

Spis treści

Common	3
Settings	3
GetSettings	3
Event	4
CreateEvent	4
UpdateEvent	4
AssigningToEvent	5
LeaveEvent	6
DeleteEvent	6
GetEvent	6
GetEvents	7
AddLogo	8
DeleteLogo	8
AddSlide	8
DeleteSlide	9
UpdateAccessCode	9
UpdateSlideShowUrl	9
Media	11
AddPhotoThumbnail	11
AddDSLRPhoto	11
AddPhotoFull	12
GetPhotos	12
GetPublicPhotos	12
Permissions	14
UpdateParticipantByEmail	14
DeleteParticipantByEmail	14
UpdateParticipantByEmail (deprecated)	14
GetParticipants	15
UpdateOwnerByEmail (deprecated)	15
GetOwners	16
Photos	16
CheckPhotosFullResolution	16
User	17
CreateOrUpdateUser	17
UpdateDataCollectionConsent	17
UpdateApplicationRatedStatus	17

GetUser.....	18
Archive.....	19
ZipPhotos.....	19
Support.....	19
CreateRequest.....	19

Common

BaseUrl:

Headers:

x-zumo-auth: "....."

Settings

GetSettings

Url:

{BaseUrl}/settings

Request Type:

GET

Response:

200 – body contains settings of application:

- "ThumbnailSize": 500
- "ThumbnailQuality": 80
- "AppcenterEnabled": true

Event

CreateEvent

Url:

{BaseUrl}/events

Request Type:

POST

Body:

```
{  
    "Name" : "Michal4",  
    "StartDateTime" : "2012-04-23T18:25:43.511Z",  
    "EndDateTime" : "2012-05-23T18:25:43.511Z",  
    "Latitude": "10.0",  
    "Longitude" : "20.0",  
    "IsOpen" : "true"  
}
```

Limitations:

Name – required, max 500 characters

StartDateTime – required

EndDateTime – required, later or equal than *StartDateTime*

Latitude – optional, value between -180 and 180

Longitude – optional, value between -90 and 90

IsOpen – optional, default value *False*

*if *Latitude* and *Longitude* are used then both values must be provided

Response:

202 with body that contains *IdEventu*

UpdateEvent

Url:

{BaseUrl}/event/{eventIdString}

Request Type:

PUT

Body:

```
{  
    "Name" : "Michal4",  
    "StartDateTime" : "2012-04-23T18:25:43.511Z",
```

```
"EndTime" : "2012-05-23T18:25:43.511Z",  
"Latitude": "10.0",  
"Longitude" : "20.0",  
"IsOpen" : "true"  
}
```

Limitations:

EventIdString – required, guid

Name – required, max 500 characters

StartTime – required

EndTime – required, laterod *StartTime*

Latitude – optional, value between -180 and 180

Longitude – optional, value between -90 and 90

IsOpen – optional, default value *False*

*if *Latitude* and *Longitude* are used then both values must be provided

Response:

202 with body that contains *IdEventu*

AssigningToEvent

Url:

{BaseUrl}/events/assign

Request Type:

POST

Body:

```
{  
"ShortCode": "ImcadoEvent"  
}
```

Limitations:

ShortCode – required; cannot be empty

Response:

202 with body that contains *IdEventu*

LeaveEvent

Url:

{BaseUrl}/event/{eventIdString}/leave

Request Type:

POST

Body:

```
{  
}
```

Limitations:

eventIdString – required, guid

Response:

202

DeleteEvent

Url:

{BaseUrl}/event/{eventIdString}/delete

Request Type:

DELETE

Body:

```
{  
}
```

Limitations:

EventIdString – required, guid

Response:

202

GetEvent

Url:

{BaseUrl}/event/{eventIdString}

Request Type:

GET

Response

200 – event details:

```
{  
    "Name" : "Michal4",
```

```

"StartDateTime" : "2012-04-23T18:25:43.511Z",
"EndDateTime" : "2012-05-23T18:25:43.511Z",
"Latitude": "10.0",
"Longitude" : "20.0",
"IsOpen" : "true"
"NumberOfPhotos": 25
"NumberOfParticipants": 40
"AccessCode": "Imcado"
"SlideShowCode": "Imcado"
"LogoLink": "https://imcado...."
"Slides":
    [
        {
            „id”: guid
            „url”: „https://
        }
    ]
}

```

GetEvents

Url:

{BaseUrl}/events

Request Type:

GET

Params

"LastCheckDate" : "2018-02-15T16:54:09.823547Z",

Limitations:

LastCheckDate – optional, when empty all events will be returned

Response

200 – list of events with a date when last time it has been checked

Remarks:

Events that have been modified after the last check date can be shown on the list.

AddLogo

Url:

{BaseUrl}/event/{eventIdString}/logo

Request Type:

POST

Headers:

“FileContentType” : “image/<typ obrazka>”

“Content-Type”: “application/octet-stream”

Body:

Image data stream

Limitations:

eventIdString – required, Guid()

FileContentType – Accepted types: png, tiff, jpg, gif

Response:

202

DeleteLogo

Url:

{BaseUrl}/event/{eventIdString}/logo

Request Type:

DELETE

Body:

{

}

Limitations:

EventIdString – required, guid

Response:

202

AddSlide

Url:

{BaseUrl}/event/{eventIdString}/slides

Request Type:

POST

Headers:

“FileContentType” : “image/<typ obrazka>”

“Content-Type”: “application/octet-stream”

Body:

Image data stream

Limitations:

eventIdString – required, Guid()

FileContentType – Accepted types: png, tiff, jpg, gif

Response:

202

DeleteSlide

Url:

{BaseUrl}/event/{eventIdString}/slide/{slideIdString}

Request Type:

DELETE

Body:

```
{  
  
}
```

Limitations:

EventIdString – required, guid

SlideIdString – required, guid

Response:

202

UpdateAccessCode

Url:

{BaseUrl}/event/{eventIdString}/accessCode

Request Type:

POST

Body:

```
{  
  
    "AccessCode": "imcado"  
  
}
```

Limitations:

eventIdString – required, Guid()

Response:

202

UpdateSlideShowUrl

Url:

{BaseUrl}/event/{eventIdString}/slideShowUrl

Request Type:

POST

Body:

```
{  
    "Url":"imcado"  
}
```

Limitations:

eventIdString – required, Guid()

Response:

202

Media

AddPhotoThumbnail

Url:

{BaseUrl}/event/{eventIdString}/photos

Request Type:

POST

Headers:

```
"PhotoTakenDate" : "2012-04-23T18:25:43.511Z",  
"FileContentType" : "image/<image type>"  
"Content-Type": "application/octet-stream"
```

Body:

Image data stream

Limitations:

eventIdString – required, Guid()

FileContentType – Accepted types: png, tiff, jpg, gif

Response:

202 – photo id, guid

AddDSLRPhoto

Url:

{BaseUrl}/event/{eventIdString}/photos/dslr

Request Type:

POST

Headers:

```
"PhotoTakenDate" : "2012-04-23T18:25:43.511Z",  
"FileContentType" : "image/<image type>"  
"Content-Type": "application/octet-stream"
```

Body:

Image data stream

Limitations:

eventIdString – required, Guid()

FileContentType – Accepted types: png, tiff, jpg, gif

Response:

202

AddPhotoFull

Url:

{BaseUrl}/event/{eventIdString}/photos/{photoIdString}

Request Type:

PUT

Headers:

"FileContentType" : "image/<typ obrazka>"

"Content-Type": "application/octet-stream"

Body:

Image data stream

Limitations:

eventIdString – required, Guid()

photoIdString – required, Guid()

FileContentType – Accepted types: png, tiff, jpg, gif

Response:

202

GetPhotos

Url:

{BaseUrl}/event/{eventIdString}/photos

Request Type:

GET

Params

"LastCheckDate" : "2018-02-15T16:54:09.823547Z",

Limitations:

eventIdString – required, Guid()

LastCheckDate – optional, when empty all photos will be returned

Response

200 – list of photos with a date when last time it has been checked

GetPublicPhotos

Url:

{BaseUrl}/event/{eventIdString}/public/photos

Request Type:

GET

Params

"LastCheckDate" : "2018-02-15T16:54:09.823547Z",

Limitations:

eventIdString – required, Guid()

LastCheckDate – optional, when empty all photos will be returned

Response

200 – list of photos with a date when last time it has been checked

Permissions

UpdateParticipantByEmail

Url:

{BaseUrl}/event/{eventIdString}/participant/{participantEmail}

Request Type:

POST, PUT

Body:

```
{  
    "IsOwner" : false,  
}
```

Limitations:

EventIdString – required, guid

ParticipantEmail – email to the person who should have access to an event as a participant; it does not need to exist in db. In this case, it will become a new login.

IsOwner – true / false – is owner

Response:

202

DeleteParticipantByEmail

Url:

{BaseUrl}/event/{eventIdString}/participant/{participantEmail}

Request Type:

DELETE

Body:

```
{  
}
```

Limitations:

EventIdString – required, guid

Response:

202

UpdateParticipantByEmail (deprecated)

Url:

{BaseUrl}/event/{eventIdString}/participants/{participantEmail}

Request Type:

POST, PUT

Body:

```
{  
    "IsParticipant" : false,  
}
```

Limitations:

EventIdString – required, guid

ParticipantEmail – email to the person who should have access to an event as a participant; it does not need to exist in db. In this case, it will become a new login.

IsParticipant – true / false – is participant

Response:

202

GetParticipants

Url:

{BaseUrl}/event/{eventIdString}/participants

Request Type:

GET

Limitations:

EventIdString – required, guid

Response:

200 – list of participants with a date when last time it has been checked

UpdateOwnerByEmail (deprecated)

Url:

{BaseUrl}/event/{eventIdString}/owners/{ownerEmail}

Request Type:

POST, PUT

Body:

```
{  
    "IsOwner" : false,  
}
```

Limitations:

EventIdString – required, guid

OwnerEmail – email to the person who should have access to an event as an owner; it does not need to exist in db. In this case, it will become a new login.

IsOwner – true / false – is owner

Response:

202

GetOwners

Url:

{BaseUrl}/event/{eventIdString}/owners

Request Type:

GET

Limitations:

EventIdString – required, Guid()

Response:

200 – list of owners with a date when last time it has been checked

Photos

CheckPhotosFullResolution

Url:

{BaseUrl}/photos/checkphotosfullresolution

Request Type:

POST

Body:

```
{  
    "Ids": ["28B181D1-FC4E-4134-97F0-4D63B0C9BBF8","260C2715-EA41-4544-BE92-  
0B46F10D841B"]  
}
```

Limitations:

Ids – required, list of photos guids that should be checked

Response

200 – list of photos with information if full resolution photo has been submitted with information about the date when it has been checked.

User

CreateOrUpdateUser

Url:

{BaseUrl}/users

Request Type:

POST, PUT

Body:

```
{  
  
  "Name" : "Michal3",  
  
  "Email" : "aa@aa.pl",  
  
  "ProfilePictureLink" : "http://.....",  
  
  "Language" : "pl",  
  
}
```

Limitations:

Name – required, max 500 characters

Email – required, max 500 characters, valid email

ProfilePictureLink – max 500 characters, link to profile picture

Response:

202

UpdateDataCollectionConsent

Url:

{BaseUrl}/user/dataCollectionConsent

Request Type:

POST

Body:

```
{  
  
  "Consent" : true  
  
}
```

Limitations:

Consent – required, bool

Response:

202

UpdateApplicationRatedStatus

Url:

{BaseUrl}/user/applictionRated

Request Type:

POST

Body:

```
{  
    "ApplicationRated" : true  
}
```

Limitations:

ApplicationRated – required, bool

Response:

202

GetUser

Url:

{BaseUrl}/users/current

Request Type:

GET

Body:

```
{  
  
}
```

Limitations:

Response:

200 – with user data:

```
{  
    "Name" : "Michal"  
    "Email" : "mail@imcado.app"  
    "ProfilePictureLink" : "https://imcado....."  
    "DataCollectionConsent" : true  
    "ApplicationRated" : true  
    "SubscriptionName" : "Basic"  
    "FeaturesFlag" : 32  
}
```

Archive

ZipPhotos

Url:

{BaseUrl}/event/{eventIdString}/photos/archive

Request Type:

POST

Body:

Limitations:

eventIdString – required, Guid()

Response:

202

Support

CreateRequest

Url:

{BaseUrl}/support/

Request Type:

POST

Body:

```
{  
    "UserEmail": "mail@imcado.pl"  
    "Title": "title",  
    "Phone": "1234567",  
    "EventIdString": "24.....",  
    "EventName": "My Event",  
    "Message": "message"  
}
```

Limitations:

userEmail – required, email

title – required

eventIdString – required, Guid()

eventName – required

message – required

Response:

202