Events of Users-Objects

What is an Event

It is something that happens with an Object. Can be anything, from application launch to file attached. Each object can have MANY Events.

Creating an Event

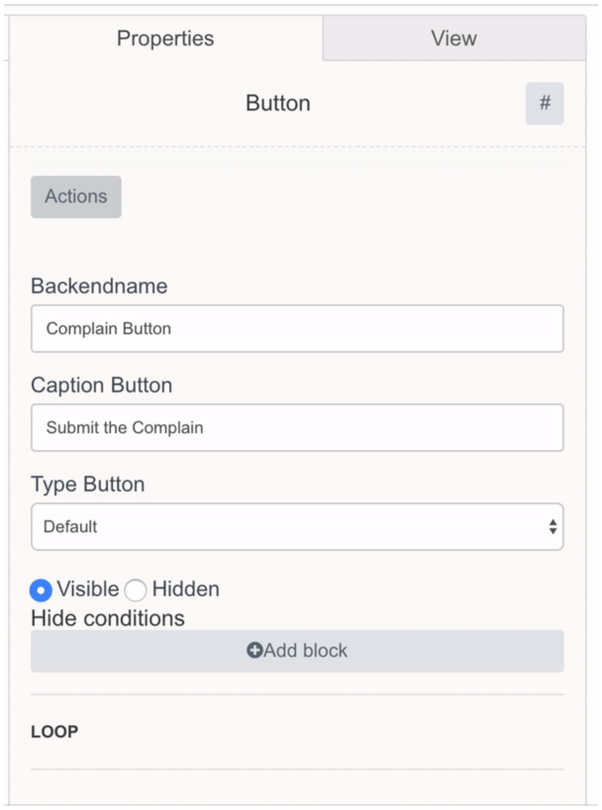
Technically an Event is the record in the Event's Table. It can be created on the following actions:

- Button / Link / Image click
- App Launch
- API request
- Triggers

This new record can contain any data you wish to be stored in it.

How to create a click Event in Custructor

1. Select "Actions" in the Properties of the Single Button / Link / Image
2. Click "Create Event"
3. Fill in the table: "Key" is the name of the field and "Value" is the parameter. The Parameter can be a static string or Hashtag reference.
4. Click "Save"



Properties | View

Button #

Actions

Backendname: Complain Button

Caption Button: Submit the Complain

Type Button: Default

[✓ Visible] Hidden

Hide conditions

Add block

LOOP

Viewing and Managing Events

Events are located in the APP DATA section of APP's menu

? Unknown Attachment

CHANGING THE TABLE

The following options are available:

- Add a custom data field to the table - Add Custom Column (ORANGE MARK)
- Actions with event (GREEN MARK)

? Unknown Attachment

- Manual Filtering (RED MARK)
- Changing

? Unknown Attachment

The following System Fields are available for Events, which you can view in the table by pressing Setup up table (BLUE MARK):

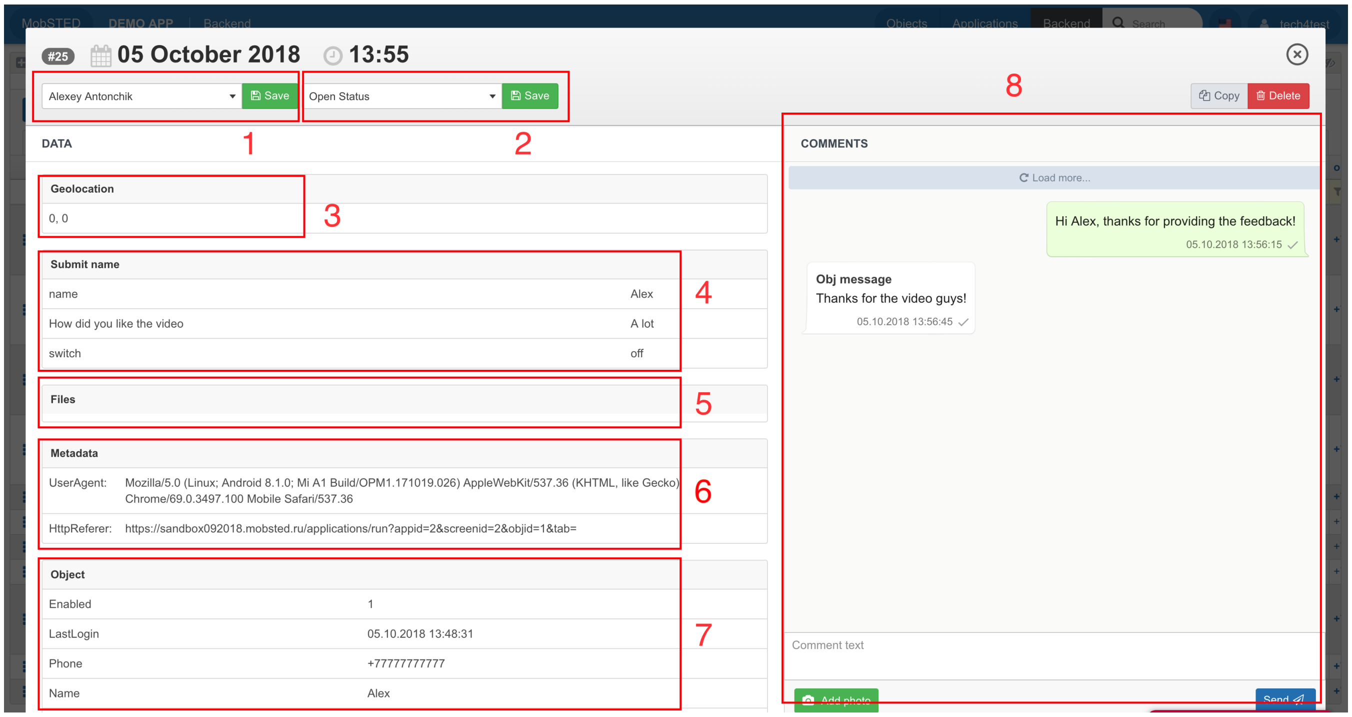
- **ID** - a unique incremental event ID
- **CreateDate** - when it was created
- **UpdateDate** - when it was last updated
- **screens_jsinName**

- **ActionName** - which Action in app had created the event (backendname of a screen's element)
- **Value** - that was passed to the event
- **SrcTemplate** -
- **SrcScreen** - from which app screen was it created
- **SrcEvent** -
- **SrcTrigger** - which trigger had created it
- **SrcChannel** - which mobile channel was accessed to create it
- **payementsystem** - name of pre-integrated payment system if event is a payment
- **paymentstatus** - status of payment returned by PSP
- **fiscalstatus** - status of tax submission for certain markets
- **StatusID** - general status of the event (refer to Statuses article)
- **ResponsibleID** - if a user responsible for this event exists (internal business users)
- **UUID** - unique and "scrambled" ID of a user in alphanumeric format - 16 random symbols 0-9, a-z, A-Z (only accessible by APIs)

READING EVENTS MANUALLY

Click on Event to open up its details.

The pop-up displays data saved to **event** during creation and **app state** snapshot data for that moment.



05 October 2018 13:55

Alexey Antonchik

Open Status

DATA

Geolocation

0, 0

Submit name	
name	Alex
How did you like the video	A lot
switch	off

Files

Metadata

UserAgent: Mozilla/5.0 (Linux; Android 8.1.0; Mi A1 Build/OPM1.171019.026) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/69.0.3497.100 Mobile Safari/537.36

HttpReferer: https://sandbox092018.mobsted.ru/applications/run?appid=2&screenid=2&objid=1&tab=

Object	
Enabled	1
LastLogin	05.10.2018 13:48:31
Phone	+77777777777
Name	Alex

COMMENTS

Hi Alex, thanks for providing the feedback!
05.10.2018 13:56:15

Obj message
Thanks for the video guys!
05.10.2018 13:56:45

Comment text

Event pop-up contains following info:

1. Responsible set for an event (internal users only, used for support purposes)
2. The current Status of the event. The status can be tracked by using filters. To learn more on - Statuses.
3. Location of the of the device when the event was created (if available)
4. The info on the current event. It contains all the information that was passed to the Event in the form of **Key** (left) and **Value** (right) pairs.
5. Files if any file was attached event using the file element.
6. The metadata on the Object like which browser he used to access the app, which version of operating system, etc.
7. The information about the Object that this event applies to. This is used for convenience purposes to avoid the need to navigate back and forth between Backend and the Objects tabs.
8. Chat with a mobile app user about this even (if connected)

WORKING EVENTS USING API CALLS

In order to Create / Update / Delete an Event using API Request you should do the following:

1. Get Access token - https://documenter.getpostman.com/view/4564343/S11LscyP#8cdbe179-7e2d-6218-9273-0df0bc8edfdf
2. Create / Update / Delete the Event - https://documenter.getpostman.com/view/4564343/S11LscyP#cfffad1b-9646-94a2-e74a-7eb03235caf3

USING EVENTS BEST PRACTICES

Events are mostly used when you want to submit some dynamic data or track user actions, not once, but periodically.

Good Use	Not such a good use
You have a "a complain form" to submit a problem. Since one AppUser can submit multiple complains and at different times, it's is good idea to use "Create an event" action for each time, by assigning such Action to a button submitting the form.	You have a "Enter your personal info form" asking the AppUser for the name. Creating an Event makes no sense, as this is entered just once. You are better off just recording this data into Object's column using "Save to Object Column" action.

PERFORM ACTIONS ON EVENTS

Events can be tracked by filters and automatic actions can be performed based on data in Events. Using filters combined with triggers allows to set up the rules to perform actions on the Events. Please refer to Triggers article for more info.

- Deploy Changes
- App users - Objects
- Events of Users-Objects
- Direct users to the app
- Platform's Entities and Relations