SAP Event Ticketing
# Table of Contents

1 Security in SAP Event Ticketing ................................................................. 3
1.1 Adjustments in Session Management .................................................. 3
1.2 Adding Additional DPP-Compliant Information ................................... 3

2 Adjustments for Cash Management .......................................................... 5
2.1 Number Range for Cash Balance Change and Cash Reference Difference Documents .................................................. 5
2.2 IDEA Export ......................................................................................... 5
2.3 Cash Journal ....................................................................................... 6
2.4 Making Balance Changes ................................................................. 7
2.5 Entering Cash Register Differences .................................................. 9
2.6 New Cover Page for Cash Balancing ................................................. 10
2.7 Change to the “Reset Coupon” Feature ............................................. 10

3 Adjustments in Back Office ...................................................................... 11
3.1 Configuring Address Data ................................................................... 11
3.2 Multiple Selection for Conversion .................................................. 11
3.3 Entry of Bar Code of the Bundle Master Ticket .............................. 12
3.4 Displaying Events in the Tour Calendar ........................................ 12
3.5 Paying with ISR ............................................................................. 12
3.6 Selling Chunked Slots After Start of the Chunk ............................. 14
3.7 Simple Forwarding of Tickets via XMLRPC .................................... 15

4 Adjustments in WOW Shop Template Set ................................................. 16
4.1 Revoking a Marketing Declaration of Consent ..................................... 16
4.2 Social Invitation ............................................................................. 17
4.3 Display of Additional Text .............................................................. 19
4.4 Member Search with External Customer Number .......................... 19
4.5 Responsive Design for Chunked Events ........................................ 20
4.6 Preventing a Date of Birth in the Future .......................................... 20
4.7 Unique ID of LOGIN Points ............................................................ 20
4.8 Adjustments to Password Change Procedure .................................. 20

5 Adjustments in XMLRPC Web Interface ................................................ 22
5.1 New Features in XMLRPC ................................................................ 22
5.1.1 tickets_session_token() .......................................................... 22
5.1.2 tickets_set_ticket_personalization() ........................................ 23
5.1.3 tickets_get_ticket_personalization() ........................................ 25
5.1.4 tickets_get_ticketlist_customertid() .......................................... 27
5.1.5 tickets_forward_ticket() .......................................................... 29
5.1.6 tickets_delete_ticket_personalization() .................................... 30
5.2 Extended Functions for XMLRPC ...................................................... 31
5.2.1 tickets_get_ticket() ................................................................. 33
1 Security in SAP Event Ticketing

1.1 Adjustments in Session Management

Feature Description

Session ID for SAP Event Ticketing in the PHP session
In the new version, session management is switched over to cookies. As a result of this change, the session ID for SAP Event Ticketing is no longer saved as a "wes" parameter in the URL (that is, the session is no longer visible "externally"). Up to SAP Event Ticketing Version 1902, the session ID was primarily transferred in the "wes" parameter. In the new version, this parameter is used to identify the tab/window in the local browser. The combination of the "wes" parameter with the session cookie enables the back end to continue to sign in to multiple sessions simultaneously in one browser. In the online shop, the user is assigned only one session per shop, even if the shop is opened on multiple tabs within one browser. The session ID for SAP Event Ticketing is no longer saved as "wes" in the URL. It is no longer visible "externally". The user of the back end or online shop no longer has any way to read the session ID for SAP Event Ticketing. The session ID for SAP Event Ticketing is now located in the array of the PHP session.

Prerequisites
No prerequisites must be observed. The change applies to all SAP Event Ticketing systems.

Notes for Implementation
The browser programs contain only one common cookie store for all browser tabs/windows. This means that one cookie is valid across all browser tabs/windows. Only this one cookie is read and used by all browser tabs/windows.
It is no longer possible to indicate SAP Event Ticketing sessions externally or process them using JavaScript.
The session ID for SAP Event Ticketing can be transferred to the SAP Event Ticketing system as a "wes" request variable on a one-time basis.

Security-Related Notes
The "Additional Session Security" feature has been removed in the new version.

1.2 Adding Additional DPP-Compliant Information

Feature
We have adjusted the way in which information relevant to data protection and privacy (DPP) regulations is saved. Whenever a user consents to the data protection regulations, the system creates a progress entry. The user consent is also displayed under "Data Protection and Privacy Options".
2 Adjustments for Cash Management

2.1 Number Range for Cash Balance Change and Cash Reference Difference Documents

Feature
You define number ranges on the “Master Data” tab under Company > Master Data > Number Ranges. You can use a number range at various locations in the system depending on its type.

For invoices, the assignment of numbers from the number range can be restricted to specific invoice types. The following types apply:
- Invoice
- Cancelation Invoice
- Option
- Advance Booking Statement
- Organizer Billing
- User Closing

- Note
  If no invoice type is selected, the defined number range applies for all invoice types.

For cash document numbers, the assignment of numbers from the number range can be restricted to specific cash document types. The following types apply:
- Change to Cash Balance
- Cash Register Difference

- Note
  If a cash document type is not created, the defined number range applies to both document types.

- Caution
  Note that multiple checkouts "share" one number range. They are therefore not necessarily sequential within one checkout.

2.2 IDEA Export

Feature
You can use this export to provide data in a form that can be imported into IDEA (the auditing software of the German Financial Administration) in the case of an audit.

Prerequisites
The **Accounting** authorization is activated in your master data.

You can create this export only as a registered system operator.

**Features**

The checkout operator can use this file to export all relevant transactions in accordance with the legally stipulated standard. Only authorized users are permitted to create this export in SAP Event Ticketing. The export is in the standard format that complies with the legal provisions.

The export can be created for a specific time period and enables the cash auditor to receive standardized data from a point-of-sale system and import this data into the required auditing software.

- To create an export, choose **Accounting > Accounting Export > IDEA Export**
- Then, choose "New".
- Select the seller by right-clicking the directory tree.
- Optionally, select a user. If you do not select a user, the system performs the export for all users at a store.
- Then, specify the export period.
- Save your settings.
- The system creates the export. Since the exports run as background processes, you can start several exports simultaneously.
- The export that has been created is displayed on the right-hand side.
- Click on the small envelope icon to download the export.

**Note**
The system also creates an export if no data exists.

### 2.3 Cash Journal

**Feature**

The cash journal report provides an overview of all transactions, cash balance changes, and cash register differences for a user within a specific time period.

**Feature Description**

You can use the **Sales > Checkout > Cash Journal** dialog to create a cash journal report. The user that is currently signed in is used by default. If a user has extended administrative rights, all users at the company are available in the selection list. The associated user rights are:

- Display All User Closings
- Accounting

You can set these up under **Master Data > Company > Users** under "Statistics/Cash Balancing" or "Ticket Management/Accounting".

You can select a period including times of day. You can restrict the search to the "cash" payment method. The output is provided in a table in HTML format. The data can also be exported to a file in Excel format.
Prerequisites
You have company authorization under Master Data > Company > Extended Cashier Statement.

Activities
- To create a report in the cash journal, go to Sales > Checkout > Cash Journal.
- Select the user.
- Enter a time period.
- Select the export format.
- Choose "Display Data".

Note
Opening balance
The opening balance displayed in the report is calculated as follows:
Opening balance = closing balance of the most recent valid cash balancing.
+ all transactions/sales by the user in the period between the date of the last cash balancing and the start date of the cash journal report
+ all balance changes and differences in this period
If no valid cash balancing exists, the opening balance is 0.

For the "cash" payment method, the report shows the subtotals and the closing balance. All other payment methods are summed up together.

For invoices with payment splitting, the report shows the gross totals per payment method. If multiple tax rates apply for each invoice, the tax values are displayed accordingly for each tax rate.

If you select "Cash Payment Method Only", the report shows only amounts and the "Various Payment Methods" column is hidden.

For an invoice with payment splitting with a cash component, only this element is displayed.

2.4 Making Balance Changes

Feature
Under Sales > Checkout > Balance Changes, you can make balance changes by entering cash movements as a withdrawal or incoming payment.

Prerequisites
In the user data under Master Data > Company > User, you have activated the Management of Cash Balances feature under Statistics/Cash Balancing.

The user that is currently signed in is used by default. If a user has one of the following administrative rights, all users at the company are available in the selection list.
• Display All User Closings
• Accounting

You can set these up under **Master Data > Company > Users** under **Statistics/Cash Balancing** or **Ticket Management/Accounting**.

A standard layout must also be set up under **Master Data > Company > Company: Documents**.

**Assignment:**
• Layout Type: Cash document
• Layout Selection: Default document for cash documents

**Note for chameleon users**

The following applies for chameleon users:
• Chameleon users with "Accountant" and "Display All User Closings" rights see all users at a store in the selection list.
• Chameleon users without these rights (only the "Management of Cash Balances" right) do not see any users for selection and cannot create any entries for a new cash balance change.

**Document numbers**

Separate document numbers can be assigned to documents for balance changes. You use number ranges to set up special document numbers.

**Display/printing of balance changes**

You can search for balance changes for a user within a specific time period. The search can be restricted to one document category.

The following document categories exist:
• Incoming payment
• Withdrawal
• Difference

The individual documents are displayed in PDF format and can also be printed out (browser printing). A standard layout exists for the documents.

**Procedure**

To enter a balance difference, proceed as follows:

1. Choose **Balance Change** in the **Checkout** area.
2. The system opens a screen in which you can enter the cash movement.
3. Select the required document category. If you select the "Incoming Payment" document category, the system enters a positive amount; for the "Withdrawal" document category, it enters a negative amount.
4. Specify the amount, currency, and **Note to Payee** and choose **Save**.
5. The incoming payment or withdrawal is displayed on the right-hand side with the corresponding document number. If you choose "Display/Print", the system displays a "Display/Print" button again. You can use this to print the cash document.
2.5 Entering Cash Register Differences

Feature
During the creation of a user closing in the Checkout area, you can specify cash register differences by entering the relevant differences.

Prerequisites
In the company master data under Master Data > Company, you have activated the Extended Cashier Statement feature under Settings > Accounting.

In the user data under Master Data > Company > User, you have activated the Management of Cash Balances feature under Statistics/Cash Balancing.

Procedure
To enter a difference, proceed as follows:

- Enter a time period for Closing Until. To enter cash register differences, you must create a final user closing.
- Select the relevant user.
- Choose Create Closing.
- The system displays a new menu for entering the cash register differences.
- Enter the cash register differences. Note that the actual cash balance is displayed on the right-hand side. This is the value that is currently in the checkout. When you make an entry, ensure that you enter the value that should subsequently appear in the checkout minus the difference (target amount).
- Choose "Enter Differences". You can specify the differences for all payment methods of the cash balancing and for two additional payment methods. The totals calculated from the cash balancing are displayed on the right and are prepopulated in the input fields. For the cash balance, you must make a cash payment to determine the current value.
- The user closing is then created.
- Now go to Specify Document Content and select Cash Balance Changes and Differences.
- Choose Display Closing. The system displays the user closing in a separate tab.
2.6 New Cover Page for Cash Balancing

Feature
The new cover page also shows all cash balance changes and differences.

Feature Description
After you activate the Extended Cashier Statement, differences can be specified under Sales > Checkout > User Closing. This applies only for a final cash balancing. You can specify the differences for all payment methods of the cash balancing and for two additional payment methods. The totals calculated from the cash balancing are displayed on the right and are prepopulated in the input fields. For the cash balance, you must make a cash payment to determine the current value.

You can specify the content of the document for displaying the cash balancing. You can also select Cash Balance Changes and Differences for this.

The following data is shown on the cover page of the cash balancing document
- Cash Balance
- Balance for Other Payment Methods
- Revenue per Tax Rate

The balance data includes the transactions, balance changes, and differences with reference to the payment methods.

Prerequisites
You have activated the Extended Cashier Statement company authorization under Master Data > Company > Settings.

2.7 Change to the "Reset Coupon" Feature

Feature
In the new version, you are no longer able to reset coupons under Master Data > Ticket Management > Coupon Management. As a result, the "Change" button is no longer available in this dialog.
3 Adjustments in Back Office

3.1 Configuring Address Data

Feature
You can show or hide address fields in Sales and under Sales > Customer. The user can also sort the sequence of the displayed address fields.

Configuration profiles enable you to use configurations throughout the company.
You can create these profiles under System > System Configuration > Configuration Profiles.

Prerequisites
To use configuration profiles, you must have activated the Assign Configuration Profiles authorization in the user settings under Master Data > Company > User.
The Edit Dialogs and Edit Dialogs for Company user settings are required to configure the dialogs.

Activities
To use configuration profiles, proceed as follows:

Go to the System > System Configuration > Configuration Profiles tab and choose New.
Enter a Name and select the language.
Then, assign companies to the profile.
Save your entries.

In Sales, you can then adjust the input fields in the Customer and Additional Addresses dialogs and on the customer data screen, and show or hide the relevant fields.
Address fields that are identified as required entry fields cannot be hidden.

3.2 Multiple Selection for Conversion

Feature
A new filter feature is provided in the Convert Booking Confirmation dialog for selecting the relevant tickets. You can use this feature to select individual tickets or specific areas.

Functionality Description
The new filter feature permits the following selections:

- 1,3,5 - for the tickets 999999999-1, 999999999-3, 999999999-5 from the invoice 999999999
- 1,2,5 - for the tickets 999999999-1, 999999999-2, 999999999-3, 999999999-4, 999999999-5 from the invoice 999999999.

Once you have defined filter criteria in an input field, you can apply them by choosing the Filter Tickets button. The selection skips non-existent tickets.
3.3 Entry of Bar Code of the Bundle Master Ticket

Feature
Admission control - transfer of bar codes for bundle events

Bundle events enable you to sell multiple products in one sales transaction. In this version, the system transfers the data for all individual events to admission control when you print the master ticket. When you print the single tickets, the system transfers the bar codes for the single tickets.

3.4 Displaying Events in the Tour Calendar

Feature
You can manage the tour calendar under Master Data > Event Series/Metadata > Tour Calendar and assign it to events of the type Chunks.

For chunked events for which you had enabled the setting under Master Data > Event > Do not display in the event calendar, the system also no longer displayed the tour calendar.

However, since many customers would like to display only the tour calendar and not the individual chunked events, the system now always displays a tour calendar if bookings can be made for the assigned chunked events.

You can therefore now decide whether the individual events are displayed in the event calendar separately and independently of the tour calendar.

3.5 Paying with ISR

Feature Description
The Event Ticketing system has been extended to support payment with an ISR (inpayment slip with reference number) in the back office.

To use this feature, you must set up a corresponding payment method and use the relevant layout variables in an invoice layout.

Functionality Description
The task of the new payment method is to permit the required configuration for the ISR and perform the relevant calculations for the layout variables.

Once the new means of payment has been set up, it can be used in the booking process as usual. In the booking process, the configured settings (participant number, bank account number) and the card data are noted in the payment transaction. After the invoice has been created in the
system, the ISR is generated based on the invoice data and transaction data. Three new layout variables have been created for calculating the required information on the ISR:

- **ESR_REFERENCENUMBER** - 27-digit reference number in the format: [bank account number(6)][customer-specific identification number(20)][check digit(1)]
- **ESR_CODE** - 01[amount(10)][check digit(1)]-[ESR_REFERENCENUMBER]+[participant number]
- **ESR_BANK_ID** - for example, 01-123456-8

In addition to these new layout variables, a new function SPLITPRICE has been added to the layout engine in order to split the price into its three component parts (euros, cents, currency).

This layout variable must be created in an existing or new layout that uses the "Invoice" document category and for which the "Single Invoice Layout" option is set.

**Prerequisites**

Additional means of payment must be activated in the payment configuration under **Master Data > Company > Payment Configuration**.

A document flow must already be set up for the company under "Master Data > Company > Doc. Flows/Messages > Document Flow".

To participate in the ISR procedure, a participant number and bank account number must be requested from the relevant bank.

To print the ISR, an ET system should be set up locally.

The shopping carts must contain amounts in one currency only.

**Activities**

To use the ISR in an SAP Event Ticketing system, you must set up a new payment method and adjust the documents in the document flow.

To set up this payment method, go to "Master Data > Company > Payment Configuration > Electronic Payment Method".

From the list of available payment methods, select **Inpayment Slip with Reference Number** and create a new payment method with the following settings.

- Currency
- Participant Number (must be specified in the format [dd]-[ddddddd]-[d] as this is required on the ISR)
- Account Number
- Reference Number - Part 1
- Reference Number - Part 2
- Reference Number - Part 3
Once the new payment method has been created, you can use it in the system under **Master Data > Company > Payment Configuration > Additional Means of Payment.** To do this, create a new means of payment with the following settings:

- **Name:** [freely selectable]
- **Payment Method:** Direct Debit
- **Gateway:** Inpayment Slip with Reference Number
- **Booking Result:** Booking
- **Can Only Be Selected with Customer Data:** x

After you complete these steps, the ISR is available in the booking process.

### Setting Up the ISR Layout in the Document Test Stream

To set up the layout, choose **Master Data > Layout System.** Create a new document with the following settings:

- **Name:** [freely selectable]
- **Description:** Layout for Inpayment Slip with Reference Number
- **Document Type:** Invoice
- **Layout Type:** "Freely Positionable Elements"

After you save this setting, choose **Edit > Change** to adjust the document layout. You can place the specific layout variables for the ISR in this layout:

- **ESR_BANK_ID**
- **ESR_REFERENCENUMBER**
- **ESR_CODE**

### Notes for Implementation

Prior to productive use of the ISR, a test sample must be sent to the relevant bank and a check must be performed of the ISR that was created.

### 3.6 Selling Chunked Slots After Start of the Chunk

#### Feature

In the online shop and in advance booking in the back office, you can now specify that bookings can be made during a time interval that has already started.

#### Functionality Description

To set that bookings can be made for a time interval that has already started, navigate to the event administration and choose **Master Data > Event > General Settings.**
If the **Time Slots Can Also Be Booked After Start** feature is active, all time slots that have already started can be booked until they end.

**Notes for Implementation**

This setting does not have an effect on evening box office operation.

### 3.7 Simple Forwarding of Tickets via XMLRPC

**Feature**

Simple forwarding of tickets using XMLRPC

Tickets can be forwarded using the internal XMLRPC service of SAP Event Ticketing. In order to transfer the tickets, you must activate the **Forwarding of Tickets via XMLRPC** option in the ticket transfer configuration at administration level. If this option is active, a ticket transfer can be performed by the customer in the online shop as described above or using the XMLRPC service. This can be integrated within an app.
4 Adjustments in WOW Shop Template Set

4.1 Revoking a Marketing Declaration of Consent

Functionality Description
Consent to the privacy statement can be revoked in the WOW shop template in SAP Event Ticketing. This action can also be performed by the data protection officer under Sales > Customer in SAP Event Ticketing.

Prerequisites
- WOW shop template set
- The parameter global_data_privacy is active.

Activities
Display under "My Account"
After signing in to the shop, the customer can choose the Marketing/Data Protection entry under My Tickets. The menu item is also displayed in the dropdown field under My Account. If fully configured, all 4 types of marketing declaration of consent displayed. The declarations of consent are displayed with a link for submitting or withdrawing consent, depending on the customer’s status.

The Marketing/Data Protection dialog displays to the customer the consent given for marketing purposes and the consent to the data protection policy, as well as the status and date of the consent. To revoke consent to the data protection policy, the customer must choose the Revocation link.

If the customer still has an active declaration of consent to marketing or an active membership, the system indicates that this consent must first be revoked or that the membership must be canceled with the organizer directly. The button remains as "OK" until all prerequisites for the revocation of the data protection policy have been met.

⚠️ Note: Message about data protection in the online shop
Note that a revocation of the organizer’s data protection policy cannot be performed due to the following factors.
In order to revoke the data protection policy, you must revoke your valid declarations of consent to marketing and cancel your active memberships. Memberships can only be canceled with the organizer directly. Once the above requirements have been met, your consent to the organizer’s data protection policy can be revoked online.

Once the prerequisites have been met, the customer can revoke consent to the data protection policy. The button changes to "Revocation". Once the revocation has been performed successfully, the system displays a corresponding message in the "Marketing/Data Protection" dialog and the customer is flagged for deletion in SAP Event Ticketing.

The system displays the following note in the dialog box:
Notes for Implementation

- The "Marketing/Data Protection" item is displayed if the parameter global_data_privacy is active.
- If the parameter is disabled and the shop value SHOW_MYACCOUNT_MARKETING is active, the system displays an item where the declaration of marketing consent can be revoked, but without the "Data Protection" area.
- The declaration of marketing consent can be revoked online under "Marketing/Data Protection".
- Memberships can be canceled only by contacting the organizer directly.
- The date of consent to the privacy statement is stored in the history of the customer data record.
- All messages can be managed using shop and message texts.

4.2 Social Invitation

Feature

The social invitation feature enables customers of the SAP Event Ticketing WOW shop template set to invite friends to an event for which they have themselves purchased tickets. The invitation prereerves seats that can be shared using a unique link. As a result, the person sending the invitation does not have to pay out the total amount for the tickets. The invitation link can be shared via Facebook, sent by email, or copied to the clipboard for use in other messenger services.

As a prerequisite for this feature, the person sending the invitation must have purchased at least one ticket themselves.

As a result, the inviter is not able to share all tickets using an invitation link. The invitation link also applies to the entire reservation and contains all seats/tickets. The link can be called up directly in the WOW shop template set. The invitation status can also be checked under "My Account" and the invitation shared again. The inviter also has the option to buy tickets from the invitation themselves.

From the organizer’s perspective, the social invitation feature increases the number of customer data records because fewer visitors buy tickets for friends who are not known to the organizer.

Feature Description

The following back-end configuration must be performed in order to use the social invitation feature in SAP Event Ticketing.

- Activate shop value for social invitation
- Configure social invitation in the event
- Make company settings for using social invitation ("Max. Option Days (Max. Reservation Days)") that are greater than zero
- Set up a message plug for the reminder email
Social invitation can be configured per shop. After signing in to the company in which the shop was set up, you can activate the shop value `SHOW_SOCIAL_INVITATION` under `Master Data > Company > Shops > Shop Values`. If you want to remove the social invitation feature from the WOW shop, you can deactivate the shop value or use the `Delete` button for deactivation.

You activate the function using "Share Tickets’ Release" under `Master Data > Event`. You can make the following configurations:

- Name (name of the release)
- Activate “Share Tickets” (option to activate and deactivate the release)
- Start Time for Release (start of the release - a release for sharing tickets cannot be issued prior to the start of advance booking)
- End Time for Release (end of the release - a release for sharing tickets cannot be issued after the end of the event)
- Maximum Tickets/Invitation (maximum number of tickets that can be shared per order)
- Validity (select minutes or days)
- Validity in Minutes/Days (validity of invitation)
- Reminder Minutes/Days (time as of which the reminder email should be sent)

To deactivate the feature in the event, you must remove the checkmark under "Activate ‘Share Tickets’".

**Note**

To use the social invitation feature, it must be active in both the shop and the event.

Social invitation is available for regular individual events. The feature cannot be used for subscriptions, resource-free events, coupon events, guided events, or ticket packages.

The validity of the invitations is checked every 15 minutes by an automated cycler job that identifies and removes expired reservations.

In addition to the shop and event administration, the company at which the online shop is established must be capable of generating options/reservations in order to create invitations. To enable this, the value for `Max. Option Days (Max. Reservation Days)` must be specified under `Master Data > Company > Sales Management`.

To remind the inviter about seats from the invitation that have not yet been taken up, SAP ET offers the option to send a reminder email to the inviter. The time at which this message is created can be specified in the event administration under "Share Tickets’ Release" (see above). The message itself is created at the company in which the online shop is set up. Under `Master Data > Company > Doc. Flows/Messages > Message Plugs`, select the message slot "Invitation Reminder Email".
4.3 Display of Additional Text

Feature
The new version includes a new shop value: SHOW_TAXES_ADVICE.

If this value is active, the system displays the text "Price Incl. VAT and Charges, Excl. Delivery Costs" for SVG events and in the shopping cart (for all events).

An "*" is also displayed alongside the word "Price" to alert customers in the shop to this information.

Notes for Implementation
This extension was made for the desktop version in the WOW_Shop template set.

4.4 Member Search with External Customer Number

Feature
We have adjusted the template for personalization in the WOW shop template set.

You can now find a customer using the following search parameters.

You can perform a customer search using the:

- Membership number
- Customer number (new)
- External customer number (ext_Kundennr - new)

Notes for Implementation
If you use the extended search with a customer number, we recommend that you also activate the query for the date of birth.
4.5  **Responsive Design for Chunked Events**

**Feature**
For chunked events, bundle events with chunked events, and tour calendars, you can now use a responsive design for bookings in the WOW shop template set on a mobile device.

**Feature Description**
In many cases, we have switched over the booking behavior for chunked events to responsive layout elements that automatically adapt to the screen size.

The selection of events (for example, chunked, resource-free, booked seats) switches automatically from horizontal to vertical.

---

4.6  **Preventing a Date of Birth in the Future**

**Feature**
In the WOW shop template set, it is no longer possible to enter a date of birth in the future.

The system displays a button for entering a date of birth. If you enter a date in the future, an error message appears. If the date is in the past, the system saves the date.

---

4.7  **Unique ID of LOGIN Points**

**Feature**
As a customer of the WOW shop template sets, I receive unique DOM element IDs for all buttons for signing in or registering.

We require these unique DOM element IDs so that we can, for example, activate the "Sign In/Register" button, irrespective of the current view or state.

---

4.8  **Adjustments to Password Change Procedure**

**Feature**
In the WOW shop template set and the WOW shop mobile facelift, you can change your password under **My Account > Password**. When you change your password, you must first enter your current password and then the new password. You must then confirm the required password.

**Activities**
Open the shop. In the menu bar, choose **Sign in/Register**.

Sign in to the WOW shop template set using your current access data. Then, choose **My Account** followed by **Password**.
To change your password, enter your **current password** once more, then enter the required new password and **confirm** this password.

Choose **Save**.

The system confirms that your password has been updated.

---

**Note**
Information

Your password has been updated

---

**Caution**

This new entry does not apply for the **Forgotten Password** feature. In this case, you can still enter a new password after receiving an email.
5 Adjustments in XMLRPC Web Interface

5.1 New Features in XMLRPC

5.1.1 tickets_session_token()

tickets_session_token()
tickets_session_token(array $params) : array

*Function for generating a session*

Parameters

<table>
<thead>
<tr>
<th>array</th>
<th>$params</th>
<th>Array with function-specific parameters</th>
<th>The following function parameters are defined:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td></td>
<td></td>
<td>token</td>
<td>API key</td>
</tr>
<tr>
<td></td>
<td></td>
<td>languageid</td>
<td>Language ID</td>
</tr>
</tbody>
</table>

Returns

array —

Contains the return values

If successful (errormessage = OK), the following return values are defined:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Data Type</th>
<th>Field Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>sessionid</td>
<td>Session ID</td>
<td>String</td>
<td>20</td>
</tr>
</tbody>
</table>

In case of an error, the following return values are defined:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Data Type</th>
<th>Field Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>errorcode</td>
<td>Contains an error message</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>errormessage</td>
<td>Describes the error</td>
<td>String</td>
<td>Variable</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Data Type</th>
<th>Field Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>errorfields</td>
<td>Elements that caused the error</td>
<td>Array</td>
<td></td>
</tr>
<tr>
<td>fieldname</td>
<td>Name of element that caused the error</td>
<td>String</td>
<td>Variable</td>
</tr>
</tbody>
</table>

#### Generate API Key

**Prerequisites**

To generate an **API key**, you must activate the **XMLRPC API User** under **Master Data > Company > User > General Settings**. Save your settings.

**Activities**

If the XMLRPC API user is active, a new **Generate New API Key** item appears at the bottom of the menu. The newly generated API key is displayed in red above the user ID.

### 5.1.2 tickets_set_ticket_personalization()

```php
tickets_set_ticket_personalization(array $params) : array
```

- Function for personalizing a ticket
- This function personalizes a ticket using a customer ID or using the personal data that has been provided.
- In this case, the system checks that the session customer is the ticket holder and that the ticket is not blocked or canceled.
- The ticket must also not be within the ticket transfer or resale process.
- The event is used to define which details are mandatory for personalization and these details are also checked by the function.
- Purchasing authorizations are not currently taken into account or evaluated within this function.

### Parameters

<table>
<thead>
<tr>
<th>array</th>
<th>$params</th>
<th>Array with function-specific parameters</th>
<th>The following function parameters are defined:</th>
</tr>
</thead>
<tbody>
<tr>
<td>parameter</td>
<td>Description</td>
<td>Data Type</td>
<td>Field Length</td>
</tr>
<tr>
<td>sessionid</td>
<td>Session ID</td>
<td>String</td>
<td>20</td>
</tr>
<tr>
<td>ticketid</td>
<td>Ticket ID</td>
<td>String</td>
<td>20</td>
</tr>
<tr>
<td>array</td>
<td>$params</td>
<td>Array with function-specific parameters</td>
<td>The following function parameters are defined:</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
<td>-----------------------------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>customertid</td>
<td>Customer TID</td>
<td>String</td>
</tr>
<tr>
<td></td>
<td>name</td>
<td>Last name (mandatory if no customer ID)</td>
<td>String</td>
</tr>
<tr>
<td></td>
<td>firstname</td>
<td>First name (mandatory if no customer ID)</td>
<td>String</td>
</tr>
<tr>
<td></td>
<td>greeting</td>
<td>Salutation (2 Mr., 3 Ms., 4 company)</td>
<td>Integer</td>
</tr>
<tr>
<td></td>
<td>street</td>
<td>Street</td>
<td>String</td>
</tr>
<tr>
<td></td>
<td>housenr</td>
<td>House number</td>
<td>String</td>
</tr>
<tr>
<td></td>
<td>city</td>
<td>City/location</td>
<td>String</td>
</tr>
<tr>
<td></td>
<td>zip</td>
<td>Zip code</td>
<td>String</td>
</tr>
<tr>
<td></td>
<td>country</td>
<td>Country</td>
<td>String</td>
</tr>
<tr>
<td></td>
<td>nationality</td>
<td>Nationality</td>
<td>String</td>
</tr>
<tr>
<td></td>
<td>birthday</td>
<td>Date of Birth</td>
<td>String</td>
</tr>
<tr>
<td></td>
<td>arrival</td>
<td>Arrival</td>
<td>String</td>
</tr>
<tr>
<td></td>
<td>departure</td>
<td>Departure</td>
<td>String</td>
</tr>
<tr>
<td></td>
<td>airport</td>
<td>Airport</td>
<td>String</td>
</tr>
<tr>
<td></td>
<td>transport</td>
<td>Means of transportation</td>
<td>String</td>
</tr>
</tbody>
</table>
Returns
array —

The following return values are defined:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Data Type</th>
<th>Field Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>errorcode</td>
<td>0</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>errormessage</td>
<td>OK</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>errorfields</td>
<td>-</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If an error occurs, the following return values are defined:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Data Type</th>
<th>Field Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>errorcode</td>
<td>Contains an error message</td>
<td>Integer</td>
<td>Variable</td>
</tr>
<tr>
<td>errormessage</td>
<td>Describes the error</td>
<td>String</td>
<td>Variable</td>
</tr>
<tr>
<td>errorfields</td>
<td>Elements that caused the error</td>
<td>Array</td>
<td></td>
</tr>
</tbody>
</table>

### 5.1.3 tickets_get_ticket_personalization()

tickets_set_ticket_personalization()

tickets_set_ticket_personalization(array $params) : array

- Function for personalizing a ticket
- This function personalizes a ticket using a customer ID or using the personal data that has been provided.
- In this case, the system checks that the session customer is the ticket holder and that the ticket is not blocked or canceled.
- The ticket must also not be within the ticket transfer or resale process.
- The event is used to define which details are mandatory for personalization and these details are also checked by the function.
- Purchasing authorizations are not currently taken into account or evaluated within this function.

Parameters
### Array with function-specific parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Data Type</th>
<th>Field Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>sessionid</td>
<td>Session ID</td>
<td>String</td>
<td>20</td>
</tr>
<tr>
<td>ticketid</td>
<td>Ticket ID</td>
<td>String</td>
<td>20</td>
</tr>
</tbody>
</table>

### The following function parameters are defined:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Data Type</th>
<th>Field Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>errorcode</td>
<td>0</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>errormessage</td>
<td>OK</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>errorfields</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>Last name (mandatory if no customer ID)</td>
<td>String</td>
<td>Variable</td>
</tr>
<tr>
<td>firstname</td>
<td>First name (mandatory if no customer ID)</td>
<td>String</td>
<td>Variable</td>
</tr>
<tr>
<td>greeting</td>
<td>Salutation (2 Mr., 3 Ms., 4 company)</td>
<td>Integer</td>
<td>1</td>
</tr>
<tr>
<td>street</td>
<td>Street</td>
<td>String</td>
<td>Variable</td>
</tr>
<tr>
<td>housenr</td>
<td>House number</td>
<td>String</td>
<td>Variable</td>
</tr>
<tr>
<td>city</td>
<td>City/location</td>
<td>String</td>
<td>Variable</td>
</tr>
<tr>
<td>zip</td>
<td>Zip code</td>
<td>String</td>
<td>Variable</td>
</tr>
<tr>
<td>country</td>
<td>Country</td>
<td>String</td>
<td>Variable</td>
</tr>
<tr>
<td>nationality</td>
<td>Nationality</td>
<td>String</td>
<td>Variable</td>
</tr>
<tr>
<td>birthday</td>
<td>Date of Birth</td>
<td>String</td>
<td>Variable</td>
</tr>
<tr>
<td>arrival</td>
<td>Arrival</td>
<td>String</td>
<td>Variable</td>
</tr>
<tr>
<td>departure</td>
<td>Departure</td>
<td>String</td>
<td>Variable</td>
</tr>
<tr>
<td>airport</td>
<td>Airport</td>
<td>String</td>
<td>Variable</td>
</tr>
</tbody>
</table>
### 5.1.4 tickets_get_ticketlist_customertid()

```php
tickets_get_ticketlist_customertid()
tickets_get_ticketlist_customertid(array $params) : array
```

List of ticket data

This function returns a list of ticket data for the customer assigned to the session.

The system returns all ticket data for each ticket that is determined.

In order to be listed, the tickets must have been booked within the last 12 months.

#### Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Data Type</th>
<th>Field Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>sessionid</td>
<td>Session ID</td>
<td>String</td>
<td>20</td>
</tr>
</tbody>
</table>

#### Returns

- **array**
  - Ticket data

If successful, ticketlistcustomer = array() with the following keys:
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Data Type</th>
<th>Field Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Ticket number</td>
<td>String</td>
<td>20</td>
</tr>
<tr>
<td>customerid</td>
<td>Cust. no.</td>
<td>String</td>
<td>20</td>
</tr>
<tr>
<td>date</td>
<td>Booking date with time YYYYMMDDhhmm</td>
<td>Integer</td>
<td>12</td>
</tr>
<tr>
<td>state</td>
<td>Booking status</td>
<td>String</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>• B = paid</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• R = reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Z = returned</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• A = declined</td>
<td></td>
<td></td>
</tr>
<tr>
<td>price</td>
<td>Ticket price with currency code</td>
<td>String</td>
<td>14</td>
</tr>
<tr>
<td>pricedesc</td>
<td>Price name</td>
<td>String</td>
<td>Variable (text)</td>
</tr>
<tr>
<td>reductiondesc</td>
<td>Reduction name</td>
<td>String</td>
<td>Variable (text)</td>
</tr>
<tr>
<td>reductionid</td>
<td>ID of the reduction</td>
<td>Integer</td>
<td>10</td>
</tr>
<tr>
<td>performance</td>
<td>Event ID</td>
<td>Integer</td>
<td>10</td>
</tr>
<tr>
<td>performancename</td>
<td>Name of the event</td>
<td>String</td>
<td>Variable (text)</td>
</tr>
<tr>
<td>performancenameshop</td>
<td>Event name in the shop</td>
<td>String</td>
<td>Variable (text)</td>
</tr>
<tr>
<td>performancedesc</td>
<td>Event description</td>
<td>String</td>
<td>Variable (text)</td>
</tr>
<tr>
<td>performancedate</td>
<td>Event day</td>
<td>String</td>
<td>8</td>
</tr>
<tr>
<td>performancedateformatted</td>
<td>Event date in formatted/localized form</td>
<td>String</td>
<td>Variable</td>
</tr>
<tr>
<td>performancetime</td>
<td>Time of event</td>
<td>String</td>
<td>4</td>
</tr>
<tr>
<td>block</td>
<td>Section</td>
<td>Integer</td>
<td>10</td>
</tr>
<tr>
<td>blockdesc</td>
<td>Section name</td>
<td>String</td>
<td>80</td>
</tr>
<tr>
<td>row</td>
<td>Seat row</td>
<td>String</td>
<td>Variable (text)</td>
</tr>
<tr>
<td>seat</td>
<td>Seat</td>
<td>String</td>
<td>Variable (text)</td>
</tr>
<tr>
<td>defaultprice</td>
<td>Default price</td>
<td>String</td>
<td>14</td>
</tr>
<tr>
<td>invoice</td>
<td>Invoice number at booking</td>
<td>String</td>
<td>20</td>
</tr>
</tbody>
</table>
### Parameter Description Table

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Data Type</th>
<th>Field Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>option</td>
<td>Invoice number at reservation</td>
<td>String</td>
<td>20</td>
</tr>
<tr>
<td>return</td>
<td>Cancellation invoice no.</td>
<td>String</td>
<td>20</td>
</tr>
<tr>
<td>barcodeid</td>
<td>Bar code</td>
<td>String</td>
<td>Variable (text)</td>
</tr>
<tr>
<td>rfidkey</td>
<td>RFID key</td>
<td>String</td>
<td>20</td>
</tr>
<tr>
<td>voucherid</td>
<td>Coupon number</td>
<td>String</td>
<td>255</td>
</tr>
<tr>
<td>entrytime</td>
<td>Event admission time</td>
<td>String</td>
<td>4</td>
</tr>
<tr>
<td>entrydesc</td>
<td>Name of entrance</td>
<td>String</td>
<td>30</td>
</tr>
<tr>
<td>errormessage</td>
<td>OK</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In case of an error, the following return values are defined:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Data Type</th>
<th>Field Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>errorcode</td>
<td>Contains an error message</td>
<td>Integer</td>
<td>Variable</td>
</tr>
<tr>
<td>errormessage</td>
<td>Describes the error</td>
<td>String</td>
<td>Variable</td>
</tr>
<tr>
<td>errorfields</td>
<td>Elements that caused the error</td>
<td>Array</td>
<td></td>
</tr>
</tbody>
</table>

### 5.1.5 tickets_forward_ticket()

**tickets_forward_ticket()**

tickets_forward_ticket() (array $params) : array

Forward Ticket

- This function provides a service for forwarding tickets.
- All parameters are required parameters. To forward a ticket successfully, a valid session must exist.
- The ticket purchaser must be a session customer. A ticket transfer must be set up for the event.
- The ticket must not be included in an open transfer or in a resale process.
- A ticket layout for Print@Home and/or Passbook must have been defined for the event.

**Parameters**
Array with function-specific parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Data Type</th>
<th>Field Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>sessionid</td>
<td>Session ID</td>
<td>String</td>
<td>20</td>
</tr>
<tr>
<td>ticketid</td>
<td>Ticket ID</td>
<td>String</td>
<td>20</td>
</tr>
<tr>
<td>name</td>
<td>Last name</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>firstname</td>
<td>First name</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>email</td>
<td>Email address</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>

The following parameters are defined:

- **$params**

The following return values are defined:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Data Type</th>
<th>Field Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>errorcode</td>
<td>0</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>errormessage</td>
<td>OK</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>errorfields</td>
<td>Array</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In case of an error, the following return values are defined:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Data Type</th>
<th>Field Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>errorcode</td>
<td>Contains an error message</td>
<td>Integer</td>
<td>Variable</td>
</tr>
<tr>
<td>errormessage</td>
<td>Describes the error</td>
<td>String</td>
<td>Variable</td>
</tr>
<tr>
<td>errorfields</td>
<td>Elements that caused the error</td>
<td>Array</td>
<td></td>
</tr>
</tbody>
</table>

5.1.6 **tickets_delete_ticket_personalization()**

tickets_delete_ticket_personalization()
tickets_delete_ticket_personalization(array $params) : array

Function for canceling the personalization of a ticket. This function cancels the personalization of a ticket. The system checks that the session customer is the owner of the ticket and that the session store is the company for
the booking.

Parameters

<table>
<thead>
<tr>
<th>array</th>
<th>$params</th>
<th>Array with function-specific parameters</th>
<th>The following function parameters are defined:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Data Type</th>
<th>Field Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>sessionid</td>
<td>Session ID</td>
<td>String</td>
<td>20</td>
</tr>
<tr>
<td>ticketid</td>
<td>Ticket ID</td>
<td>String</td>
<td>20</td>
</tr>
</tbody>
</table>

Returns

array —

The following return values are defined:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Data Type</th>
<th>Field Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>errorcode</td>
<td>0</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>errormessage</td>
<td>OK</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>errorfields</td>
<td>-</td>
<td>Array</td>
<td></td>
</tr>
</tbody>
</table>

If an error occurs, the following return values are defined:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Data Type</th>
<th>Field Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>errorcode</td>
<td>Contains an error message</td>
<td>Integer</td>
<td>Variable</td>
</tr>
<tr>
<td>errormessage</td>
<td>Describes the error</td>
<td>String</td>
<td>Variable</td>
</tr>
<tr>
<td>errorfields</td>
<td>Elements that caused the error</td>
<td>Array</td>
<td></td>
</tr>
</tbody>
</table>

5.2 Extended Functions for XMLRPC

The XMLRPC functions tickets_get_enhancedperformance and tickets_get_perfprices have been extended.
The function has been extended to include the output of quantity allocations.

- Quantity allocation per customer
- Quantity allocation per shopping cart
- Quantity allocation per company
- Quantity allocation per event

The function tickets_get_perprices was extended for each ticket type.

- Number of tickets per customer per event
- Number of tickets per event
- Number of seats
- Minimum group size
- Fixed group size

**Adjustment of the XMLRPC function tickets_get_enhancedperformance()**

The documentation for the function has been adjusted. The “notbookable” parameter does not refer to the event, but rather to the client separation for the companies (that is, whether an event can be booked for a company).

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Data Type</th>
<th>Field Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>sessionid</td>
<td>Session ID</td>
<td>String</td>
<td>20</td>
</tr>
<tr>
<td>transactionid</td>
<td>Transaction ID</td>
<td>Integer</td>
<td>10</td>
</tr>
<tr>
<td>performanceid</td>
<td>Event ID</td>
<td>Integer</td>
<td>10</td>
</tr>
<tr>
<td>notbookable</td>
<td>Also provide data for events that cannot be booked from the perspective of the session store (Client separation), (optional, value 0/1, 1 = perform client check)</td>
<td>Integer</td>
<td>1</td>
</tr>
<tr>
<td>ok_translation_data</td>
<td>OK data, i.e. original event IDs are fetched</td>
<td>Integer</td>
<td>1</td>
</tr>
</tbody>
</table>
The following function parameters are defined:

- $params: Array with function-specific parameters
- ok_system_id: System from which import originated (optional)
- featureToggleQuota: Extension of output to include quantity allocation

### 5.2.1 tickets_get_ticket()

tickets_get_ticket()
tickets_get_ticket(array $params) : array

*Ticket data*
This function returns all data of a ticket.

**Parameters**

<table>
<thead>
<tr>
<th>array</th>
<th>$params</th>
<th>Array with function-specific parameters</th>
<th>The following function parameters are defined:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Parameter Description Data Type Field Length</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>sessionid Session ID String 20</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ticketid Ticket number String 20</td>
</tr>
</tbody>
</table>

**Returns**

array —

Ticket data

If successful, ticket = array() with the following keys:
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Data Type</th>
<th>Field Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Ticket number</td>
<td>String</td>
<td>20</td>
</tr>
<tr>
<td>customerid</td>
<td>Cust. no.</td>
<td>String</td>
<td>20</td>
</tr>
<tr>
<td>date</td>
<td>Booking date with time YYYYMMDDhhmm</td>
<td>Integer</td>
<td>12</td>
</tr>
<tr>
<td>state</td>
<td>Booking status</td>
<td>String</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>B = paid</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>R = reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Z = returned</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>A = declined</td>
<td></td>
<td></td>
</tr>
<tr>
<td>price</td>
<td>Ticket price with currency code</td>
<td>String</td>
<td>14</td>
</tr>
<tr>
<td>pricedesc</td>
<td>Price name</td>
<td>String</td>
<td>Variable (text)</td>
</tr>
<tr>
<td>reductiondesc</td>
<td>Reduction name</td>
<td>String</td>
<td>Variable (text)</td>
</tr>
<tr>
<td>reductionid</td>
<td>ID of the reduction</td>
<td>Integer</td>
<td>10</td>
</tr>
<tr>
<td>performance</td>
<td>Event ID</td>
<td>Integer</td>
<td>10</td>
</tr>
<tr>
<td>performancename</td>
<td>Name of the event</td>
<td>String</td>
<td>Variable (text)</td>
</tr>
<tr>
<td>performancenameshop</td>
<td>Event name in the shop</td>
<td>String</td>
<td>Variable (text)</td>
</tr>
<tr>
<td>performancedesc</td>
<td>Event description</td>
<td>String</td>
<td>Variable (text)</td>
</tr>
<tr>
<td>performancedate</td>
<td>Event day</td>
<td>String</td>
<td>8</td>
</tr>
<tr>
<td>performancedateformatted</td>
<td>Event date in formatted/localized form</td>
<td>String</td>
<td>Variable</td>
</tr>
<tr>
<td>performancetime</td>
<td>Time of event</td>
<td>String</td>
<td>4</td>
</tr>
<tr>
<td>block</td>
<td>Section</td>
<td>Integer</td>
<td>10</td>
</tr>
<tr>
<td>blockdesc</td>
<td>Section name</td>
<td>String</td>
<td>80</td>
</tr>
<tr>
<td>row</td>
<td>Seat row</td>
<td>String</td>
<td>Variable (text)</td>
</tr>
<tr>
<td>seat</td>
<td>Seat</td>
<td>String</td>
<td>Variable (text)</td>
</tr>
<tr>
<td>defaultprice</td>
<td>Default price</td>
<td>String</td>
<td>14</td>
</tr>
</tbody>
</table>
### Parameter Configuration

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Data Type</th>
<th>Field Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>invoice</td>
<td>Invoice number at booking</td>
<td>String</td>
<td>20</td>
</tr>
<tr>
<td>option</td>
<td>Invoice number at reservation</td>
<td>String</td>
<td>20</td>
</tr>
<tr>
<td>return</td>
<td>Cancelation invoice number</td>
<td>String</td>
<td>20</td>
</tr>
<tr>
<td>barcodeid</td>
<td>Bar code</td>
<td>String</td>
<td>Variable (text)</td>
</tr>
<tr>
<td>rfidkey</td>
<td>RFID key</td>
<td>String</td>
<td>20</td>
</tr>
<tr>
<td>voucherid</td>
<td>Coupon number</td>
<td>String</td>
<td>255</td>
</tr>
<tr>
<td>errormessage</td>
<td>OK</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Error Handling

In case of an error, the following return values are defined:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Data Type</th>
<th>Field Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>errorcode</td>
<td>Contains an error message</td>
<td>Integer</td>
<td>Variable</td>
</tr>
<tr>
<td>errormessage</td>
<td>Describes the error</td>
<td>String</td>
<td>Variable</td>
</tr>
<tr>
<td>errorfields</td>
<td>Elements that caused the error</td>
<td>Array</td>
<td></td>
</tr>
</tbody>
</table>