "Robert@Robert.com"
}
```

If the trigger is “onbump” and an item is bumped (on SCROLL mode), then KDS will send the same content as order bump, together with `itemid`, `itemname` and `modifiers` for this item.

```
{
  "type" : "callback",
  "callbackid" : 1,
  "trigger" : "onbump",
  "check" : "12",
  "table" : "Table 7",
  "cust_name" : "Robert",
  "cust_phone" : "5555555555",
  "cust_email" : "Robert@Robert.com",
  "itemid" : 8
  "item" : "Classic Burger",
  "modifierlist" :
  [
    { "modifier" : "No pickle" }
  ]
}
```

7.3. “onpriority” Callback

If the trigger is “onpriority” or “onrush”, and an order becomes Priority or Rush, respectively, then KDS will send `check`, `table` of that order.

```
{
  "type" : "callback",
  "callbackid" : 2,
  "trigger" : "onpriority",
  "check" : "12",
  "table" : "Table 7"
}
```

EPSON	KDS API User Manual Page 22 of 26	111-38-URM-009 R2.30
--------------	---	-------------------------

7.4. “onrush” Callback

```
{
  "type" : "callback",
  "callbackid" : 3,
  "trigger" : "onrush",
  "check" : "12",
  "table" : "Table 7"
}
```

7.5. “onunbump” Callback

If the trigger is “onunbump” and an order item is unbumped (on SCROLL mode or directly unbump from touch screen), then KDS will send `check`, `table` of the bumped order and the bumped item id, itemname and modifiers.

```
{
  "type" : "callback",
  "callbackid" : 1,
  "trigger" : "onunbump",
  "check" : "12",
  "table" : "Table 7",
  "cust_name" : "Robert",
  "cust_phone" : "5555555555",
  "cust_email" : "Robert@Robert.com",
  "itemid" : 8,
  "item" : "Classic Burger",
  "modifierlist" :
  [
    { "modifier" : "No pickle" }
  ]
}
```

7.6. “onrecall” Callback

If the trigger is “onrecall” and an order is recalled, then KDS will send `check`, `table` of the bumped order.

```
{
  "type" : "callback",
  "callbackid" : 1,
  "trigger" : "onrecall",
  "check" : "12",
  "table" : "Table 7",
  "cust_name" : "Robert",
  "cust_phone" : "5555555555",
  "cust_email" : "Robert@Robert.com"
}
```

7.7. POS/Server Response

The Epson KDS ignores the response from the POS or Server.

EPSON	KDS API User Manual Page 23 of 26	111-38-URM-009 R2.30
--------------	---	-------------------------

8. Status of Callback

8.1. POS Request

```
{
  "type" : "status",
  "seg" : 12,
  "statusof" : "callback",
  "callbacklist" :
  [
    { "callbackid" : 1 },
    { "callbackid" : 2 },
    { "callbackid" : 3 }
  ]
}
```

If `callbacklist` is missing or empty, then it implies all existing callbacks.

8.2. KDS Response

```
{
  "errorcode" : 0,
  "description" : "success",
  "callbacklist" :
  [
    {
      "callbackid" : 1,
      "url" :
      "http://192.168.1.100:8080/kdsapi/callback_listener.cgi",
      "trigger" : "onbump"
    },
    {
      "callbackid" : 2,
      "url" :
      "http://192.168.1.100:8080/kdsapi/callback_listener1.cgi",
      "trigger" : "onpriority"
    }
    {
      "callbackid" : 3,
      "url" : "",
      "trigger" : ""
    }
  ]
}
```

If there is an error, then `errorcode` will be non-zero with appropriate `description`. List of error conditions are in Section 4.3.

EPSON	KDS API User Manual Page 24 of 26	111-38-URM-009 R2.30
--------------	---	-------------------------

9. Callback behavior in multistation mode

In single station mode the behavior here is simple: when a certain action occurs, and if the corresponding callback is registered, then the KDS station will send the callback to the registered URL.

For multi-station mode, each KDS device will generate callbacks for actions related to its own display. However, the Master device (generally a TM-i printer) can generate callbacks for itself, as well as for certain callbacks on behalf of other displays too. This behavior is described in the table below:

	Slave Device	Master Device		
		POS Routing With Display	KDS Routing With Display	KDS/POS Routing Without Display
Onenter	<ul style="list-style-type: none"> - New order: Trigger “new” onenter callback on the device if this order will be displayed - Append: Trigger “append” onenter callback on the device if this append will happen on it - Void: Trigger “void” onenter callback on the device if this void will happen on it 	Same as Slave	<ul style="list-style-type: none"> - New order: All new order coming to the whole system trigger master “new” onenter callback - Append: All append action coming to the whole system trigger “append” onenter callback - Void: All void action coming to the whole system trigger “void” onenter callback 	Same as KDS Routing With Display
Onpriority	Triggered by orders displayed on its screen hitting priority time	Same as Slave	Same as Slave	Not generating
Onrush	Triggered by orders displayed on its screen hitting rush time	Same as Slave	Same as Slave	Not generating
Onbump (order)	Triggered by order bumped from itself	Same as Slave	Triggered by all order bump from all devices	Same as KDS Routing With Display
Onbump (item)	Triggered by item bumped from itself	Same as Slave	Triggered by all item bump from all devices	Same as KDS Routing With Display
Onunbump (item) (New callback)	Triggered by item unbumped from itself	Same as Slave	Triggered by all item unbump from all devices	Same as KDS Routing With Display
Onrecall (order) (New callback)	Triggered by item recalled from itself	Same as Slave	Same as Slave	Not generating

This page is purposely left blank.

EPSON	KDS API User Manual Page 26 of 26	111-38-URM-009 R2.30
--------------	---	-------------------------