Содержание

Принцип работы триггеров

> Триггеры пишутся на языке C#.

Где интерфейс настройки триггеров?

Триггеры – это отдельный подключаемый модуль. Чтобы получить к нему доступ, обратитесь к вашему менеджеру или в техническую поддержку.

Отфильтровать

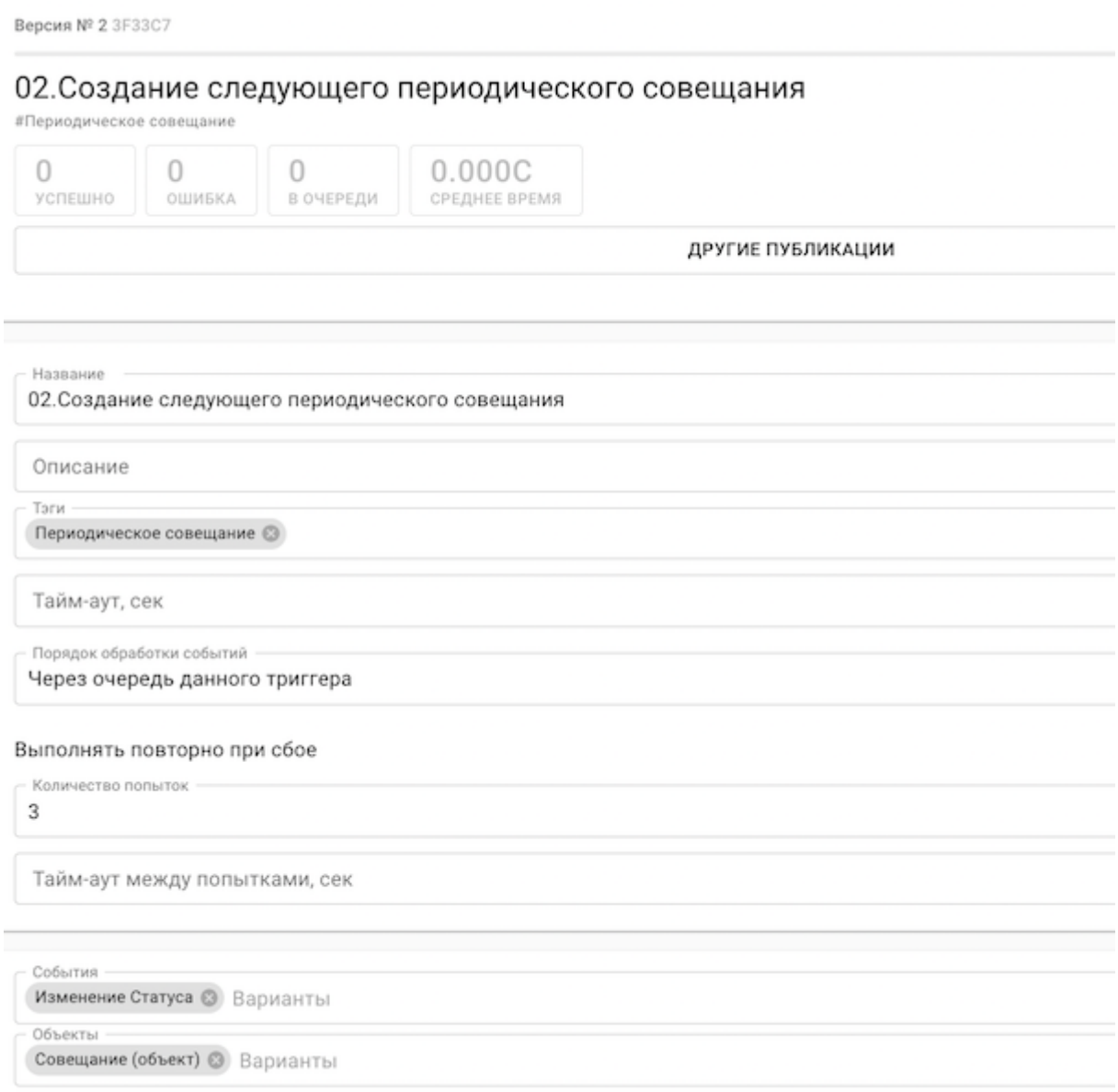
> Половина работы триггера: это понять, на какие события он должен реагировать.

Фильтрация событий проходит в несколько этапов с помощью 3 инструментов, которые дополняют друг друга.

1. Простой фильтр на события, который есть прямо в интерфейсе модуля триггеров.
2. LINQ-запрос(ы). Может как использоваться, так и не использоваться. Удобно, чтобы выбрать из большой базы данных какие-то специфичные объекты.
3. В рамках написания прописываются дополнительные условия – как раз здесь используется настроенный LINQ-запрос для последней тонкой фильтрации событий.

Как это выглядит в интерфейсе модуля триггеров:

Фильтрация на уровне интерфейса модуля триггеров:



Тонкая фильтрация в условии триггера:

Условие. 02.Создание следующего периодического совещания

```
var linqGetMeeting = "Trigger02.GetMeetingData";

TriggerConsole.WriteLine($"ID события - {Context.EventId}, отправлено - {Context.EventSentTime}, вызвал пользователь - {Context.PersonId}");
if (Event.NewStatus.ToString() != "Completed"){
    TriggerConsole.WriteLine($"Новый статус != Закрыто");
    return false;
}

// Получение Сделки и ее параметров
var jMeeting = await Api.Queries.Get(x => x
    .DataSourceKey(linqGetMeeting)
    .Parameters(p =>
    {
        p["ObjectId"] = $"'{Event.ProjectId}'";
    })
    .PageStartIndex(0)
    .PageSize(1)
    );
// Конвертируем в переменные результат запроса
TriggerConsole.WriteLine($"Результат запроса - {jMeeting}");
var jsonjMeeting = JsonConvert.DeserializeObject<List<Newtonsoft.Json.Linq.JObject>>(jMeeting);
TriggerConsole.WriteLine($"Число найденных Совещаний с повторением - '{jsonjMeeting.Count()}'");
if (jsonjMeeting.Count() == 0){
    TriggerConsole.WriteLine($"Не найдено совещания с периодом");
    return false;
}

// Сохраним переменные
Context["jMeeting"] = jMeeting;

return true;

```

Отреагировать

События, которые попали под условия фильтров, запускают написанный скрипт. Результатом работы скрипта могут быть небольшие вычисления, создание новых записей в справочнике, новых объектов, изменение статусов, изменение значений реквизитов, их блокировка и т.д.

Триггеры работают через интеграционное API ADVANTA. ⇒ Всё, что можно сделать через API, триггер может запустить в системе.

Как это выглядит в интерфейсе модуля триггеров: