

Содержание

Передаваемые параметры	3
Возвращаемые параметры	4
Пример вызова API «InsertTimerEvent» в Postman	4
Пример триггера для обработки события	5

Создание вызова в очереди событий

Метод: **InsertTimerEvent**

Позволяет запланировать запуск бизнес-операции на определённый день и время.

Примечание: Информацию о способе вызова также можно найти на странице службы «APIScheduledEvents» (относительный путь на стенде ADVANTA: «/components/services/APIScheduledEvents.asmx»).

Передаваемые параметры

Параметр	Тип	Назначение
ASPNETSessionId	string	Идентификатор сессии пользователя, от имени которой создается событие
CallTime	DateTime	Обязательный. Дата и время планируемого отправки сообщения на шину
ObjectId	Guid	

\\Необязательный. Допускается значение null|

OperationCode	int	Обязательный. Целочисленный код операции. Код должен быть предварительно зарегистрирован в соответствующем справочнике кодов операций. В противном случае метод вернет ошибку «Код операции не зарегистрирован»
Parameters	string	Строка в формате JSON , содержащая произвольные дополнительные параметры, необходимые для последующей обработки события.

```
<?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	Название	Описание	Тип данных
InsertTimerEventResponse	Ответ на добавление события	Подтверждение успешного создания события таймера	Контейнер

Примечания:

- Пустой элемент означает успешное выполнение операции (без дополнительных данных)
- Пространство имен XML: <http://streamline/>
- Код ответа HTTP 200 OK также подтверждает успешное выполнение

```
<?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:

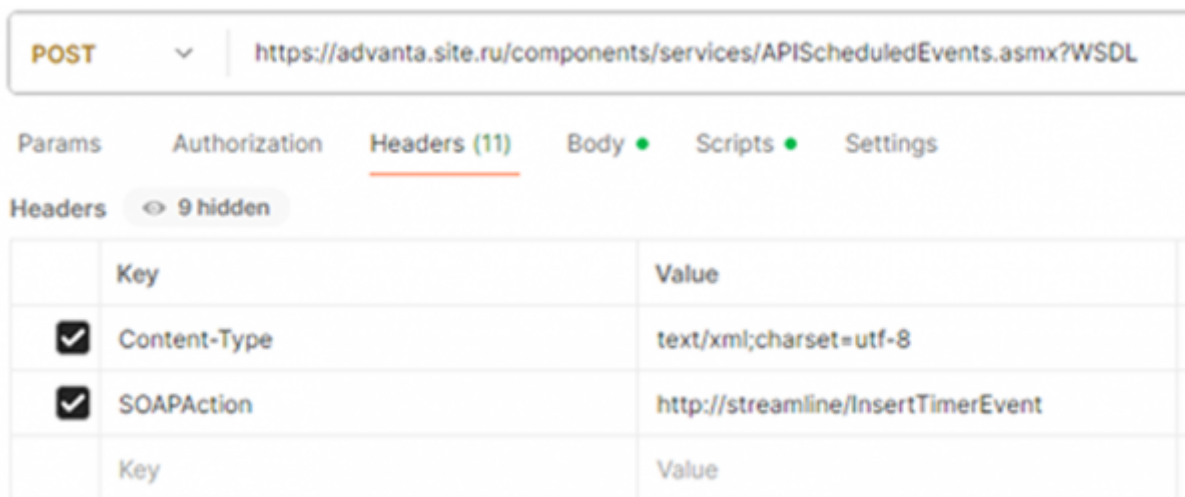


Рисунок 1. Настройки заголовков запроса (закладка «Headers»)

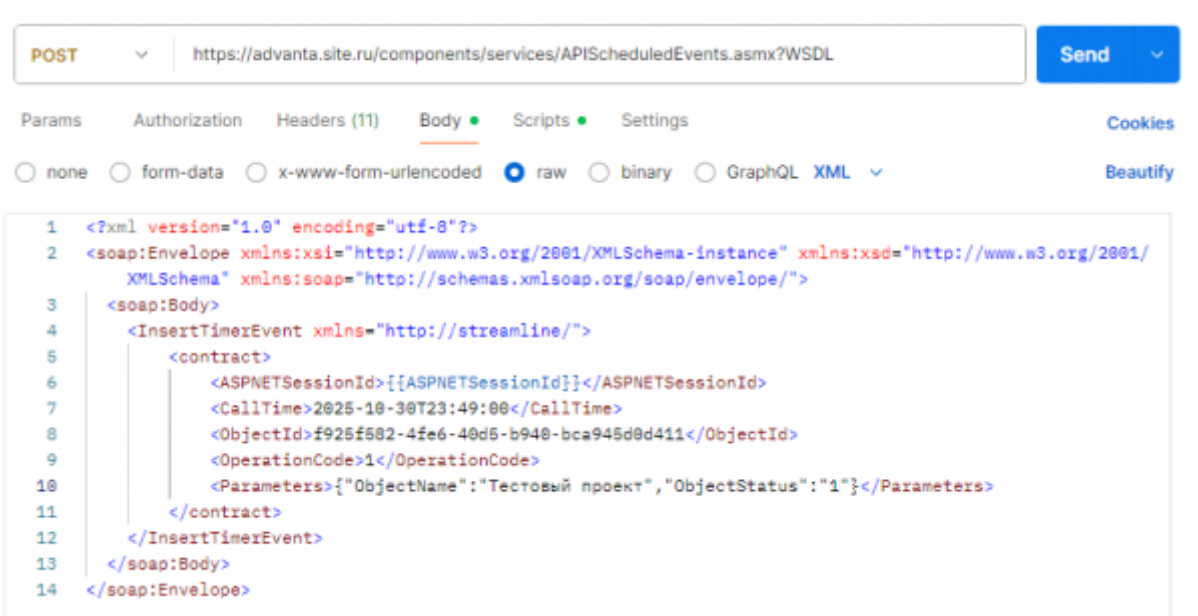


Рисунок 2. Настройки тела запроса (закладка «Body»)

Пример триггера для обработки события

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

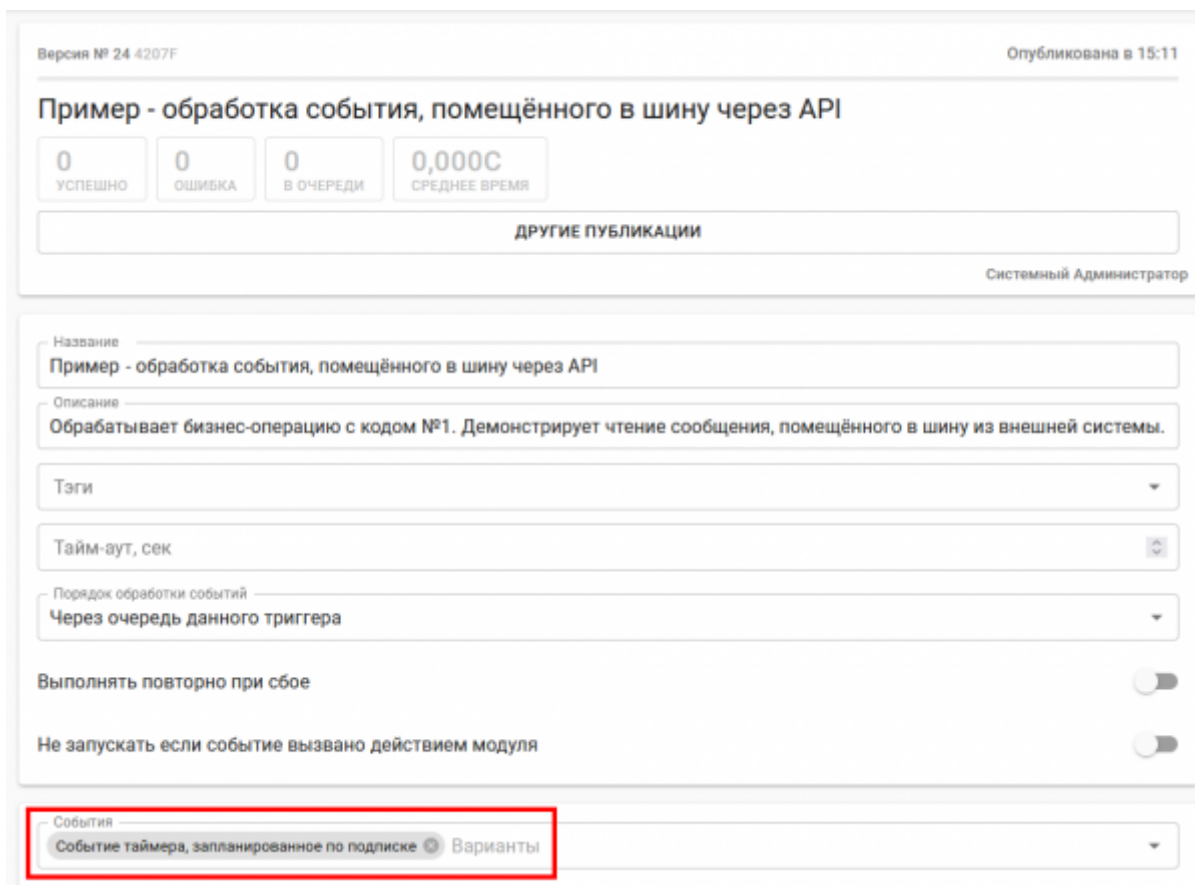


Рисунок 3. Настройки триггера

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 логе триггера будут видны переданные через API значения:

Лог

```
31.10.2025, 02:20:05 Событие помещено в очередь на проверку
31.10.2025, 02:20:05 Началась проверка события
31.10.2025, 02:20:05 Проверка условия: ID события - 2a082421-3f03-444a-9654-6338599df9ce,
31.10.2025, 02:20:05 ID события таймера: 1
31.10.2025, 02:20:05 Событие помещено в очередь на выполнение триггера
31.10.2025, 02:20:05 Началось выполнение триггера
31.10.2025, 02:20:05 Context.EventId: ID события - 2a082421-3f03-444a-9654-6338599df9ce
31.10.2025, 02:20:05 Context.EventSentTime = 31.10.2025 2:20:04
31.10.2025, 02:20:05 BusinessOperationCode = 1
31.10.2025, 02:20:05 Objectid = f925f582-4fe6-40d5-b940-bca945d0d411
31.10.2025, 02:20:05 Parameters = {"ObjectName": "Тестовый проект", "ObjectStatus": "1"}
31.10.2025, 02:20:05 Обработка события успешно завершена
```

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=1771328393>

Last update: **17.02.2026 11:39**

