

Lemonn Backend

Docker

Github Registry

```
$ cat ./token.txt | docker login https://docker.pkg.github.com -u USERNAME
--password-stdin
$ docker tag IMAGE_ID docker.pkg.github.com/gsnetitalia/Lemonn-Backend/app:latest
```

Socket

Room

Room Join

The user requests to access a chat room through an event on the topic **room-join** the event will take place following structure:

- Input Path **room-join**

Accepts:

```
{
  "token": "<JWT_TOKEN>",
  "conferenceId": "<MONGO_OBJECT_ID>",
  "roomId": "<ROOM_NAME_MD5>"
}
```

- Workflow

Adds participant on room.joinedParticipant array. Saves conference. Responds.

- Output Path **room-joined**

Retruns:

```
{
  "timestamp": "timestamp",
  "roomName": "roomName",
  "participant": "email",
  "conferenceId": "<MONGO_OBJECT_ID>",
  "roomId": "<ROOM_NAME_MD5>",
  "agoraKey": "<String>"
}
```

Send Message

The user asks to post a chat message via an event on the topic `message-sent` the event will have the following structure:

- Input Path `message-sent`

Accepts:

```
{
  "token": "<JWT_TOKEN>",
  "conferenceId": "<MONGO_OBJECT_ID>",
  "roomId": "<ROOM_NAME_MD5>",
  "message": "<MESSAGE>",
}
```

- Workflow

Adds message on room.messages array. Saves conference. Responds.

- Output Path `message-received`

Retruns:

```
{
  "timestamp": "timestamp",
  "roomName": "roomName",
  "participant": "email",
  "conferenceId": "<MONGO_OBJECT_ID>",
  "roomId": "<ROOM_NAME_MD5>",
  "message": "<MESSAGE>",
}
```

Webinar

Intervention Request

The user requests to intervene by publishing an event on the topic `intervention-request` the event will take place following structure

```
{
  "token": "<JWT_TOKEN>",
  "conferenceId": "<MONGO_OBJECT_ID>",
  "roomId": "<ROOM_NAME_MD5>"
}
```

The checks will be carried out on the backend and a subsequent event will be issued on the topic: **intervention-requested** with the following structure:

```
{
  "time": "timestamp",
  "participant": "<PARTICIPANT_EMAIL>",
  "room": "<ROOM_NAME_UNHASHED>",
}
```

Intervention Approve

The webinar host user can approve the request to speak by posting an event on the topic **intervention-approve** the event will have the following structure

```
{
  "token": "<JWT_TOKEN>",
  "conferenceId": "<MONGO_OBJECT_ID>",
  "roomId": "<ROOM_NAME_MD5>",
  "participant": "<PARTICIPANT_EMAIL>"
}
```

The checks will be carried out on the backend and a subsequent event will be issued on the topic: **intervention-approved** with the following structure:

```
{
  "time": "timestamp",
  "participant": "<PARTICIPANT_EMAIL>",
  "room": "<ROOM_NAME_UNHASHED>",
}
```

Room

```
const room = {
  hash: md5(r.roomName),
  name: r.roomName,
  public: r.public ? true : false,
  description: r.roomDescription,
  type: r.roomType,
  joinedParticipants: [],
  startsAt: eventStartsAt,
  endsAt: eventStartsAt, // TODO Definire da dove prendere il valore.
  messages: [],
  //TODO Definire da dove prendere il valore.
  layoutOptions: {
    headerBackground: '',
    chatBackground: '',
    bodyBackground: ''
  }
}
```