

External Service Interface

Best Practice Recommendations
Version 2.2

Version	Changes	Responsible	Date
<hr/>			
1.0	Creation	TW	2018-08-29
2.0	Additions: creation of customer and tickets, ticket sale	TW	2018-09-19
2.1	Additions: Vat Codes	TW	2018-09-27
2.2	Cancel Booking	TW	2019-02-18

Table of Contents

1	Introduction.....	4
1.1	Interfacing F&B-Systems.....	4
1.2	Interfacing webshops.....	4
2	Interface methods.....	4
2.1	Catalog request	4
2.2	Get media details.....	5
2.2.1	Example 1 (day guest).....	6
2.2.2	Example 2 (value card).....	7
2.2.3	Calculation of available credit.....	7
2.2.4	Calculation of available advance payments	8
2.2.5	Example 3 (value card with discount).....	8
2.3	Create Customers.....	8
2.4	Create Tickets / Voucher	10
2.5	Sale Request	10
2.6	Cancellation Requests.....	12
2.6.1	Cancel Receipt	12
2.6.2	Cancel Receipt Position	12
2.7	Receipt information.....	14

1 Introduction

This document describes best practice recommendations to use the HKS External Service Interface. Please read at first the External Service Documentation. (e.g. <http://192.168.133.129:9006/metadata>)

For test purposes, a swagger UI is available. (e.g. <http://192.168.133.129:9006/swagger-ui/>)

1.1 Interfacing F&B-Systems

Required functions:

- Catalog request (get the articles to send in sale requests)
- Query media details (get credits, advance payments, discounts, benefits, customer properties)
- Sale request
- Cancel request

1.2 Interfacing webshops

Required functions:

- Catalog request (get the articles to send in Sale Requests)
- Sale request
- Cancel request

Optional functions:

- Create media (if the webshop sells Tickets / Vouchers)
- Create customers (if the tickets are to be personalized)
- Query media details (get credits, advance payments, discounts, benefits, customer properties)

2 Interface methods

2.1 Catalog request

GET /catalog

➔ receive a list of available sale articles

There is an optional Parameter "DeviceID". This parameter can be used in case of serving several tenants / sale outlets / cash desks over the same interface. (Only the articles for the specified device will be delivered) DeviceIDs have to be configured in the HKS-System.

Please note the following properties:

- **Number:** article number in HKS system. Use this number for sale requests
- **ShortText:** shorttext (max 30 characters)
- **VATSplitting:** possible Values:
 - o "0=100" (vat code 0 on 100% of the turnover)
 - o "1=100" (vat code 1 on 100% of the turnover)
 - o "2=100" (vat code 2 on 100% of the turnover).

The Vat values to the Vat Codes are declared on top of the catalog response.

E.G: Germany 2018: 0 = 0%, 1= 7%, 2=19%

When a tenant operates the F&B System, the HKS-System only collects the debt. In this case VatSplitting is "0=100"

- **PriceType:** (0 = zero price 1 = fixed Price, 3 = variable price)
- **UnitPrice:** list price

Response Sample:

```
{
  "CurrencyCode":978,
  "VATSplittingType":0,
  "TaxRates":[
    {
      "Id":0,
      "Rate":0
    },
    {
      "Id":1,
      "Rate":7
    },
    {
      "Id":2,
      "Rate":19
    }
  ],
  "CatalogArticles":[
    {
      "Id":"cefd328d98bc8540b39c01f304ee9142",
      "Number":"11.1.110.212.0.0",
      "UnitPrice":null,
      "PriceType":3,
      "SaleValidFrom":null,
      "SaleValidTo":null,
      "ValidityType":0,
      "ValidFrom":null,
      "ValidTo":null,
      "ValidDuration":null,
      "Sellingrule":0,
      "ShortText":"Gastro Galaxy Tabakwaren",
      "Description":null,
      "VATSplitting":"0=100",
      "PaymentTypes":null,
      "NeedsMedium":true,
      "ConjugatedArticleIds":null,
      "IsSubscription":false,
      "NeedsCustomer":false,
      "OnlyOncePerMedium":false
    }
  ],
  "ResponseStatus":null
}
```

2.2 Get media details

[GET](#) /mediumdata/{Number}

- ➔ Query medium number
- ➔ Returns medium details including credit / advance payments

Set the parameter "NumberType" = 1 to search for a hardwareserial (e.g. Mifare UID, barcode).

Set the parameter "NumberType" = 0 to search for an internal HKS media number.

2.2.1 Example 1 (day guest)

Request

Number: 1414343904 (Mifare UID of a test badge)

NumberType: 1 (Look for UID)

IncludePersonalization: false

IncludeMedia: false

SellingruleFilter <not set>

ValidPerformancesOnly: <not set>

Response:

```
{
  "Medium": {
    "Number": 7,
    "HardwareSerial": "1414343904",
    "Type": 11,
    "ValidTo": null,
    "Lock": 0
  },
  "Owner": null,
  "Performances": [
    {
      "Shorttext": "Vitaltheme&Sauna Tag",
      "Articlenumber": "6.1.30.3.1.0",
      "ExternalArticleId": null,
      "Articlegroup": "Sauna",
      "SellingRule": 1060,
      "ValidFrom": null,
      "ValidTo": null,
      "ActivationValidTo": null,
      "ActivatedAt": "2018-08-27T17:09:58.3130000",
      "UsedCount": 1,
      "BorderAmount": 1,
      "UnitPrice": 40,
      "VatCalculation": "1=100"
    },
    {
      "Shorttext": "Hauskredit",
      "Articlenumber": "6.1.0.1.0.0",
      "ExternalArticleId": null,
      "Articlegroup": "Systemartikel",
      "SellingRule": 1,
      "ValidFrom": null,
      "ValidTo": null,
      "ActivationValidTo": null,
      "ActivatedAt": null,
      "UsedCount": -20.9,
      "BorderAmount": -150,
      "UnitPrice": null,
      "VatCalculation": "9=100"
    }
  ],
  "ResponseStatus": null
}
```

The list of performances can be filtered by using the parameter „SellingruleFilter“ e.g.

- 1 show only credit
- 1080 show only advance payments
- 1021 show only prepaid package components

2.2.2 Example 2 (value card)

Request:

Number: 3
NumberType: 0
IncludePersonalization: true
IncludeMedia: false
SellingruleFilter: 1080
ValidPerformancesOnly: <not set>

Response:

```
{
  "Medium": {
    "Number": 3,
    "HardwareSerial": "1586122953",
    "Type": 15,
    "ValidTo": null,
    "Lock": 0
  },
  "Owner": {
    "FirstName": "Max",
    "LastName": "Mustermann",
    "DOB": "1975-05-10T00:00:00.0000000",
    "Gender": 0,
    "Salutation": 0,
    "Name": "Max Mustermann",
    "CustomerName": "Mustermann, Max",
    "Number": "6.1.1",
    "Id": "55f9a730079b4727a76c9b2c2cd950dd",
    "Picture": null,
    "Zip": "33100",
    "City": "Paderborn",
    "Street": "Lindenweg 6"
  },
  "Performances": [
    {
      "Shorttext": "ThermenCard Gold",
      "Articlenumber": "6.1.40.3.0.0",
      "ExternalArticleId": null,
      "Articlegroup": "Geldwertkarten",
      "SellingRule": 1080,
      "ValidFrom": null,
      "ValidTo": null,
      "ActivationValidTo": null,
      "ActivatedAt": null,
      "UsedCount": 982,
      "BorderAmount": 0,
      "UnitPrice": 1000,
      "VatCalculation": "9=100"
    },
    {
      "Shorttext": "ThermenCard Gold",
      "Articlenumber": "6.1.40.3.0.0",
      "ExternalArticleId": null,
      "Articlegroup": "Geldwertkarten",
      "SellingRule": 1080,
      "ValidFrom": null,
      "ValidTo": null,
      "ActivationValidTo": null,
      "ActivatedAt": null,
      "UsedCount": 1000,
      "BorderAmount": 0,
      "UnitPrice": 1000,
      "VatCalculation": "9=100"
    }
  ],
  "ResponseStatus": null
}
```

2.2.3 Calculation of available credit

- Find performance where SellingRule = 1
 - o initial credit = BorderAmount *-1
 - o used credit = UsedCount *-1

- available credit = (BorderAmount – UsedCount) * -1

2.2.4 Calculation of available advance payments

- Find performance(s) where SellingRule = 1080
 - available advance payments = UsedCount

There may be several advance payments in the performances list. Usually the values can be summarized. In some cases, an advance payment can only be used to pay special articles.

2.2.5 Example 3 (value card with discount)

The discount appears as a separate performance. In the example below: Articulenumber 6.1.9.200.0.0

Response:

```
{
  "Medium": {
    "Number": 2,
    "HardwareSerial": "1323066865965952",
    "Type": 11,
    "ValidTo": null,
    "Lock": 0
  },
  "Owner": null,
  "Performances": [
    {
      "Shorttext": "Mitarbeiterabbatt",
      "Articulenumber": "6.1.9.200.0.0",
      "ExternalArticleId": null,
      "Articlegroup": "Rabatte",
      "SellingRule": 1060,
      "ValidFrom": null,
      "ValidTo": null,
      "ActivationValidTo": null,
      "ActivatedAt": null,
      "UsedCount": 0,
      "BorderAmount": 1,
      "UnitPrice": null,
      "VatCalculation": "0=100"
    },
    {
      "Shorttext": "Mitarbeiter GWK",
      "Articulenumber": "6.1.40.100.0.0",
      "ExternalArticleId": null,
      "Articlegroup": "Geldwertkarten",
      "SellingRule": 1080,
      "ValidFrom": null,
      "ValidTo": null,
      "ActivationValidTo": null,
      "ActivatedAt": null,
      "UsedCount": 50,
      "BorderAmount": 0,
      "UnitPrice": null,
      "VatCalculation": "9=100"
    }
  ],
  "ResponseStatus": null
}
```

2.3 Create Customers

POST /customers/flat

→ Create a customer /person

If a ticket or receipt is to be personalized, create the customer at first.

The value of "ClientId" is configured in the HKS interface. Lastname and City are required fields.

Request Example:

```
{
  "Customer": {
    "Street": "Lindenweg 6",
    "Zip": "33102",
    "City": "Paderborn",
    "Firstname": "Max",
    "Lastname": "Muster",
    "Email": "max@Mustermann.de",
    "Birthdate": "2018-09-13T10:06:51.454Z",
  },
  "ClientId": "Webshop"
}
```

Response:

```
{
  "Customer": {
    "Operator": 6,
    "Facility": 1,
    "Id": "6.1.8",
    "Company": "Muster, Max",
    "Department": null,
    "Comment": null,
    "Street": "Lindenweg 6",
    "Zip": "33102",
    "City": "Paderborn",
    "Country": null,
    "Locked": false,
    "DebitorAccount": "32000016",
    "PersonId": "5f624eb8c1b74b1faa8b363a4eea635a",
    "Salutation": 0,
    "Gender": 0,
    "Title": null,
    "NearTitle": null,
    "Firstname": "Max",
    "Lastname": "Muster",
    "Email": "max@Mustermann.de",
    "Phone": null,
    "Mobile": null,
    "Fax": null,
    "Birthdate": "2018-09-13T12:06:51.4540000+02:00",
    "PersonComment": null,
    "DisabilityDegree": null,
    "DisabilityIdExpiry": null,
    "IsTheBear": true,
    "AllowedPaymentTypes": [
      "Cash",
      "Invoice",
      "BankCollect",
      "CreditCard"
    ],
    "SepsMandate": null,
    "UpdatedAt": "2018-09-13T13:47:40.3273572+02:00",
    "AdvanceOrderMediumNumber": null
  },
  "ResponseStatus": null
}
```

2.4 Create Tickets / Voucher

POST /mediumdata

→ Creates a Ticket / Voucher or add a personalization to an existing Ticket

By using the optional parameter "PersonId" the tickets will be personalized. Use the PersonId from the "Create Customer" response.

Request Example:

```
{
  "Number": "Mat1233333433",
  "PersonId": "7431667c0f364a20b09d95c3f2f41305"
}
```

Response

```
{
  "Medium": {
    "Number": 16,
    "HardwareSerial": "Mat1233333433",
    "Type": 11,
    "ValidTo": null,
    "Lock": 0
  },
  "Owner": {
    "FirstName": "Max",
    "LastName": "Mustermann",
    "DOB": null,
    "Gender": 0,
    "Salutation": 0,
    "Name": "Max Mustermann",
    "CustomerName": "Mustermann, Max",
    "Number": "6.1.9",
    "Id": "a8de9ca51fce455390aa343cc208f1e9",
    "Picture": null,
    "Zip": "33102",
    "City": "Paderborn",
    "Street": "Lindenweg 6"
  },
  "Performances": null,
  "ResponseStatus": null
}
```

2.5 Sale Request

POST /pos/sale

→ Sell a cart synchronously

Examples for using sale requests:

- sell Products and pay with HKS Medium (credit / debit)
- sell Tickets and pay with credit card

Paymenttype = -1 means type of Payment is determined by HKS (inhouse credit /advance payment / invoice) – only valid with specification of "MediumNumber"

Paymenttype = 36 means type of Payment is determined by the given brandname – only valid with specification of "brandname". The brandnames must be configured in the HKS System.

MediumNumber: Use Medium/Number from "Create Media" response

AllowPartialSale should be false.

Request Example 1: Selling a Product, paying with HKS wristband (at an inhouse F&B POS)

```
{
  "SaleDate":null,
  "ItemsToSale":[
    {
      "Id":"05280014-7A1F-C944-ABE7-002A361D3BC3",
      "ArticleId":"21.01.210.0205.0.0",
      "ShortText":"Bier 0,3",
      "Amount":1,
      "UnitPrice":3.3,
      "TotalPrice":3.3
    },
    {
      "Id":"CEFD328D-98BC-8540-B39C-01F304EE9142",
      "ArticleId":"21.01.210.0202.0.0",
      "ShortText":"Chicken Nuggets m. Pommes",
      "Amount":1,
      "UnitPrice":6,
      "TotalPrice":6
    }
  ],
  "AllowPartialSale":false,
  "PaymentInfo":{
    "PaymentType":-1,
    "MediumNumber":7
  }
}
```

Response

```
{
  "ReceiptNumber": "86010004-2018-9-1",
  "CustomerNumber": null,
  "SaleItemResponses": null,
  "ResponseStatus": null
}
```

Request Example 2: Selling a Ticket, paying with creditcard Visa

```
{
  "SaleDate":null,
  "ItemsToSale":[
    {
      "Id":"05280014-7A1F-C944-ABE7-002A361D3BC3",
      "ArticleId":"06.01.012.01.1.0",
      "ShortText":"Eintritt 2h",
      "Amount":1,
      "UnitPrice":25.3,
      "TotalPrice":25.3,
      "MediumNumber":"17"
    }
  ],
  "AllowPartialSale":false,
  "PaymentInfo":{
    "PaymentType":36,
    "BrandName":"Visa"
  }
}
```

Response

```
{
  "ReceiptNumber": "86010006-2018-4-2",
  "CustomerNumber": null,
  "SaleItemResponses": null,
  "ResponseStatus": null
}
```

2.6 Cancellation Requests

2.6.1 Cancel Receipt

POST /pos/cancellation/receipt

→ Cancel a receipt synchronously

Request Example:

```
{
  "ReceiptNumber": "86010004-2018-9-1",
  "CancelDate": "2018-08-29T11:49:14.558Z"
}
```

Response:

```
{
  "ReceiptNumber": "86010004-2018-9-2",
  "CancelMode": 0,
  "CanceledBookings": [
    {
      "ArticleNumber": "21.01.210.0205.00.00",
      "ShortText": "Bier 0,3",
      "Amount": 1,
      "OriginalAmount": 1,
      "Price": 3.3
    },
    {
      "ArticleNumber": "21.01.210.0202.00.00",
      "ShortText": "Chicken Nuggets m. Pommes",
      "Amount": 1,
      "OriginalAmount": 1,
      "Price": 6
    }
  ],
  "ResponseStatus": null
}
```

2.6.2 Cancel Receipt Position

POST /pos/cancellation/bookings

→ Cancel a receipt position synchronously

Request Example:

```
{
  "BookingsToCancel": [
    {
      "ReceiptNumber": "26010701-2019-1-1",
      "Position": 1,
      "Amount": 3
    },
    {
      "ReceiptNumber": "26010701-2019-1-1",
      "Position": 2,
      "Amount": 4
    }
  ]
}
```

Response:

```
{
  "ReceiptNumber": "86010004-2019-47-1",
  "CancelMode": 1,
  "CanceledBookings": [
    {
      "ArticleNumber": "06.01.070.0002.00.00",
      "ShortText": "Kuscheldecke",
      "Amount": 3,
      "OriginalAmount": 5,
      "Price": 78
    }
  ],
  "ResponseStatus": null
}
```

```
{
  "ArticleNumber": "06.01.070.0001.00.00",
  "ShortText": "Handtuch",
  "Amount": 4,
  "OriginalAmount": 7,
  "Price": 20.9
},
"ResponseStatus": null
}
```

⚠ Attention! A cancellation of 2 of 5 pieces cancels the entire booking and produces a new interface receipt: -5x and +3x.

Example:

Receipt 26010701-2019-1-1 contains in position 1 a sale of 5 towels.

```
{
  "ReceiptNumber": "26010701-2019-1-1",
  "Bookings": [
    {
      "Position": 1,
      "Amount": 5,
      "UnitPrice": 3,
      "TotalPrice": 15,
      "ArticleNumber": "21.1.210.205.0.0",
      "ShortText": "Towel",
      "IsCancelled": false,
      ...
    }
  ]
}
```

After a cancellation of 2 pieces, the entire booking will be marked as "cancelled".

```
{
  "ReceiptNumber": "26010701-2019-1-1",
  "Bookings": [
    {
      "Position": 1,
      "Amount": 5,
      "UnitPrice": 3,
      "TotalPrice": 15,
      "ArticleNumber": "21.1.210.205.0.0",
      "ShortText": "Towel",
      "IsCancelled": true,
      ...
    }
  ]
}
```

The Interface creates a new receipt: (return of 5 pieces, sale of 3 pieces)

```
{
  "ReceiptNumber": "86010001-2019-13-122",
  "Bookings": [
    {
      "Position": 2,
      "Amount": 3,
      "UnitPrice": 3,
      "TotalPrice": 9,
      "ArticleNumber": "21.1.210.205.0.0",
      "ShortText": "Towel",
      "IsCancelled": false,
      "IsCancellation": false,
      "CanBeCancelled": true,
      ...
    },
    {
      "Position": 1,
      "Amount": -5,
      "UnitPrice": 3,
      "TotalPrice": -15,
      "ArticleNumber": "21.1.210.205.0.0",
      "ShortText": "Towel",
      "IsCancelled": false,
      "IsCancellation": true,
      "CanBeCancelled": false,
      ...
    }
  ]
}
```

```
}
```

2.7 Receipt information

GET /pos/receipt/{ReceiptNumber}

→ Query receipt information

This function is usually only used for tests (examination of the results)

Request Example:

Number: 86010004-2018-9-1

Response:

```
{
  "ReceiptNumber": "86010004-2018-9-1",
  "CreationDate": "2018-08-29T14:08:48.0000000",
  "CustomerNumber": null,
  "Bookings": [
    {
      "Position": 1,
      "Amount": 1,
      "UnitPrice": 3.3,
      "TotalPrice": 3.3,
      "ArticleNumber": "21.1.210.205.0.0",
      "ShortText": "Bier 0,3",
      "IsCancelled": true,
      "IsCancellation": false,
      "CanBeCancelled": false,
      "AlreadyUsed": false,
      "BookingMediums": [
        {
          "Number": 7,
          "HardwareSerial": "1414343904",
          "Type": 1
        }
      ]
    },
    {
      "Position": 3,
      "Amount": 1,
      "UnitPrice": 6,
      "TotalPrice": 6,
      "ArticleNumber": "21.1.210.202.0.0",
      "ShortText": "Chicken Nuggets m. Pommes",
      "IsCancelled": true,
      "IsCancellation": false,
      "CanBeCancelled": false,
      "AlreadyUsed": false,
      "BookingMediums": [
        {
          "Number": 7,
          "HardwareSerial": "1414343904",
          "Type": 1
        }
      ]
    }
  ]
}
```