

# Low code

- [Модули](#)
  - [Проверка идентификатора модуля в событии](#)
- [Примеры](#)
  - [Сложный фильтр](#)
  - [Ожидание загрузки всех скриптов](#)
- [Процессы](#)
  - [Ручка удаления процесса из списка](#)
  - [Запустить процесс и перейти в Задачи](#)
- [CRM](#)
  - [Фильтр для \\_\\_parentCompany](#)

??????

## Модули

???????? ???? ????  
????? ? ???? ?

При реализации обработчика событий в модуле иногда необходимо проверить, что событие это возникло именно для нашего модуля. В таком случае код скрипта будет выглядеть следующим образом:

```
async function action(): Promise<void> {  
    // Проверяем наш ли модуль  
    if (Namespace.code !== `ext_${Context.data.__item!.id}`)  
        return;  
  
    // Здесь уже выполняем нужные действия  
}
```

???????

## Примеры

???????? ???? ???

```
const leads = await Global.ns._clients.app._leads.search()
  .where((f, g) => {
    let operands: Filter[] = [];
    operands.push(f.__deletedAt.eq(null));
    operands.push(f.__createdAt.gte(Context.data.start_date!));
    operands.push(f.__createdAt.lte(Context.data.end_date!));
    if (Context.data.status?.length) {
      let filterStatus: Filter[] = [];
      for (const adress of Context.data.adresses) {
        filterStatus.push(f._address.eq(adress))
      }
      operands.push(g.or(...filterStatus))
    }
    if (Context.data.companyes?.length) {
      let filterChannel: Filter[] = [];
      for (const company of Context.data.companies) {
        filterChannel.push(f._companies.link(company))
      }
      operands.push(g.or(...filterChannel))
    }
    return g.and(...operands)
  })
  .size(10000)
  .all()
```

## Примеры

???????? ???? ????  
????????

```
$(function() {  
    // Загрузка внешних скриптов  
    const loadScript = url => new Promise(resolve => {  
        const script = document.createElement('script')  
        script.addEventListener('load', () => {  
            resolve()  
        })  
        script.src = url  
        document.body.append(script)  
    })  
  
    const srcIndex = loadScript('https://cdn.amcharts.com/lib/5/index.js')  
    const srcXY = loadScript('https://cdn.amcharts.com/lib/5/xy.js')  
    const srcAm = loadScript('https://cdn.amcharts.com/lib/5/themes/Animated.js')  
  
    Promise.all([srcIndex, srcXY, srcAm])  
        .then([srcIndex, srcXY, srcAm]) => {  
            console.log('Скрипт index.js загружен')  
            console.log('Скрипт xy.js загружен')  
            console.log('Скрипт Animated.js загружен')  
            amLoadScript()  
        })  
  
    function amLoadScript() {  
        //  
    }  
});
```

????????

## Процессы

????? ?????????? ?????????? ??  
???????

Если процесс не используется и его нужно удалить из списка в системе, можно воспользоваться "ручками" и следующим кодом:

```
const processID = "0a46f3fc-d78a-4afb-af4b-cd6ab0fab711"; //id процесса
const host = "http://elma36511.ru"; //Хост
const lockResp = await fetch(`${host}/api/bpm/templates/${processID}/lock`, {method: "POST"});
const lockData = await lockResp.json();
const deleteResp = await fetch(`${host}/api/processor/templates/${processID}`, {method:
"DELETE", headers: {"X-Lock-Hash": lockData.hash}});
```

## Процессы

????????? ???? ? ?????  
? ?????

Чтобы после запуска процесса, например по кнопке в виджете, а затем перескочить автоматически в список задач, необходимо реализовать следующий код в клиентском сценарии:

```
declare const window: any;

async function checkOutInvoices(): Promise<void> {
  await Namespace.processes.checking_payment_of_out_invoices.run({});
  window.location.replace('/tasks/income');
}
```

CRM

# ?????? ??? \_\_parentCompany

При использовании виджета Иерархия, если указать в качестве материнской компании саму себя наступает бесконечная рекурсия (хоть виджет это и отлавливает - он не покажет нам иерархию).

Чтобы решить данную проблему, при инициализации формы редактирования (при создании компании еще нет, так что необязательно) нужно добавить следующий код:

```
async function onInit() {  
  Context.fields._parentCompany.data.setFilter((f, c, g) => f.__id.neq(Context.data.__id))  
  Context.fields._childCompanies.data.setFilter((f, c, g) => f.__id.neq(Context.data.__id))  
}
```

После этого выбрать саму себя уже не получится.