

Handling credit cards

Last Updated: 2021-03-04

Sterling Call Center enable you to validate the credit card payment method. Based on the configuration, the credit card payment method can be validated through Sterling Call Center, , or SSDCS.

Sterling Call Center also enable you to configure the way credit card names are displayed.

Credit card validation using Sterling Call Center

All credit cards follow a pattern in their number depending on their type. For example, the first two digits of all MasterCard credit cards are between 51 and 55, and the total length of a MasterCard card number is 16. An American Express card number has either 34 or 37 as its first two digits, and has a total length of 15.

Additionally, all numbers of a credit card follow a pattern, which can be validated with Luhn's algorithm. However, this algorithm only ensures that the given credit card number is a valid number. It does not ensure that it exists in any bank database.

It is useful to run this validation every time a new credit card number is added to save payment processing validations, which will fail every time against an invalid number.

Sterling Call Center provides functionality to process real-time credit card authorizations while confirming draft orders.

Credit card validation using the Sterling Sensitive Data Capture Server

Sterling Call Center enables you to validate a credit card payment method using SSDCS in order to comply with the PA-DSS. For more information about configuring SSDCS, refer to the *Sterling Selling and Fulfillment Foundation: Sterling Sensitive Data Capture Server, Release 1.1: Configuration Guide*.

During order creation, when a user is adding or modifying a credit card payment method, the Sterling Call Center application will verify whether the value of the `yfs.ssdcs.tokenize.cc` property is Y, if the value is Y, the `createAccessToken` API is called to generate an authentication token which in turn is passed to SSDCS. A browser, which is embedded in the Credit Card # field is opened by SSDCS with a single entry field for the credit card number. During installation, you can customize the entry field provided by the embedded browser to suit your preferences.

The user first selects a credit card type from the Credit Card Type drop-down list. The user then enters the credit card number in the Credit Card # field. When the user tabs out of the embedded browser, the credit card number is automatically passed to SSDCS, which in turn validates the credit card. If validation succeeds, it returns a generated credit card token, a display number, and a credit card type to the Sterling Call Center application. The display number is displayed instead of the credit card number in the Credit Card # field. If the credit card type returned by SSDCS is not the credit card type that was selected by the user, the Credit Card Type field will be updated to reflect the credit card type returned by SSDCS. If the credit card validation fails, an appropriate error message is displayed, indicating that the credit card number that has been entered is invalid. The error message can be customized through SSDCS. Also, the error message will be subjected to run through the localization bundle.

For more information about credit card validation using SSDCS, refer to the *Sterling Selling and Fulfillment Foundation: Sterling Sensitive Data Capture Server, Release 1.1: Configuration Guide*.

Displaying credit card name

Sterling Call Center enables you to configure whether a text field or multiple text fields should be used for entering credit card name. For more information about displaying credit card name, see the [Configuring supplied and add-on solutions](#).

Note: Sterling Call Center does not allow you to retrieve sensitive payment information such as credit card number, SVC number, customer's account number.

Note: Ensure that the Display Credit Card Name rule is same for an enterprise for which you are creating an order, and for the customers defined for an enterprise.

The credit card number is validated as follows:

- Prefix and Length Validation
- Luhn's Algorithm

Prefix and length validation

The table below specifies the exact numbers to prefix for a card number of a specific card type, and how many digits the card number must contain. For example, American Express cards all start with either 34 or 37 as their first digits, and contain 15 digits.

Note: The JCB credit card type has two entries in the table below. This is due to the fact that a JCB card can follow the prefix system listed in the table.

Table 1. Credit card types

Credit Card Type	Prefix	Length
American Express	34, or 37	15
MasterCard	51 through 55	16
Visa	4	13 or 16
Diners Club and Carte Blanche	36, 38, or 300 through 305	14
Discover	6011	16
JCB	2123 or 1800	15
JCB	3	16

When a credit card type is selected and a credit card number is entered in the Application Console, the system validates the number prefixed and the length. If both are valid, the system executes the Luhn's algorithm.

Luhn's algorithm

Luhn's algorithm determines whether or not a credit card number is valid. For a given credit card number:

1. Double the value of every other digit from right to left, beginning with the second to last digit.
2. Add the digits of the results of Step 1 to the remaining digits in the credit card number.
3. If the result mod 10 is equal to 0, the number is valid. If the result mod 10 is not equal to 0, the validation fails.

Example

A credit card of type Visa is added with the number 4624 7482 3324 9080.

This credit card number starts with 4, as it should be in all Visa cards. It also has 16 digits, which is valid for a Visa card. Now, execute Luhn's algorithm:

- a. Double the value of every other digit from right to left, beginning with the second to last digit.

4624 7482 3324 9080:

- $8 * 2 = 16$
- $9 * 2 = 18$

- $2 \times 2 = 4$
 - $3 \times 2 = 6$
 - $8 \times 2 = 16$
 - $7 \times 2 = 14$
 - $2 \times 2 = 4$
 - $4 \times 2 = 8$
- b. Add the digits of the results of the previous step to the remaining digits in the credit card number.
- The digits of the result of **Step a** amount to the following:
 - $1+6+1+8+4+6+1+6+1+4+4+8 = 50$
 - The remaining digits in the credit card number amount to the following:
 - $6+4+4+2+3+4+0+0 = 23$
 - The sum of the two sub results is:
 - $23+50 = 73$
- c. If the result mod 10 is equal to 0, the number is valid. If the result mod 10 is not equal to 0, the validation fails.
- $73 \bmod 10$ is 3, therefore the card number is not valid.

A Visa credit card of number 4624 7482 3324 9780, on the other hand, passes the validation.

i Note: You can override the validation for the credit card types listed in the table above by implementing YCDValidateCreditCardInfoUE user exit. If you are configuring new credit card types, the validation for the new credit card types should also be implemented by the YCDValidateCreditCardInfoUE user exit.

Parent topic:

→ [Change payment method solution](#)