

1. События

Доступ

Чтобы пользоваться API событий:

- В лицензии должен присутствовать модуль 2200 (прочие названия: API_EVENTS(2200), 'iikoAPI (ApiEvents)').
- У пользователя, под чьим именем осуществляется вход, должно быть право В_VTJ "Просматривать журнал событий".

Список событий

Версия iiko: 3.9

GET <https://host:port/resto/api/events>

Параметры запроса

Название	Значение	Описание
from_time	yyyy-MM-ddTHH:mm:ss.SSS	Время с которого запрашиваются события, в формате ISO: yyyy-MM-ddTHH:mm:ss.SSS, по умолчанию – начало текущих суток
to_time	yyyy-MM-ddTHH:mm:ss.SSS	Время по которое (не включительно) запрашиваются события в формате ISO: yyyy-MM-ddTHH:mm:ss.SSS,, по умолчанию граница не установлена
from_rev	Число	Ревизия, с которой запрашиваются события, число. Каждый ответ содержит тэг revision, значение которого соответствует ревизии, по которую включительно отданы события; при новых запросах следует использовать revision + 1 (revision из предыдущего ответа) для получения только новых событий. В штатном режиме одно и тоже событие повторно с разными ревизиями не приходит, однако такой гарантии не даётся. ID (UUID) события уникален, может использоваться в качестве ключа.

Что в ответе

Список событий в формате *eventsList* (см. XSD Список событий).

Поле *<type>* использует значение поля *<id>* структуры *groupsList*.

Примеры запроса

```
https://localhost:8080/resto/api/events?key=39c4c88f-4758-64e3-3b4c-ad6f589c0dc2
```

```
https://localhost:8080/resto/api/events?key=46dcaad5-b139-b976-88a2-1cf9b567195c&from_time=2014-06-16T00:00:00.000&to_time=2014-06-18T23:59:59.999
```

```
https://localhost:8181/resto/api/events?key=3192935b-26f7-4d61-b11d-58feb1a7d69b&from_rev=10016007
```

Список событий по фильтру событий и номеру заказа

Версия iiko: 5.0

POST <https://host:port/resto/api/events>

Тело запроса

Список id событий, по которым производится фильтрация (application/xml).

[Copy Code](#)
XML

```
<eventsRequestData>
<events>
<event>orderCancelPrecheque</event>
<event>orderPaid</event>
</events>
<orderNums>
<orderNum>175658</orderNum>
</orderNums>
</eventsRequestData>
```

Что в ответе

Список событий в формате *eventsList* (см. XSD Список событий).

Поле *<type>* использует значение поля *<id>* структуры *groupsList*.

Сохранение событий

Версия iiko: 8.0.3

POST <https://host:port/resto/api/events/add>

Тело запроса

Список событий в формате *eventsList*.

[Copy Code](#)

Код

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<eventsList>
  <event>
    <date>2021-12-31T19:24:29.033+03:00</date>
    <type>banana</type>
    <departmentId>2</departmentId>
    <attribute>
      <name>userName</name>
      <value>Администратор</value>
      <type>java.lang.String</type>
    </attribute>
    <attribute>
      <name>user</name>
      <value>c831367e-778f-e80f-18f7-bd0843cd10c6</value>
      <type>User</type>
    </attribute>
    <attribute>
      <name>success</name>
      <value>1.000000000</value>
      <type>java.lang.Boolean</type>
    </attribute>
  </event>
  <event>
    <date>2021-12-31T19:25:29.033+03:00</date>
    <type>banana</type>
    <departmentId>2</departmentId>
    <attribute>
      <name>userName</name>
      <value>Администратор</value>
      <type>java.lang.String</type>
    </attribute>
    <attribute>
      <name>user</name>
      <value>c831367e-778f-e80f-18f7-bd0843cd10c6</value>
      <type>User</type>
    </attribute>
    <attribute>
      <name>success</name>
      <value>1.000000000</value>
      <type>java.lang.Boolean</type>
    </attribute>
  </event>
</eventsList>
```

```
</event>
</eventsList>
```

Что в ответе

Список событий в формате *eventsList*.

Copy Code

Код

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<eventsList>
  <event>
    <id>9a48d9d4-8196-40e7-017e-e94b95b20023</id>
    <date>2021-12-31T19:24:29.033+03:00</date>
    <type>banana</type>
    <departmentId>2</departmentId>
    <attribute>
      <name>user</name>
      <value>c831367e-778f-e80f-18f7-bd0843cd10c6</value>
      <type>User</type>
    </attribute>
    <attribute>
      <name>success</name>
      <value>1</value>
      <type>java.lang.Boolean</type>
    </attribute>
    <attribute>
      <name>userName</name>
      <value>Администратор</value>
      <type>java.lang.String</type>
    </attribute>
  </event>
  <event>
    <id>9a48d9d4-8196-40e7-017e-e94b95b20024</id>
    <date>2021-12-31T19:25:29.033+03:00</date>
    <type>banana</type>
    <departmentId>2</departmentId>
    <attribute>
      <name>user</name>
      <value>c831367e-778f-e80f-18f7-bd0843cd10c6</value>
      <type>User</type>
    </attribute>
    <attribute>
      <name>userName</name>
      <value>Администратор</value>
      <type>java.lang.String</type>
    </attribute>
  </event>
</eventsList>
```

```
</attribute>
<attribute>
  <name>success</name>
  <value>1</value>
  <type>java.lang.Boolean</type>
</attribute>
</event>
</eventsList>
```

Описание

Тип события может быть любым (с количеством символов не больше 255), но если он зарегистрирован в events.xml, то событие должно содержать все атрибуты, обязательные для данного типа.

У атрибута добавилось поле 'type'. Значения, которые может принимать это поле, перечислены в таблице. Тип атрибута накладывает ограничения на поле 'value'.

type	value
java.lang.Boolean	"0" - false, "1" - true
java.lang.String	Текст
java.util.Date	Дата в формате "yyyy-MM-dd'T'HH:mm:ss.SSS"
resto.db.Guid	UUID
Наследники java.lang.Number. Примеры: java.lang.Integer, java.math.BigDecimal	Число.
Наследники resto.db.CachedEntity в простой форме. Примеры: User, Department, Terminal	UUID соответствующего справочника.

В таблице UserEventAttribute для поля 'value' есть 4 колонки (valueDate, valueGuid, valueNumber, valueString) разных типов, и, в зависимости от типа атрибута, значение будет записано в одну из этих колонок.

Дерево событий

Версия iiko: 3.9

GET <https://host:port/resto/api/events/metadata>

Что в ответе

Дерево событий в формате *groupsList* (см. XSD Дерево событий).

Возвращает дерево событий (аналог иерархии событий журнала событий в iikoOffice). Поле `<id>` используется как значение поля `<type>` структуры *eventList*. Также структура определяет список атрибутов, специфичных для данного события.

Пример запроса

```
https://localhost:8080/resto/api/events/metadata?key=43d272ed-cfb1-64b1-9842-f3078cd69172
```

Дерево событий по фильтру

Версия iiko: 5.0

POST <https://host:port/resto/api/events/metadata>

Тело запроса

Список id событий, по которым производится фильтрация (application/xml).

[Copy Code](#)
XML

```
<eventsRequestData>
  <events>
    <event>orderOpened</event>
  </events>
</eventsRequestData>
```

Что в ответе

Дерево событий в формате *groupsList* (см. XSD Дерево событий).

Возвращает дерево событий (аналог иерархии событий журнала событий в iikoOffice). Поле `<id>` используется как значение поля `<type>` структуры *eventList*. Также структура определяет список атрибутов, специфичных для данного события.

Пример запроса

```
https://localhost:8080/resto/api/events/metadata?key=933a2e7c-be36-d2ee-0b96-9509e6f24fa1
```

Информация о кассовых сменах

Версия iiko: 5.0

GET <https://host:port/resto/api/events/sessions>

Параметры запроса

Название	Значение	Описание
from_time	yyyy-MM-ddTHH:mm:ss.SSS	Время с которого запрашиваются данные по кассовым сменам, в формате ISO: yyyy-MM-ddTHH:mm:ss.SSS.
to_time	yyyy-MM-ddTHH:mm:ss.SSS	Время по которое (не включительно) запрашиваются данные по кассовым сменам в формате ISO: yyyy-MM-ddTHH:mm:ss.SSS,

Что в ответе

Информация о кассовых сменах. Время открытия, время закрытия, менеджер, номер смены, номер кассы, опер. день.

Пример вызова

```
https://localhost:8080/resto/api/events/sessions?key=43d272ed-cfb1-64b1-9842-f3078cd69172&from_time=2015-03-28T00:00:00.000&to_time=2015-03-28T00:00:00.000
```



Если работаете по ревизии, не нужно (нельзя) указывать *to_time*

[+] [XSD Дерево событий](#)

[-] [XSD Дерево событий](#)

[Copy Code](#)

XML

```
<!-- Схема описания дерева событий журнала событий -->
<!-- Дерево выдается в качестве линейной структуры -->
```

```

<!-- групп событий. Информация об иерархии групп -->
<!-- внутри древовидной структуры теряется -->
<!-- Кроме этого, теряется информация об обязательности и
необязательности атрибутов событий -->
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="group" type="group"/>
  <xs:element name="type" type="type"/>
  <xs:element name="attribute" type="attribute"/>
  <xs:element name="groupsList" type="groupsList"/>
  <!-- Дерево событий состоит из списка групп -->
  <xs:complexType name="groupsList">
    <xs:sequence>
      <xs:element ref="group" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <!-- Элементы этого списка - группы событий -->
  <xs:complexType name="group">
    <xs:sequence>
      <!-- Идентификатор группы = resto.event.EventGroupMetadata.id -->
      <xs:element name="id" type="xs:string" minOccurs="0"/>
      <!-- Имя группы = resto.event.EventGroupMetadata.name -->
      <xs:element name="name" type="xs:string" minOccurs="0"/>
      <!-- Список событий, входящих в данную группу-->
      <xs:element ref="type" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <!-- Событие Журнала Событий -->
  <xs:complexType name="type">
    <!-- Каждое событие состоит из набора -->
    <xs:sequence>
      <!-- Список Атрибутов события -->
      <xs:element name="attribute" type="attribute" minOccurs="0"
maxOccurs="unbounded"/>
      <!-- Идентификатор события = resto.event.EventTypeMetadata.id -->
      <xs:element name="id" type="xs:string" minOccurs="0"/>
      <!-- Имя события = resto.event.EventTypeMetadata.name -->
      <xs:element name="name" type="xs:string" minOccurs="0"/>
      <!-- Важность события (число) (0 - Низкая, 1 - Средняя, 2 -
Высокая) -->
      <xs:element name="severity" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <!-- Атрибут события -->
  <xs:complexType name="attribute">
    <xs:sequence>
      <!-- Идентификатор обязательного атрибута =
resto.event.EventAttributeMetadata.id -->
      <xs:element name="id" type="xs:string" minOccurs="0"/>
      <!-- Имя обязательного атрибута =
resto.event.EventAttributeMetadata.name -->
      <xs:element name="name" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

```



```
</xs:complexType>
</xs:schema>
```

[+] XSD Список событий

[-] XSD Список событий

Copy Code

XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="event" type="event"/>
  <xs:element name="eventAttribute" type="eventAttribute"/>
  <xs:element name="eventsList" type="eventsList"/>
  <!-- События -->
  <xs:complexType name="eventsList">
    <xs:sequence>
      <!-- Список событий -->
      <xs:element ref="event" minOccurs="0" maxOccurs="unbounded"/>
      <!-- Ревизия списка событий -->
      <xs:element name="revision" type="xs:int" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <!-- Событие -->
  <xs:complexType name="event">
    <xs:sequence>
      <!-- Список атрибутов события -->
      <!-- Атрибуты события могут частично пересекаться с атрибутами
событий, перечисленных в groupsList.group.type.attribute -->
      <!-- Некоторые события имеют обязательные атрибуты - по таким
пересечению обязательно -->
      <xs:element name="attribute" type="eventAttribute" minOccurs="0"
maxOccurs="unbounded"/>
      <!-- Дата и время события -->
      <xs:element name="date" type="xs:dateTime" minOccurs="0"/>
      <!-- id департамента (guid) - для iikoChain -->
      <xs:element name="departmentId" type="xs:string" minOccurs="0"/>
      <!-- Уникальный идентификатор события (guid) -->
      <xs:element name="id" type="xs:string" minOccurs="0"/>
      <!-- Тип события = resto.event.EventTypeMetadata.name =
groupsList.group.type.id -->
      <xs:element name="type" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <!-- Атрибут события -->
  <xs:complexType name="eventAttribute">
    <xs:sequence>
      <!-- Имя атрибута -->
      <xs:element name="name" type="xs:string" minOccurs="0"/>
```

```
<!-- Значение атрибута -->  
<xs:element name="value" type="xs:string" minOccurs="0"/>  
</xs:sequence>  
</xs:complexType>  
</xs:schema>
```