

POST /api/register

Creates a user on the system, with the data field as {"friends": [1, 2]}. Note: the example user will always exist, any other registered users might be erased from time to time, as the database might need to be reset.

Input

POST <http://n42k.ddns.net:8899/api/register>

Content-Type: application/json

```
{
  "username": "manel",
  "password": "voar2",
  "firstName": "Manuel",
  "lastName": "Oliveira",
  "data": "{\"friends\": [1,2]}"
}
```

Output

Success:

```
201
{
  "success": {
    "id": 2
  }
}
```

Failure:

```
500
{
  "failed": "Username already exists!"
}
```

POST /api/login

Input

```
POST http://n42k.ddns.net:8899/api/login
Content-Type: application/json
{
  "username": "manel",
  "password": "gaivota2"
}
```

Output

Success:

```
201
{
  "success": {
    "id": 2,
    "username": "manel",
    "firstName": "Manuel",
    "lastName": "Oliveira",
    "data": "{\"friends\": [1, 2]}",
    "token":
"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZCI6MiwiYWVhbnRpIjoibmVudC50dG4LCJleH
AiOiE1MjYwNTAwMjN9.LsRPUvhghEF2UqPbxaiysbmopKMvIM6bw4L9SRppSus"
  }
}
```

Failure:

```
500
{
  "failed": "Invalid username/password combination!"
}
```

GET /api/user/<id>

Gets the data field of an user based on the given auth token.

Input

```
GET http://n42k.ddns.net:8899/api/user/1
auth:eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZCI6MiwianRpIjoianNExNTM3MjctZDY2OS00ZmI3LWE1YjItODljZGVlM2NjZTJiIiwiaWF0IjoxNTI2MDM5ODg4LCJleHAiOjE1MjYwNTAwMjN9.LsRPUvhghEF2UqPbxaiysbmopKMvIM6bw4L9SRppSus
```

Output

Success:

```
201
{
  "success": {
    "id": 1,
    "username": "manel",
    "firstName": "Manuel",
    "lastName": "Oliveira",
    "data": "{\"friends\": [1,2]}"
  }
}
```

Failure:

```
500
{
  "failed": "Invalid token!"
}
```

POST /api/data

Updates the data field of the user based on the given auth token.

Input

```
POST http://n42k.ddns.net:8899/api/data
Content-Type:application/json
auth:eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZCI6MiwianRpIjoianNExNTM3MjctZDY2OS00ZmI3LWE1YjItODljZGV1M2NjZTJiIiwiaWF0IjoxNTI2MDM5ODg4LCJleHAiOiE1MjYwNTAwMjN9.LsRPUvhghEF2UqPbxaiysbmopKMvIM6bw4L9SRppSus
{
  "data": "{\"friends\": [1,2,3]}"
}
```

Output

Success:

```
201
{
  "success": "Successfully updated data!"
}
```

Failure:

```
501
{
  "failed": "Invalid token!"
}
```

POST /api/event/beacon

Creates a beacon event, notifying the server that the user given by the auth token was present in the room *roomId* at the sent time. Note: we could have simply used the time the server receives the request, but doing it this way allows handling cases where we wish to send old events (e.g. the user has no internet after logging in, but we still wish to send his generated events when possible).

Available rooms:

```
"B002", "B003", "B217", "B001", "#XP-PX", "B229", "Auditorium",  
"XPSpaces", "B336", "B232C", "B110", "B113", "B105", "B111"
```

Input

```
POST http://n42k.ddns.net:8899/api/event/beacon  
auth:eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZCI6MiwiYWVhbmopKMvIM6bW4L9SRppSus  
M3MjctZDY2OS00ZmI3LWE1YjItODljZGVlM2NjZTJiIiwiaWF0IjoxNTI2MDM5ODg4LC  
JleHAiOiE1MjYwNTAwMjN9.LsRPUvhghEF2UqPbxaiysbmopKMvIM6bW4L9SRppSus  
{  
  "roomId": "B002",  
  "sessionId": null,  
  "time": 1527160920000  
}
```

Output

Success:

```
201  
{  
  "success": "Event handled!"  
}
```

Failure:

```
501  
{  
  "failed": "Invalid token!"  
}
```

GET /api/eventsInput

GET <http://n42k.ddns.net:8899/api/events>

Output

Success:

```
201
{"success": [
  {
    "id": 1,
    "EventTypeId": 1,
    "properties": {
      "time": "31432",
      "sessionId": "5",
      "roomId": null
    }
  },
  {
    "id": 2,
    "EventTypeId": 1,
    "properties": {
      "time": "31432",
      "sessionId": "5",
      "roomId": null
    }
  },
  {
    "id": 3,
    "EventTypeId": 1,
    "properties": {
      "time": "31432",
      "sessionId": "5",
      "roomId": null
    }
  }
]}
```

Failure:

```
500
{
  "failed": "Error!"
}
```