

Functional design: BookitLab sales order integration K6-K7

Table of Contents

Table of Contents	1
1. Introduction.....	1
1.1 Overview of data flow K6 and K7	2
1.2 Sequence diagram	3
2. Data flow K6 – Send sales orders (invoices) to Unit4 ERP	3
2.1 Use case	3
2.2 Characteristics of the data flow	4
2.3 Data export to Unit4 ERP	4
Legge inn link til swagger fra Ruben??	Error! Bookmark not defined.
2.4 Example of invoice generated in Unit4 ERP with data transferred from BookitLab	7
2.5 Transaction API	7
3. Data flow K7 – Send sales order (invoice) status to BookitLab	8
3.1 Use case	8
3.2 Characteristics of the data flow	8
3.3 Data export to BookitLab	8
3. Error handling.....	9
3.1 K6 error handling	9
An error handling scenario could be as follows:	9
3.1 K7 error handling	9

Version:

v.0.8 27.january 2022: Ernst Pedersen: Added algorithm suggestions and some history.

1. Introduction

Bookitlab supports two different methods of billing/invoicing: Internal and External.

Internal invoicing is a method where Bookitlab generates a posting file that can be imported into general ledger module in Unit4. This integration is established during 2021 and named K4 for sending data to Unit4, and K5 for receiving receipt back from Unit4.

External invoicing is a method where Bookitlab generates a PDF file with invoicing details. This is in a format as a regular invoice form. At universities in Norway this is used as an invoice basis, and the actual invoice is made manually by entering a sales order in Unit4.

During the GAP analysis and pilot phase back in 2020, NTNU and UiB agreed to settle for using the PDF output for external billing. The setup of this method for UiB was described in document “External company direct billing” in august 2020.

Later, when universities started implementing the new finance system, Unit4 ERP, the need for automated integration of sales orders from Bookitlab to Unit4 was raised. Then the definition of integration K6 – sales order integration to Unit4 and K7 – sales order response from Unit4 was defined.

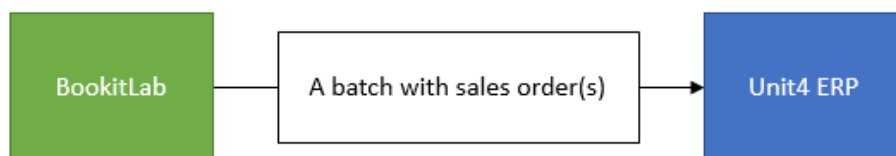
During 2021 the integration K6 and K7 has been described and much of the work on the university side is already implemented.

This document describes the integration at a functional level. In addition universities have made API-definitions, examples and test cases on a more technical level.

1.1 Overview of data flow K6 and K7

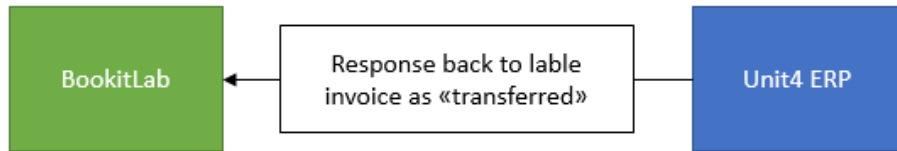
Data flow K6: Sales Order (invoice) sent from BookitLab to Unit4 ERP

- The data flow is triggered by a manual process in which a sales order (invoice) is being sent by lab resource owner
- A batch with several sales orders (invoices) is sent to Unit4 ERP for creation of invoices to external customers

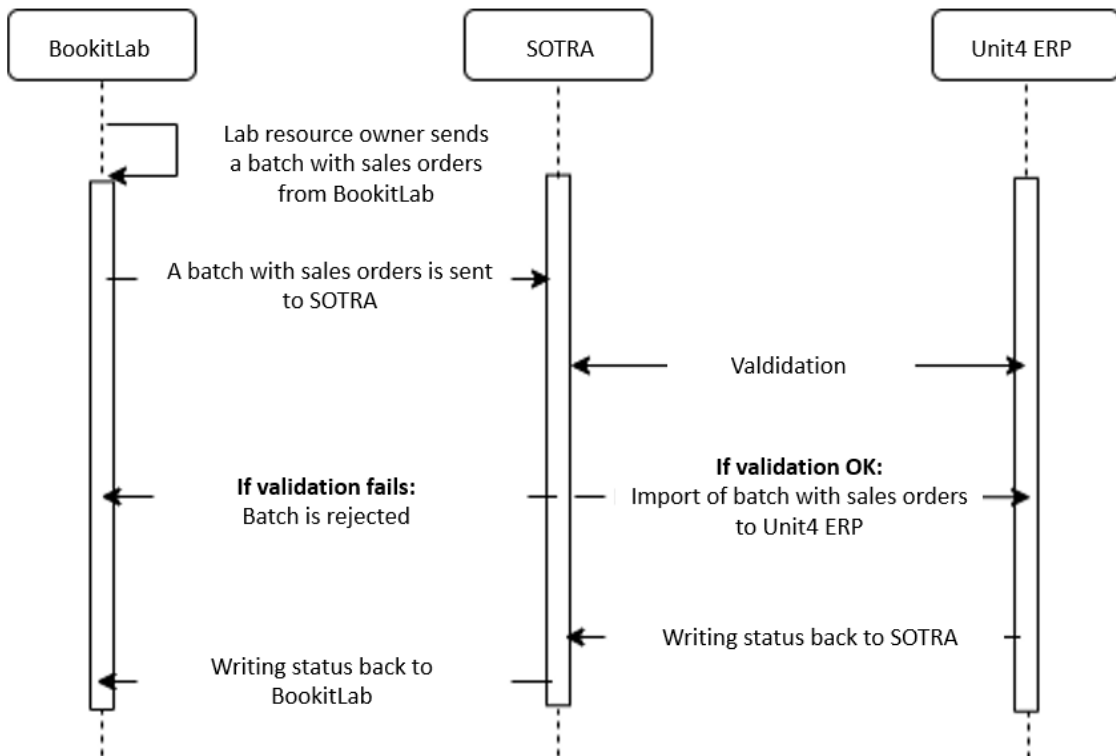


Data flow K7: A response from Unit4 to BookitLab (to be implemented on BookitLab side)

- When the batch with sales orders (invoices) have been processed in Unit4 ERP, a permanent sales order number is generated. A response is then sent back to BookitLab with the status “transferred” per batch



1.2 Sequence diagram



2. Data flow K6 – Send sales orders (invoices) to Unit4 ERP

2.1 Use case

Use case: As a lab resource owner, I want to charge external consumers when they have booked and used my resources. In practice I want to send a batch with sales orders to the financial system so that it can automatically create invoices to send out to external consumers.

2.2 Characteristics of the data flow

- In BookitLab, one sales order (invoice) is generated per project per rental place (core) every time the invoicing process is run. This is done monthly or weekly based on the tenants' routines.
- When sales orders (invoices) are generated in BookitLab, these end up in a batch, and the invoicing process can be done collectively for several rental locations at a time.
- The batches are assigned a batch number consisting of a sequence number and a timestamp for when the invoices have been generated.
- Only a whole batch can be sent at a time, i.e. no partial batching, and no more than one batch at a time.
- Data will go through a gateway at the universities (SOTRA), not directly to the financial system.

2.3 Data export to Unit4 ERP

This chapter describes the fields that are included in the transfer from BookitLab to Unit4 ERP.

The invoices sent from BookitLab will follow the same structure as they have previously had in the invoice exports but will be delivered as **json files** going forward.

Based on examples from exported invoices, the structure of a typical transaction is as follows below. Field starting with "@" means it should be a field from BookitLab. Example data/format in parenthesis:

```
{
  "batch_id": "@batchId",
  "invoices": [
    {
      "invoice_id": "@InvoiceId",
      "id_no": "@IDNo",
      "project_name": "Project name/@InvoiceProject",
      "invoice_remarks": "@InvoiceRemarks",
      "invoice_date": "@InvoiceDate",
      "invoice_core": "@InvoiceCore",
      "invoice_date_from": "@InvoiceFromDate",
      "invoice_date_to": "@InvoiceToDate",
      "processed_by_user_name": "@InvoiceProcessedUser",
      "processed_by_user_email": "@InvoiceProcessedUserEmail",
      "items": [
        {
          "description": "description value from @InvoiceSummury_BudgetUserService",
          "quantity": "total quantity value from @InvoiceSummury_BudgetUserService",
          "total_amount": "total amount value from @InvoiceSummury_BudgetUserService",
          "core_cost_center": "@CoreCostCenter",
          "project_number": "@project_number",
          "project_activity": "@ProjectActivity"
        }
      ]
    }
  ]
}
```

Below is a data table with fields in a sales order (invoice) mapped to the equivalent fields in an invoice in Unit4 ERP (see Unit4 ERP invoice example in 2.4).

Important note: The @<fieldname> in column 2 of the table refers to the field names in the “Invoice Export Template” screen of Bookitlab, where user can select which fields that should be printed on the PDF invoice. This is for illustration; we do not expect the PDF-export program or export template should be used for this integration at all. Instead K6 should start as a copy of the K4 program.

BookitLab field name	BookitLab API	Unit4 ERP field name	Unit4 ERP XML	Comments
BatchID	BatchID	Buntnr		BookitLab will send a BatchID to Unit4, for example sequence number “897”. This number should have “FB” in front. There should be nine characters in total, for example FB0000097.
InvoiceID	@InvoiceId		OrderNo	OrderNo is used in XML to separate several sales orders from each other in the same transfer.
Customer number?	@IDNo	Kundenr.	BuyerNo	This identifies the customers in BookitLab with the customers in Unit4
Project Name (Invoice/project reference)	@InvoiceProject	Deres referanse	Accountable	This name is shown as project name in BookitLab
Invoice remarks	New field in sprint summer 2021	Bestillingsnummer	ExtOrderRef	Staff and/or order number of the customer
InvoiceDate	@InvoiceDate	Bestillingsdato	OrderDate	This field is not shown on the invoice, but is an obligatory field in Unit4
InvoiceDate	@InvoiceDate	Leveringsdato	DelivDate	This field is shown on the invoice Unit4 generates
InvoiceID	@InvoiceId	Bilagstekst	ExtOrderId	The integration needs to combine: 'InvoiceID' + 'Invoice/projectreference' in the same field for the transfer to work
Invoice/project reference	@InvoiceProject			
InvoiceCore	@InvoiceCore	Topptekst	HeaderText	The integration needs to enrich the field with the text: Invoice from core: 'InvoiceCore' <jump one line down>
InvoiceFromDate	@InvoiceFromDate			
InvoiceToDate	@InvoiceToDate			

BookitLab field name	BookitLab API	Unit4 ERP field name	Unit4 ERP XML	Comments
				Periode: 'InvoiceFromDate' - InvoiceToDate' See invoice example in 2.4.
InvoiceID	@InvoiceId			Integration need to enrich the field with the text Bookitlab referanse: + 'InvoiceId'. See invoice example in 2.4.
Processed by	@InvoiceProcessedUser	Bunntekst	FooterText	Name of person who sends batch from BookitLab
Processed by email	@InvoiceProcessedUserEmail			Email address of the person who sends batch from BookitLab
Item description	Field from @InvoiceSummary_BudgetUserService	Beskrivelse	BuyerProductDescription	
Quantity	Field from @InvoiceSummary_BudgetUserService	Antall	Quantity	
Amount	Field from @InvoiceSummary_BudgetUserService	Totalt valutabeløp	LineTotal	Transfer of the total amount per item description
CoreCostCenter	@CoreCostCenter	Koststed	C1	
Delprosjektnr	project_number	Delprosjekt	BF	BookitLab allocates subproject number per invoice. 1:1 transfer is used for posting line
Analyse-nummer	@ProjectActivity	Aktivitet/Byggnr	B1	Prog4Biz needs to evaluate if this should be included, but the integration supports that this field will be populated

2.4 Example of invoice generated in Unit4 ERP with data transferred from BookitLab

2.5 Transaction API

BOTT
Bottveien 1
0001 OSLO

BOTT

SINTEF AS
c/o Fakturamottak
Postboks 4515
8608 MO I RANA

FAKTURA

Fakturanr: 60000121
Fakturadato: 10.09.2021
Forfallsdato: 24.09.2021
Lev.dato: 10.09.2021
KID: 1100050600001214
Kundenr: 10005
Ordrenr: 1000219
Side: 1 / 1

Deres referanse: Bestillingsnummer:
Referanse hos kunden Ansattnr / bestillingsnr hos kunden

Vår referanse:
Bookitlab

Header text

Invoice from core: MH - CoMed - Small animal facility
Periode: 01.06.21-15.06.21

Artikkel	Enh	Antall	Pris	MVA	Beløp i NOK
SPF - Large surgery room - Table 1	ANT	27,0	560,00	25%	15 120,00
SPF - Large surgery room - Table 2	ANT	1,0	1 500,00	25%	1 500,00

MVA sats: 25%	MVA gr.lag:	16 620,00	MVA beløp:	4 155,00	Sum:	20 775,00
Sum i NOK						20 775,00

Footer text

Bookitlab referanse: 878
Kontaktperson: Ola Pedersen / ola.pedersen@bott.no

Org.nr: 970422528MVA
E-post:
Telefon: 81549300

Banknavn: Dummybanken
Bankkonto: 9775 40 89222
IBAN: NO4797754089222
SWIFT: DNBANOKK

BookitLab is able to export transaction data in json format. This json payload is described by BOTT INT, and should be configured by Prog4Biz to fit the API definition provided by the developer team at BOTT INT.

Development team at BOTT INT have developed a transaction API for the microservice with PUT/POST capabilities.

Data will go through a gateway at the universities, not directly to UNIT4 ERP. Integration project at the universities must give examples and more detailed descriptions.

Implementation of the sales order export in Bookitlab should start with a copy of the current K4 program, and not use the PDF export program. The K4 program has most of the necessary algorithms needed. Instead of selecting "Internal" invoices, only "External" invoiced should be included. Some extra fields must be collected from the project and the cores. Instead of generating debet and credit postings, only debit postings with summary of invoice detail lines by instrument/service are needed.

3. Data flow K7 – Send sales order (invoice) status to BookitLab

3.1 Use case

Use case: As a BookitLab user, I want to know if the batch with sales orders (invoices) that was sent to the financial system was transferred successfully, and if not, what steps to take to make sure that they will.

3.2 Characteristics of the data flow

- A message from SOTRA notifies that the processing has ended and that the response is available in the standard interface.
- Status is written back to BookitLab as **"transferred"** when the data has been validated correctly and read into Unit4 ERP.
- Permanent Unit4 ERP reference is written back together with status. This is done per «invoice», status, date and document no. written back with invoice number as identifier.

3.3 Data export to BookitLab

This chapter describes the fields that are included in the transfer from Unit4 ERP to BookitLab (to be implemented on BookitLab side).

```
{
  "invoice_number": "1235",
  "status": "transferred",
  "salesorder_number": "6246342"
}
```

Below is a data table with fields transferred from Unit4 ERP to BookitLab:

Field name	BookitLab field name	Unit4 ERP field name	Comments
Invoice number	Invoice number	Order_id	Works as ID

Status		status	Status is written back as 'transferred' to BookitLab.
Unit4 ERP sales order number		order_id	This is a permanent number that is generated in Unit4 ERP. This number will be sent back to BookitLab and shown in the records.

3. Error handling

3.1 K6 error handling

If a batch contains an error, the batch is rejected in its entirety, and is returned to Bookitlab where a user must correct any errors contained in the batch before resending it.

An error handling scenario could be as follows:

If K6 is called again with the same data (same batch number), all data contained in that batch will be deleted from SOTRA, and the new data is saved. The deletion process must include all levels in the data structure, including related error messages in log tables.

Before potential deletion in SOTRA and data replacement, there must be a control mechanism to see if receiving new data is possible, meaning that status codes in the interface must allow replacement. If the batch has already been sent to UNIT4 ERP, the data should not be able to be replaced.

An example of an error like this could be if the transaction data has been rejected in the validation step in SOTRA, the error is fixed in Bookitlab and data is reposted to K6.

Automatic error messages must be given. The message should have specific info about where the process is failing, and identification of what data. There must be a restart mechanism to ensure that all invoices are processed even if operational problems occur. The alternative is that sufficient identification is given of which data from Unit4 ERP it applied to, so that it can be entered manually in BookitLab.

3.1 K7 error handling

For every sales order (invoice) sent from BookitLab, one of two things can happen:

- 1) Sales order (invoice) has been validated and transferred into Unit4 ERP is successful. This can be reported back to BookitLab automatically with a permanent invoice number from Unit4 ERP and a status that indicates that the sales order (invoice) has been transferred successfully.
- 2) Something goes wrong along the way, either because validation on the BOTT side failed or that Unit4 ERP rejected the sales order (invoice) during the import. An error message can then be sent via TOPdesk to a general queue or a specific user if this is defined in the transaction API out of BookitLab.

BookitLab guarantees delivery for BOTT's integration. If the bundle is stored in the integration, code «201 created» is sent back to BookitLab, if something goes wrong, an error code is sent back.

If a bundle that already exists in the integration is resent, an http response is sent back stating that the bundle already exists.