

Understanding TIXnGO Ticket Injection

1. Preface

In this page we will describe every possible parameters Organizers can change in order to inject the perfect tickets for their events. Tickets are injected via the TIXnGO API, encapsulating our Blockchain. To call our API and inject new tickets, you will need a valid Organizer API key, if you do not have one yet follow the sign-up steps on the Console (<https://console.tixngo.io/signup/step1> or [How to add a new Organizer \(Premium TIXnGO account\)](#) for a Premium account, this will require actions from the TIXnGO Team).

The injection of Tickets is done through the call POST /organizer/tickets, with a JSON body containing the list of tickets to be injected (example provided in 2).

An injection tool using .csv files is provided via the TIXnGO console (<https://console.tixngo.io/injection>). WARNING: under development, not every parameter is yet available.

2. JSON Example

(*) = Mandatory.

```
{
  "tickets": [
    {
      "activationParameters": {
        "groupId": "string",
        "method": "online",
        "time": "2020-10-05T11:19:24.927Z",
        "instanceId": "string"
      },
      "image": "string",
      "event" (*): {
        "id" (*): "string",
        "address" (*): {
          "site" (*): "string",
          "line1": "string",
          "line2": "string",
          "line3": "string",
          "city" (*): "string",
          "countryCode" (*): "string",
          "region": "string",
          "zip" (*): "string",
          "longitude": 0,
          "latitude": 0
        },
        "name" (*): "string",
        "name2": "string",
        "website": "string",
        "startTime" (*): "string",
        "expirationDate": "string",
        "metadata": [
          {
            "key" (*): "string",
            "value" (*): "string"
          }
        ],
        "group": {
          "id": "string",
          "name": "string",
          "image": "string"
        }
      },
      "proofOfIdentity": {
        "publicKey" (*): "string",
        "signedMessage" (*): "string"
      },
      "purchaseDetails" (*): {
        "price" (*): 0,
        "currency" (*): "string",
        "date" (*): "2020-10-05T11:19:24.927Z",
        "priceCategory" (*): "string",
        "taxationNumber" (*): "string"
      }
    }
  ]
}
```

```

},
"ticketDetails": {
  "main": [
    {
      "key" (*): "string",
      "value" (*): "string"
    }
  ],
  "extra": [
    {
      "key" (*): "string",
      "value" (*): "string"
    }
  ],
  "hidden": [
    {
      "key" (*): "string",
      "value" (*): "string"
    }
  ]
},
"security" (*): {
  "barcode" (*): "string",
  "clearanceLevel": 0
},
"spectatorDetails" (*): {
  "phoneNumber": "string",
  "email" (*): "user@example.com",
  "firstName" (*): "string",
  "lastName" (*): "string",
  "gender": "string",
  "dateOfBirth": "2020-10-05",
  "nationality": "string",
  "passportNumber": "string",
  "idCardNumber": "string",
  "spectatorStatus": "activated",
  "ticketHolder": {
    "identity" (*): {
      "firstName" (*): "string",
      "lastName" (*): "string",
      "gender" (*): "m",
      "face": "string",
      "dateOfBirth": "2020-10-05",
      "nationality": "string",
      "passportNumber": "string",
      "idCardNumber": "string",
      "email": "string",
      "phoneNumber": "string",
      "address": {
        "line1 (*)": "string",
        "line2": "string",
        "line3": "string",
        "city (*)": "string",
        "zip (*)": "string",
        "countryCode (*)": "string"
      },
      "birthCity": "string",
      "birthCountry": "string",
      "residenceCountry": "string"
    },
    "extraInfo1": "string",
    "extraInfo2": "string",
  }
}

```

```

    "reason": string
  },
  "organizerSpectatorPreference": {
    "allowPromoEmail" (*): true,
    "allowPromoThirdPartyEmail" (*): true,
    "allowTracking" (*): true
  },
  "createdDate": "string"
},
"ticketId" (*): "string",
"lang": "string",
"transferRules": {
  "allowResell": true,
  "allowResellAfterActivation": true,
  "allowTransferAfterControl": true,
  "allowTransferAfterActivation": true,
  "allowTransfer": true,
  "allowTransferMainApplicant": true,
  "assignTransferLimit": -1,
  "maxNumberOfTicketPerPhoneInGroupId": 0,
  "maxNumberOfTransferPerPhoneInGroupId": 0,
  "maxNumberOfTransferPerTicketInGroupId": 0,
  "maxResellPriceIncrease": 0,
  "keepOneInGroupId": true,
  "keepOneAtInjectInGroupId": true,
  "groupId": "string"
},
"ticketInjectionDate": "2020-10-05T11:19:24.927Z",
"mainApplicant": false,
}
},
"mobileAppId": "string",
"validate": false
}

```

3. Variable Details

Variable Name	Variable Type	Value length	Variable usage/description
tickets	list		The list of Tickets to be injected. every ticket can hold the entire set of parameters displayed in 2.
activationParameters	object		The parameters defining the access control method and details. An activated ticket display its barcode; before activation the barcode is encrypted.
activationParameters.groupId	string	50 char	The activation group to which the ticket belongs to. Tickets sharing the same activation id will have the same activation method & decryption code.
activationParameters.method	string	"online" /"offline"	The method using to activate the ticket. online : when activationTime is reached (backend clock), the backend activates the tickets and sends push notifications to trigger a GET /spectator/tickets on the app. offline: the mobile application activates the tickets when time is reaches (mobile clock). Online requires the user to be online, Offline does not. More on Ticket activation: 5-Understanding TIXnGO Activation flow
activationParameters.time	string	Date follows RFC 3339 standard	If online is chosen, define the time at which the tickets are activated (decrypted).
activationParameters.instanceId	string	128 char	If offline is chosen, define the bluetooth instance id that the mobile app should listen to to get the decryption key.
image	string	256 char	The uri to the image that will be displayed to the user with its ticket. It must have a width:length ratio of X:Y (to be defined) . (displayed on the "Ticket" view).
event	object		The Event the Ticket gives access to.
event.id	string	256 char	ID representing the event (unique).
event.address	object		The Event's Address
event.address.site	string	200 char	Name of the site the Event will take place.

event.address.line1	string	200 char	Address line 1
event.address.line2	string	200 char	Address line 2
event.address.line3	string	200 char	Address line 3
event.address.city	string	200 char	City where the event takes place
event.address.countryCode	string	200 char	Country code for the Address
event.address.region	string	200 char	Region where the Event takes place
event.address.zip	string	50 char	Zip code where the Event takes place
event.address.longitude	double	double	Longitude where the Event takes place
event.address.latitude	double	double	Latitude where the Event takes place
event.name	string	128 char	Name of the Event
event.name2	string	128 char	Second name of the Event (optional)
event.website	string	128 char	Website URL of the Event.
event.startTime	string	Date follows RFC 3339 standard	Time at which the event starts. Used for app logic such as reminders etc...
event.expirationDate	string	Date follows RFC 3339 standard	Time at which the event expire. Expired event cannot be used anymore.
event.metadata	list	list of key-value metadata	<p>Specific "key-value" fields provided to Organizer for special information about the event. For now we handle the reserved keys below. You can override all those keys if you put them (same name, case sensitive) in the ticket details hidden section of the ticket.</p> <p>Reserved Keys :</p> <ul style="list-style-type: none"> ▪ <code>BARCODE_SIZE</code>: value = Size of the barcode display on the Ticket (16 < x < 32) [mm] ▪ <code>BARCODE_POSITION</code>: value = Position of the barcode on the Ticket (LEFT, CENTER, RIGHT) ▪ <code>prohibited_items_image</code>: value = URL of the prohibited items image ▪ <code>prohibited_items_link</code>: value = URL of the prohibited items link ▪ <code>sponsorImage</code>: value = URL of the sponsors image ▪ <code>sponsorLink</code>: value = URL of the sponsors link ▪ <code>seat_map_image</code>: value = URL of the seat map image ▪ <code>seat_map_link</code>: value = URL of the seat map link
event.group	object		Group the Event belongs to. In TIXnGO multiple events can be grouped together (ex: The same competition could have multiple events).
event.group.id	string	256 char	Id of the Group. Every Event with this Group ID will be part of that Group.
event.group.name	string	128 char	Name of the Group.
event.group.image	string	256 char	The Image of the Group (displayed on the "My Events" view).
proofOfIdentity	object		The informations needed to verify the authenticity of the organizer identity.

proofOfIdentity. publicKey	string	256 char	The webRootUrl of the organizer. A request on _tixngo.publicKey should give us the organizer public key used to verify the signature.
proofOfIdentity. signedMessage	string	256 char	Signature of the message (which is a concatenation of different fields. it is available in the documentation).
purchaseDetails	object		The information related to the initial purchase of the ticket.
purchaseDetails. price	double	double	The Ticket's price.
purchaseDetails. currency	string	20 char	The currency used in the transaction using ISO-4217 standard (example: CHF, VND, USD).
purchaseDetails. date	string	Date follows RFC 3339 standard	The date at which the Ticket was initially bought.
purchaseDetails. priceCategory	string	512 char	Examples of price category are adult, child, student, senior, VIP, etc.
purchaseDetails. taxationNumber	string	512 char	(UUID) taxation number of the Ticket.
ticketDetails	object		Some supplementary details about the Ticket.
ticketDetails. main	list	list of key-value metadata	Free "key-value" fields usually used to define spectator seating information such as: gate:A, row:2, seat:7 (Displayed on the "front" of the Ticket).
ticketDetails. extra	list	list of key-value metadata	Free "key-value" fields usually used to define extra information (Displayed on the "back" of the Ticket). Reserved Keys : file_id : optional field set by the Organizer at injection.
ticketDetails. hidden	list	list of key-value metadata	Free "key-value" fields usually used to add hidden information, ex: used between Organizer and TIXnGO (Not displayed on the Ticket). Reserved Keys : transfer_extra : optional field set by the sender during a ticket transfer in transferOrder.extra Furthermore, you can override the all the event.metadata keys on a per ticket level.
security	object		The Ticket's security information, used by the access control and to enter the event.
security. barcode	string	512 char	Barcode of the Ticket.
security. clearanceLevel	integer	integer	Clearance level of the ticket. If 1 is chosen: the organizer require the spectator to provide its name,surname,gender. If 2 is chosen, then all field of SpectatorIdentity are required except for the passport and idcard number, only one of them is required.
spectatorDetails	object		The information that will be used to link the ticket to a spectator (the ticket's buyer) in our system. We call this spectator the Initial spectator.
spectatorDetails. phoneNumber	string	50 char	Initial spectator's phone number.
spectatorDetails. email	string	256 char	Initial spectator's email.
spectatorDetails. firstName	string	50 char	Initial spectator's first name.
spectatorDetails. lastName	string	100 char	Initial spectator's last name.
spectatorDetails. gender	string	1 char (m/f/u/x)	Initial spectator's gender. m, f, u or x (For Male, Female, Other, Unknown)
spectatorDetails. dateOfBirth	string	Date follows RFC 3339 standard	Initial spectator's birth date.
spectatorDetails. nationality	string	ISO Alpha-2 format	Initial spectator's nationality.

spectatorDetails. passportNumber	string	200 char	Initial spectator's passport number.
spectatorDetails. idCardNumber	string	200 char	Initial spectator's ID card number.
spectatorDetails. spectatorStatus	string	"Activated" / "Not activated"	If the initial spectator is registered in the TIXnGO system.
spectatorDetails. ticketHolder	object		Used to show that this Ticket is held for someone else on the Initial spectator's phone.
spectatorDetails. ticketHolder. identity	object		Ticket Holder's identity. (The name "Ticket Holder" might be confusing, this is the person for whom the ticket is held).
spectatorDetails. ticketHolder.identity. firstName	string	50 char	Ticket Holder's first name.
spectatorDetails. ticketHolder.identity. lastName	string	100 char	Ticket Holder's last name.
spectatorDetails. ticketHolder.identity. gender	string	1 char (m/f/u/x)	Ticket Holder's gender m, f, u or x(For Male, Female, Other, Unknown)
spectatorDetails. ticketHolder.identity. dateOfBirth	string	Date follows RFC 3339 standard	Ticket Holder's birth date.
spectatorDetails. ticketHolder.identity. nationality	string	ISO Alpha-2 format	Ticket Holder's nationality.
spectatorDetails. ticketHolder.identity. passportNumber	string	200 char	Ticket Holder's passport number.
spectatorDetails. ticketHolder.identity. idCardNumber	string	200 char	Ticket Holder's ID card number.
spectatorDetails. ticketHolder.identity. email	string	256 char	Ticket Holder's email.
spectatorDetails. ticketHolder.identity. phoneNumber	string	50 char	Ticket Holder's phone number.
spectatorDetails. ticketHolder.identity. birthCity	string	50 char	Ticket Holder's birth city.
spectatorDetails. ticketHolder.identity. birthCountry	string	ISO Alpha-2 format	Ticket Holder's birth country.
spectatorDetails. ticketHolder.identity. residenceCountry	string	ISO Alpha-2 format	Ticket Holder's residence country.
spectatorDetails. ticketHolder.identity. address	object		Ticket Holder's address.
spectatorDetails. ticketHolder.identity. address. line1	string	200 char	Ticket Holder's address line 1.
spectatorDetails. ticketHolder.identity. address. line2	string	200 char	Ticket Holder's address line 2.
spectatorDetails. ticketHolder.identity. address. line3	string	200 char	Ticket Holder's address line 3.
spectatorDetails. ticketHolder.identity. address. city	string	200 char	Ticket Holder's residence city.

spectatorDetails.ticketHolder.identity.address.countryCode	string	200 char	Ticket Holder's country code.
spectatorDetails.ticketHolder.identity.address.zip	string	50 char	Ticket Holder's zip code.
spectatorDetails.ticketHolder.extralInfo1	string	100 char	Free field of extra info about the Ticket Holder.
spectatorDetails.ticketHolder.extralInfo2	string	100 char	Second free field of extra info about the Ticket Holder.
spectatorDetails.ticketHolder.reason	string	512 char	Reason of why the assignment happen. Can be modified later by the spectator. (Optional)
spectatorDetails.organizerSpectatorPreference	object		Contains all spectator preferences concerning promo emails and tracking.
spectatorDetails.organizerSpectatorPreference.allowPromoEmail	boolean	boolean	Default false. Spectator accept promotional email from TIXnGO.
spectatorDetails.organizerSpectatorPreference.allowPromoThirdPartyEmail	boolean	boolean	Default false. Spectator accept promotional email from Third-Party.
spectatorDetails.organizerSpectatorPreference.allowTracking	boolean	boolean	Default false. Spectator accept to be tracked and data matching.
ticketId	string	256 char	Ticket ID differentiating each ticket. It must be unique (UUID).
lang	string	5 char ISO 6391 format	The default language for the Ticket. Injection communications will be sent in this language. Example: fr, en, de, it, es
transferRules	object		All of the Ticket's transfer rules.
transferRules.groupId	string	256 char	Transfer rules can technically be different for every ticket. Organizer might want to chose to have different transfer rules for different types of ticket (ex: Adult, child, VIP, etc...). This GroupID is the Transfer rules group ID representing a group of transfer rules applied to certain tickets. If Organizers want to apply the same rules to every tickets, by default this groupID is set to the EventID and every ticket will be in this group.
transferRules.maxNumberOfTicketPerPhoneInGroupid	integer	integer	Maximum amount of tickets per phone for this Transfer rules group id.
transferRules.maxResellPriceIncrease	integer	smallint	Maximum profit (price increase in %) of the original ticket price for a resell between spectator.
transferRules.maxNumberOfTransferPerPhoneInGroupid	integer	integer	Total amount of transfers per Phone (-1 feature is disabled).
transferRules.maxNumberOfTransferPerTicketInGroupid	integer	integer	Total amount of transfers per Ticket (-1 feature is disabled).
transferRules.assignTransferLimit	integer	integer	Define upper bound of ticket's transfer level for the "Assign" function
transferRules.allowTransfer	boolean	boolean	Enables or Disables transfer functionality for given match (Send menu item in the ticket view hidden or displayed).
transferRules.allowTransferMainApplicant	boolean	boolean	Enable or Disable transfer functionality for Main Applicant ticket's.
transferRules.allowTransferAfterActivation	boolean	boolean	Enables or Disables transfer functionality for given match for Activated Tickets (Send menu item in the ticket view hidden or displayed).
transferRules.allowTransferAfterControl	boolean	boolean	Enables or Disables transfer functionality for given match for Controlled Tickets (Send menu item in the ticket view hidden or displayed).

transferRules.allowResell	boolean	boolean	Enables or Disables resale functionality for given match for all Tickets (Send menu item in the ticket view hidden or displayed).
transferRules.allowResellAfterActivation	boolean	boolean	Enables or Disables resale functionality for given match for ActivatedTickets (Send menu item in the ticket view hidden or displayed).
transferRules.keepOnInGroupId	boolean	boolean	Force ALL ticket wallet holders to keep at least 1 ticket per given Match.
transferRules.keepOnAtInjectionInGroupId	boolean	boolean	Force ticket wallet holder to whom tickets were originally injected to keep at least 1 ticket per given Match. E.g. all his transfer guests will be able to forward all the tickets they received.
ticketInjectionDate	string	Date follows RFC 3339 standard	The date at which the ticket was injected.
mainApplicant	boolean	boolean	Optional Main Applicant flag (SecuTix)
mobileAppId	string	256 char	Optional Mobile Application App Id to use associated invitation email template (used by organizer using their own app or branded apps).
validate	boolean	boolean	Flag to only validate tickets and not injecting them for real. It does not test interaction with the blockchain.

4. Ticket Injection Through CSV file

The TIXnGO Console allows organizers to inject their tickets via CSV files, please look at the following rules:

1. Please follow the same wording for the header keys, i.e. for the Event name, the column name in csv should be; *event.name*.
2. If you are using Microsoft Excel to create your CSV file, make sure the cells are in *text* format and that they are separated by a semicolon ";".
3. Please use the **same** mobileAppId for all tickets in the CSV.
4. You cannot inject more than 20'000 tickets at a time.

An easy trick to make sure that this is respected is to download the provided template, open Excel, then under the *Data* tab click *From Text/CSV*, select the template you desire make sure Delimiter is *Semicolon* and Data Type Detection is *Do not detect data types*, then click *Load*.