

! INFO

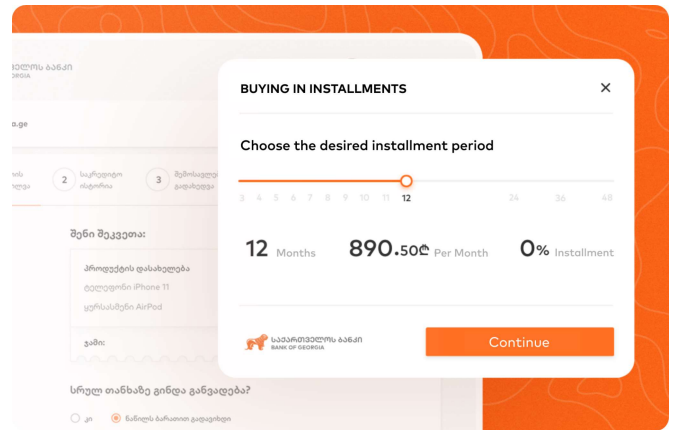
Integration of the installment plan model is necessary only in the case, if you want a consumer to choose the installment plan conditions on your page.

Online Installment Plan Modal

To choose the online installment plan conditions, we have prepared a special modal. The modal is a kind of loan calculator, where a customer, resulting from the basket cost, can see an approximate monthly payment and annual interest rate.

As the conditions are always individual and rely on the agreement concluded between the Bank of Georgia and Merchant (including a 0% installment plan condition), they differ from the standard conditions. Providing a borrower with this information is important. All the information is given in the modal, thus helping a customer in decision-making.

Respectively, prior to activation of the installment plan, a kind of loan calculator is necessary, which is an independent microservice and **needs integration** on the Merchant's site. The Bank of Georgia is offering a ready-made SDK, compatible with all devices (Desktop, Mobile, Tablet) and needs addition of a ready code fragment on the site.



Example:

APPLY FOR A LOAN

Integration

To begin the installment plan modal SDK integration, copy the code in HTML page. Put our `client_id` instead of the `{client_id}`.

```
<script src="https://webstatic.bog.ge/bog-sdk/bog-sdk.js?version=2&client_id={client_id}">
</script>
```

Modal demonstration

To show the installment plan modal use the `BOG.Calculator.open` method.

Parameters

`amount` **REQUIRED** `number`

Installment plan amount

`bnpl` optional `boolean`

The parameter defines whether to show the `standard installment plan` or/and `payment in installments` to a customer.

- `true` - only the payment in installments will be shown to a customer
- `false` - only the standard installment plan will be shown to a customer
- `undefined` | `null` - both methods will be shown to a customer.

`onClose` optional `function`

The method, called by the SDK, if a customer closes the modal.

`onRequest` optional `function`

The method, called by the SDK, upon selecting the conditions by a customer. At this time the installment plan order should be **requested**. The `config.loan` parameter should be set.

```
BOG.Calculator.open({
  amount: 500,
  onClose: () => {
    // Modal close callback
  },
  onRequest: (selected, successCb,
closeCb) => {
    const {
      amount, month, discount_code
    } = selected;
    fetch('url-to-backend-api', {
      method: 'POST',
      body:
JSON.stringify(selected)
    })
      .then(response =>
response.json())
      .then(data =>
successCb(data.orderId))
      .catch(err => closeCb());
  },
  onComplete: ({redirectUrl}) => {
    return false;
  }
})
```

The following arguments will be transmitted to the method:

- `selected` - the object, into which there will be transmitted the data selected by a customer.
 - `amount` - the amount to be paid on monthly basis
 - `month` - number of the loan months
 - `discount_code` - the discount code
 - `successCb` - the method to be called, if you wish the SDK to continue the installment plan process.
 - `closeCb` - the method to be called, if you wish to close the modal.
-

`onComplete` optional `function`

The method being called by the SDK upon completion of the online installment plan process. At this time a customer is automatically re-directed to the Merchant's success link. If you do not want the re-direction, return the `false` meaning (`return false`;) in the method. The method will be given the object as an argument with the following parameters:

- `redirectUrl` - the link, where a customer is to be redirected to.
-

There are two methods of process continuation:

1. If you want the SDK to continue the online installment plan process, upon generation of `orderId` in the `onRequest` method, call `successCb` and transmit the **generated** `orderId` parameter to it. Upon completion of the process, the `onComplete` method will be called.
2. If the SDK is needed only for a calculator and want a customer to

continue the process after selection of the desirable condition, return the `false` meaning (`return false;`) into the `onRequest` method and the SDK will not continue the process. At this time the modal remains open. If you wish to close the modal, use the `closeCb` or `Bog.Calculator.close()` method.

The SDK also gives you the opportunity to add the BoG button to your site. Transmit the DOM element to the method, where you wish to insert the button.

```
<div class="bog-smart-button">
<script>
  const button = BOG.SmartButton.render(document.querySelector('.bog-smart-button'), {
    text: 'loan request',
    onClick: () => {
      // Open Installment Calculator Here
    }
  })
</script>
```