

## Содержание

<b>Передаваемые параметры</b> .....	3
<b>Возвращаемые параметры</b> .....	3
<b>Пример вызова API «InsertTimerEvent» в Postman</b> .....	4
<b>Пример триггера для обработки события</b> .....	5



# Создание вызова в очереди событий

Метод: **InsertTimerEvent** Для планирования запуска бизнес-операции на определённый день и время служит API-функция «InsertTimerEvent». Описание этой API можно найти на странице службы «APIScheduledEvents». Относительный путь к странице этой службы на вашем стенде ADVANTA: «/components/services/APIScheduledEvents.asmx».

## Передаваемые параметры

Параметр	Описание
SessionID	ID сессии пользователя
CallTime	обязательный параметр типа DateTime в котором содержится дата и время, когда необходимо послать сообщение на шину.
ObjectID	GUID объекта, идентификатор, смысл которого будет известен при создании логики обработки, может быть идентификатором объекта дерева, записи справочника, пользователя и т.п. параметр не обязательный, если он не требуется, то вместо него должен допускаться null
OperationCode	обязательный целочисленный параметр с кодом операции. Код должен быть зарегистрирован в справочнике кодов операций, если его там нет, то ошибка «Код операции не зарегистрирован»
Parameters	объект типа JSON, в котором вызывающий может сохранить дополнительные параметры, которые будут необходимы при последующей обработке.

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <InsertTimerEvent xmlns="http://streamline/">
      <contract>
        <ASPNETSessionId>string</ASPNETSessionId>
        <CallTime>dateTime</CallTime>
        <ObjectID>guid</ObjectID>
        <OperationCode>int</OperationCode>
        <Parameters>string</Parameters>
      </contract>
    </InsertTimerEvent>
  </soap:Body>
</soap:Envelope>
```

## Возвращаемые параметры

Параметр	Описание
InsertTimerEventResponse	Признак успешного создания вызова

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xmlns:xsd="http://www.w3.org/2001/XMLSchema">  
  <soap:Body>  
    <InsertTimerEventResponse xmlns="http://streamline/" />  
  </soap:Body>  
</soap:Envelope>
```

## Пример вызова API «InsertTimerEvent» в Postman

Протестировать создание сообщения о выполнении бизнес-операции проще всего через систему Postman. Ниже приведены снимки настроек SOAP-запроса в ней:

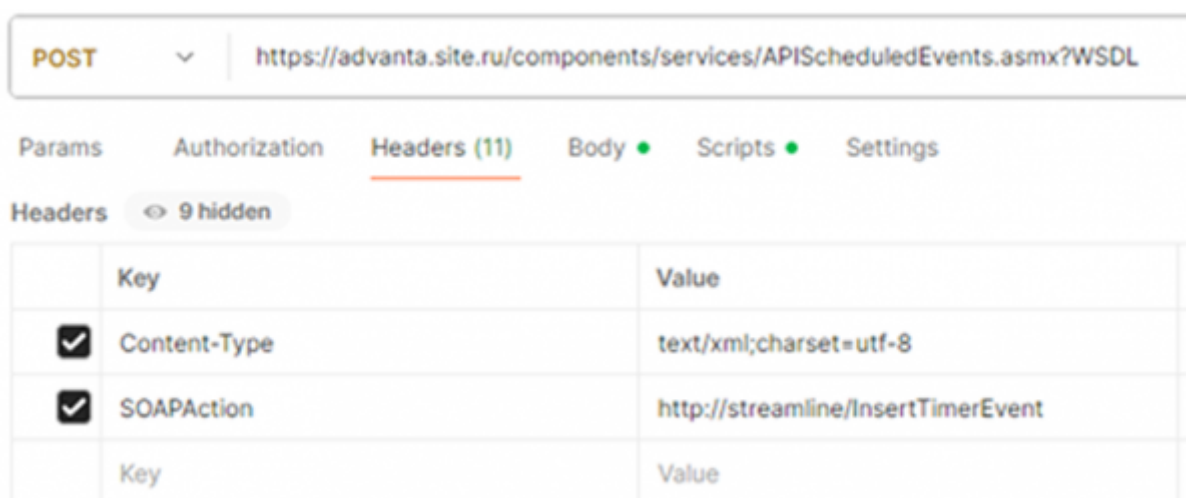


Рисунок 1. Настройки заголовков запроса (закладка «Headers»)

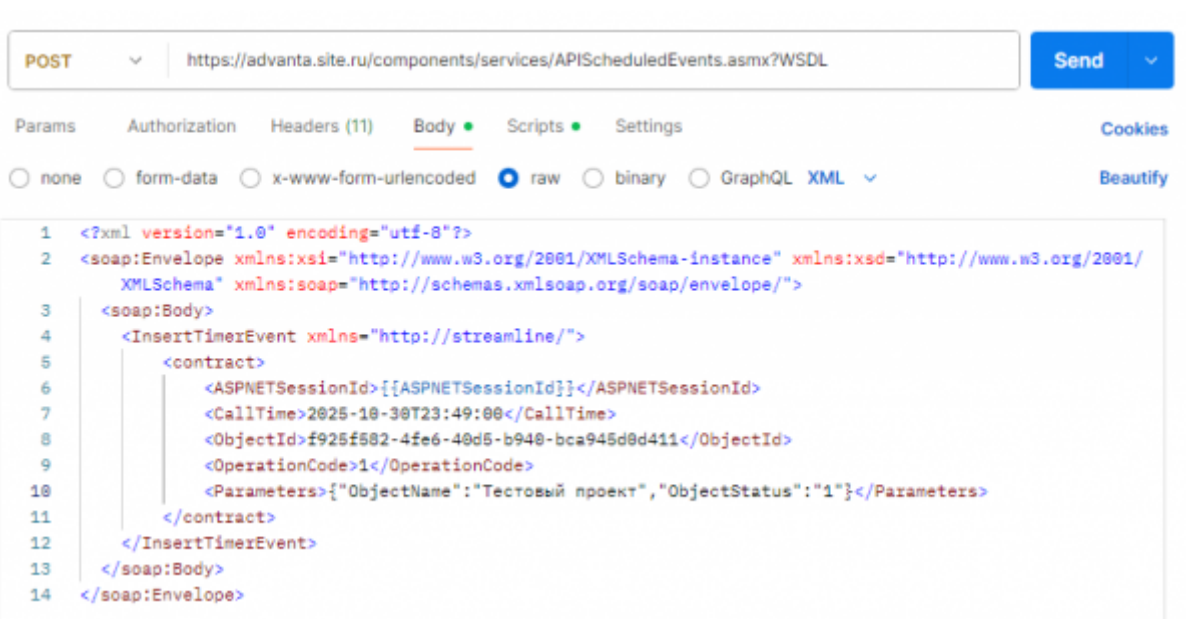


Рисунок 2. Настройки тела запроса (закладка «Body»)

## Пример триггера для обработки события

Проверить, что сообщение появилось в шине, можно с помощью триггера. Он должен реагировать на событие «Событие таймера, запланированное по подписке». Общие настройки триггера:

Версия № 24 4207F Опубликована в 15:11

### Пример - обработка события, помещённого в шину через API

0 УСПЕШНО    0 ОШИБКА    0 В ОЧЕРЕДИ    0,000С СРЕДНЕЕ ВРЕМЯ

ДРУГИЕ ПУБЛИКАЦИИ Системный Администратор

Название:

Описание:

Тэги:

Тайм-аут, сек:

Порядок обработки событий:

Выполнять повторно при сбое:

Не запускать если событие вызвано действием модуля:

События:

Рисунок 3. Настройки триггера

Код проверки «условия» запуска триггера:

```
TriggerConsole.WriteLine($"Проверка условия: ID события - {Context.EventId},  
отправлено - {Context.EventSentTime}");  
  
// Реагируем только на бизнес-операцию с кодом №1  
if (Event.BusinessOperationCode == 1){  
    TriggerConsole.WriteLine($"ID события таймера:  
{Event.BusinessOperationCode}");  
} else {  
    return false; // Если это не операция с кодом №1, не запускаем триггер  
}  
  
return true;
```

Код «скрипта». В данном примере просто выводим в консоль ID и время сообщения, а также параметры, переданные в API-функции «InsertTimerEvent»:

```
TriggerConsole.WriteLine($"ID события = {Context.EventId}");  
TriggerConsole.WriteLine($"Дата и время = {Context.EventSentTime}");  
  
// Код бизнес-операции:  
TriggerConsole.WriteLine($"BusinessOperationCode =  
{Event.BusinessOperationCode}");  
  
// UID объекта, указанный при вызове API InsertTimerEvent:  
TriggerConsole.WriteLine($"ObjectId = {Event.ObjectId}");  
  
// Параметры, переданные через API InsertTimerEvent:  
TriggerConsole.WriteLine($"Parameters = {Event.Parameters}");
```

В результате, в логге триггера будут видны переданные через API значения:

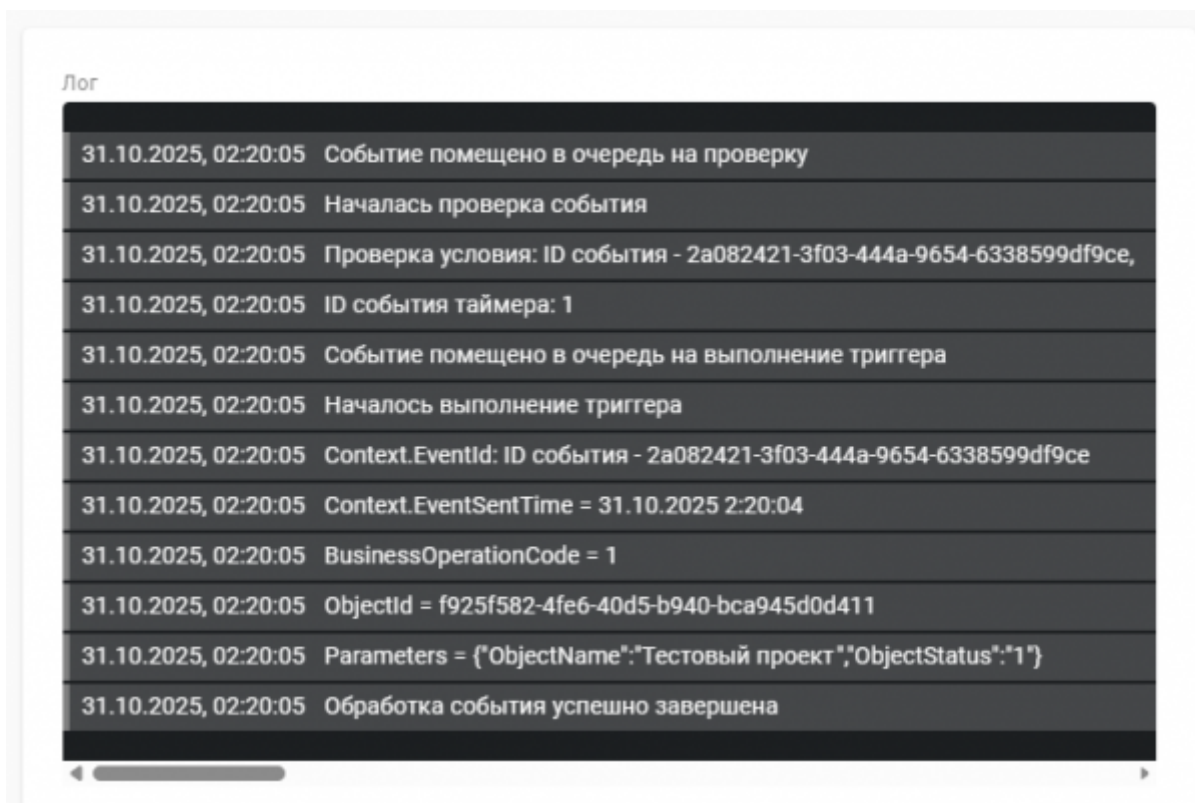


Рисунок 4. Лог триггера

From:  
<https://wiki.a2nta.ru/> - Wiki [3.x]

Permanent link:  
<https://wiki.a2nta.ru/doku.php/product/api/list/inserttimerevent?rev=1770962227>

Last update: **13.02.2026 05:57**

