Contents

1 About

Documentation for the Backend API and architecture of the Client.

2 Device Tasks

2.1 List Devices

2.1.1 Client request

```
// 1st argument, optional, options
{
  queue: true || false
}
```

2.1.2 Backend request

```
{
  timestamp: 1706709130813,
}
```

2.1.3 Backend response

```
  {
  timestamp: 1706709130813,
  result: 'OK',
  devices: [
    {
      deviceType: 'SCOREBOARD_SCREEN',
      roomType: 'SCOREBOARD1',
      deviceId: 'scor1',
      macAddress: null,
      ipAddress: null,
      bootedTimestamp: 1702243701606
    },
    {
      deviceType: 'SCOREBOARD_SCREEN',
      roomType: 'SCOREBOARD2',
      deviceId: 'scor2',
      macAddress: null,
      ipAddress: null,
      bootedTimestamp: 1702243701625
    },
    {
      deviceType: 'REGISTRATION_SCREEN',
      roomType: 'ADMINISTRATION1',
      deviceId: '001',
      macAddress: null,
      ipAddress: null,
      bootedTimestamp: 1706707719741
    },
    {
      deviceType: 'RPI_READER',
      roomType: 'ADMINISTRATION1',
      deviceId: 'ADMINISTRATION1Reader',
      macAddress: null,
      ipAddress: null,
      bootedTimestamp: 1705889333198
    }
  ]
}
```

2.1.4 Client response

```
{
  ok: true,
  devices: [
    {
      id: 'scor1',
      type: 'SCOREBOARD_SCREEN',
      room: 'SCOREBOARD1',
      view: null
    },
    {
      id: 'scor2',
      type: 'SCOREBOARD_SCREEN',
      room: 'SCOREBOARD2',
      view: null
    },
    {
      id: '001',
      type: 'REGISTRATION_SCREEN',
      room: 'ADMINISTRATION1',
      view: null
    },
    {
      id: 'ADMINISTRATION1Reader',
      type: 'RPI_READER',
      room: 'ADMINISTRATION1',
      view: null
    }
  ]
}
```

2.2 List Scoreboard Devices

2.2.1 Client request

```
// 1st argument, optional, options
{
  queue: true | false
}
```

2.2.2 Backend request

```
{
  timestamp: 1706711522546,
}
```

2.2.3 Backend response

```
{
  timestamp: 1706711522546,
  result: 'OK',
  scoreboardDevices: [
    {
      deviceId: 'scor1',
      deviceType: 'SCOREBOARD_SCREEN',
      roomType: 'SCOREBOARD1',
      status: 'ROTATING'
    },
    {
      deviceId: 'scor2',
      deviceType: 'SCOREBOARD_SCREEN',
      roomType: 'SCOREBOARD2',
      status: 'MONTHLY'
    }
  ]
}
```

2.2.4 Client response

```
{
  ok: true,
  scoreboardDevices: [
    {
      id: 'scor1',
      type: 'SCOREBOARD_SCREEN',
      room: 'SCOREBOARD1',
      view: 'ROTATING'
    },
    {
      id: 'scor2',
      type: 'SCOREBOARD_SCREEN',
      room: 'SCOREBOARD2',
      view: 'MONTHLY'
    }
  ]
}
```

2.3 List Scoreboard Device Views

2.3.1 Client request

```
// 1st argument, optional, options
{
  queue: true | false
}
```

2.3.2 Backend request

```
{
  timestamp: 1706712075044,
}
```

2.3.3 Backend response

```
{
  timestamp: 1706712075044,
  result: 'OK',
  scoreboardStatuses: [
    'ROTATING',
    'ALL_TIME',
    'MONTHLY',
    'WEEKLY',
    'DAILY',
    'ELEMENTS',
    'ROOMS'
  ]
}
```

2.3.4 Client response

```
{
  ok: true,
  scoreboardViews: [
    'ROTATING',
    'ALL_TIME',
    'MONTHLY',
    'WEEKLY',
    'DAILY',
    'ELEMENTS',
    'ROOMS'
  ]
}
```

2.4 Update Scoreboard Device View

2.4.1 Client request

```
// 1st argument, device
{
  id: 'scor1',
  type: 'SCOREBOARD_SCREEN',
  room: 'SCOREBOARD1',
  view: 'ROTATING'
}

// 2nd argument, new View
```

```
view: "string",

// 3rd argument, options, optional
{
  queue: true || false
}
```

2.4.2 Backend request

```
{
  timestamp : 1707072209571,
  deviceId : "scor1",
  status : "WEEKLY"
}
```

2.4.3 Backend response

```
{
  timestamp : 1707072209641,
  result : "OK"
}
```

2.4.4 Client response

```
{
  ok: true,
  device: {
    id: 'scor1',
    type: 'SCOREBOARD_SCREEN',
    room: 'SCOREBOARD1',
    view: 'new view'
  }
}
```

2.5 Boot Device

2.5.1 Client request

```
// 1st argument, optional, device
{
  id: "",
}
// 2nd argument, optional, options
{
  queue: true | false
}
```

2.5.2 Backend request

```
// Boot the device identified by deviceId
{
  timestamp: 1706724066778,
  devicesAction: "WAKE_UP",
  deviceId: "someDevice"
}

// Boot all devices
{
  timestamp: 1706724066778,
  devicesAction: "WAKEUP_ALL",
  deviceId: "",
}
```

2.5.3 Backend response

```
{
  timestamp: 1706724066778,
  result: 'OK',
  message: 'action executed'
}
```

2.5.4 Client response

```
{
  device: null || {
    id: "",
  }
}
```

2.6 Shutdown Device

2.6.1 Client request

```
// 1st argument, optional, device
{
  id: "",
}
// 2nd argument, optional, options
{
  queue: true | false
}
```

2.6.2 Backend request

```
// Shutdown the device identified by deviceId
{
  timestamp: 1706724066778,
  devicesAction: "SHUTDOWN",
  deviceId: "someDevice"
}

// Shutdown all devices
{
  timestamp: 1706724066778,
  devicesAction: "SHUTDOWN_ALL",
  deviceId: "",
}
```

2.6.3 Backend response

```
{
  timestamp: 1706726298103,
  result: 'OK',
  message: 'action executed'
}
```

2.6.4 Client response

```
{
  ok: true,
  device: null || {
    id: ""
  }
}
```

2.7 Restart Device

2.7.1 Client request

```
// 1st argument, optional, device
{
  id: "",
}
// 2nd argument, optional, options
{
  queue: true | false
}
```

2.7.2 Backend request

```
// Shutdown the device identified by deviceId
{
  timestamp: 1706724066778,
  devicesAction: "RESTART",
  deviceId: "someDevice"
}

// Shutdown all devices
{
  timestamp: 1706724066778,
  devicesAction: "RESTART_ALL",
  deviceId: "",
}
```

2.7.3 Backend response

```
{ timestamp: 1706726929389,
  result: 'OK',
  message: 'action executed'
}
```

2.7.4 Client response

```
{
  ok: true,
  device: null || {
    id: ""
  }
}
```

3 Scoreboard Tasks

3.1 List Scoreboard

3.1.1 Client request

```
// 1st argument, optional, options
{
  queue: true | false
}
```

3.1.2 Backend request

```
{
  timestamp: 1706716622912,
}
```

3.1.3 Backend response

```
{
  timestamp: 1706716622912,
  result: 'OK',
  roomElementAssociations: {
    JOKER: 'AIR',
    BUBBLEBOBBLE: 'WATER',
    SUCKERPUNCH: 'FIRE',
    GRANDPIANO: 'AIR',
    JUSTDOIT: 'FIRE',
    REFLECTIONS: 'AIR',
    SPECTRUMDICE: 'AIR',
    HIGHLIGHTBARS: 'AIR',
    LASERDANCE: 'WATER',
    FUNINTHEBARN: 'FIRE',
    SPACEJAM: 'WATER',
    ALLEYOOPS: 'WATER',
    GOAL: 'WATER',
    LETTERFLOOR: 'AIR'
  }
  live: [
    {
      teamName: "",
      numberOfPlayers: "",
      timeUsed: "",
      played: "",
      won: "",
      lost: "",
      totalPoints: ""
    }],
  teamAllTime: [],
  teamMonthly: [],
  teamWeekly: [],
  teamDaily: [],

  perRoom: {
    JUSTDOIT: [
      {
        teamName: 'team6',
        totalPoints: 298,
        numberOfPlayers: 2,
        created: 1702243702887
      },
    ],
    SUCKERPUNCH: [
      {
        teamName: 'team13',
        totalPoints: 297,
        numberOfPlayers: 2,
        created: 1702243704124
      },
    ],
    LASERDANCE: [
      {
```

```
    teamName: 'team5',
    totalPoints: 293,
    numberOfPlayers: 2,
    created: 1702243702676
  },
],
SPECTRUMDICE: [
  {
    teamName: 'team18',
    totalPoints: 288,
    numberOfPlayers: 2,
    created: 1702243704904
  },
],
FUNINTHEBARN: [
  {
    teamName: 'team2',
    totalPoints: 284,
    numberOfPlayers: 2,
    created: 1702243702245
  },
],
SPACEJAM: [
  {
    teamName: 'team7',
    totalPoints: 290,
    numberOfPlayers: 2,
    created: 1702243703043
  },
],
LETTERFLOOR: [
  {
    teamName: 'team10',
    totalPoints: 265,
    numberOfPlayers: 2,
    created: 1702243703549
  },
],
ALLEYOOPS: [
  {
      teamName: 'team16',
      totalPoints: 297,
      numberOfPlayers: 2,
      created: 1702243704522
    },
  ],
  GRANDPIANO: [
    {
      teamName: 'team4',
      totalPoints: 291,
      numberOfPlayers: 2,
      created: 1702243702512
    },
  ],
  BUBBLEBOBBLE: [
    {
      teamName: 'team2',
      totalPoints: 285,
      numberOfPlayers: 2,
      created: 1702243702213
    },
  ],
  JOKER: [
    {
      teamName: 'team6',
      totalPoints: 283,
      numberOfPlayers: 2,
      created: 1702243702860
    },
  ],
  HIGHLIGHTBARS: [
    {
      teamName: 'team10',
      totalPoints: 298,
      numberOfPlayers: 2,
      created: 1702243703579
    },
  ]
},
perElement: {
    FIRE: [
      {
        teamName: 'team6',
        totalPoints: 298,
        numberOfPlayers: 2,
        created: 1702243702887
      },
    ],
    AIR: [
      {
        teamName: 'team10',
        totalPoints: 298,
        numberOfPlayers: 2,
        created: 1702243703579
      },
    ],
    WATER: [
      {
        teamName: 'team16',
        totalPoints: 297,
        numberOfPlayers: 2,
        created: 1702243704522
      },
    ]
  },
}
```

3.1.4 Client response

```
{
  ok: true,
  scoreboard: {
    roomElementAssociations: ctx.raw.roomElementAssociations,
    live: ctx.raw.live,
    teamAllTime: ctx.raw.teamAllTime,
    teamMonthly: ctx.raw.teamMonthly,
    teamWeekly: ctx.raw.teamWeekly,
    teamDaily: ctx.raw.teamDaily,
    perRoom: ctx.raw.perRoom,
    perElement: ctx.raw.perElement,
  }
}
```

4 Player Tasks

4.1 List Registered Players

4.1.1 Client request

```
// arg #1, optional, options
{
  queue: true | false,
}
```

4.1.2 Backend request

```
{
  timestamp: 1706642934817,
}
```

4.1.3 Backend response

```
{
  timestamp: 1706642934817,
  result: 'OK',
  players: [
    {
      username: 'Merry_2mpmnxcgv1s',
      name: 'Merry',
      surname: 'compassionate',
      email: 'Merry@gmail.com',
      wristbandMerged: false,
      wristband: null
    },
    {
      username: 'Wormtongue_klagnkjxqla',
      name: 'Wormtongue',
      surname: 'jovial',
      email: 'Wormtongue@gmail.com',
      wristbandMerged: false,
      wristband: { wristbandNumber: 230, wristbandColor: 3, active: true }
    },
    {
      username: '6t3o5ds227u',
      name: null,
      surname: null,
      email: null,
      wristbandMerged: false,
      wristband: null
    },
    {
      username: 'Elrond_6ofeexn83ma',
      name: 'Elrond',
      surname: 'vigilant',
      email: 'Elrond@gmail.com',
      wristbandMerged: true,
      wristband: { wristbandNumber: 231, wristbandColor: 4, active: true }
    },
    {
      username: 'ppthree',
      name: 'yolothree',
      surname: 'ggthree',
      email: 'ggthree@gmail.com',
      wristbandMerged: false,
      wristband: null
    },
  ]
}
```

4.1.4 Client response

```
{
  ok: true,
  players: [
    {
      username: 'Merry_2mpmnxcgv1s',
      name: 'Merry',
      surname: 'compassionate',
      email: 'Merry@gmail.com',
      state: 'registered',
      wristband: { id: null, color: '', colorCode: null, state: 'unpaired' }
    },
    {
      username: 'Wormtongue_klagnkjxqla',
      name: 'Wormtongue',
      surname: 'jovial',
      email: 'Wormtongue@gmail.com',
      state: 'registered',
      wristband: { id: 230, color: 'green', colorCode: 3, state: 'paired' }
    },
    {
      username: '6t3o5ds227u',
      name: '',
      surname: '',
      email: '',
      state: 'registered',
      wristband: { id: null, color: '', colorCode: null, state: 'unpaired' }
    },
    {
      username: 'Elrond_6ofeexn83ma',
      name: 'Elrond',
      surname: 'vigilant',
      email: 'Elrond@gmail.com',
      state: 'inTeam',
      wristband: { id: 231, color: 'yellow', colorCode: 4, state: 'paired' }
    },
    {
      username: 'ppthree',
      name: 'yolothree',
      surname: 'ggthree',
      email: 'ggthree@gmail.com',
      state: 'registered',
      wristband: { id: null, color: '', colorCode: null, state: 'unpaired' }
    }
  ]
}
```

4.2 List Players with a Wristband

4.2.1 Client request

```
// 1st argument, optional, options
{
  queue: true || false
}
```

4.2.2 Backend request

```
{
  timestamp: 1706649848057,
}
```

4.2.3 Backend response

```
const response = {
  timestamp: 1706649848057,
  result: 'OK',
  players: [
    {
      username: 'Gilgalad_wsai1ooow3',
      name: 'Gilgalad',
      surname: 'sweet',
      email: 'Gilgalad@gmail.com',
      wristbandMerged: false,
      wristband: { wristbandNumber: 232, wristbandColor: 4, active: true }
    },
    {
      username: 'Gandalf_deil7sv8j4c',
      name: 'Gandalf',
      surname: 'busy',
      email: 'Gandalf@gmail.com',
      wristbandMerged: false,
      wristband: { wristbandNumber: 233, wristbandColor: 4, active: true }
    },
    {
      username: 'Galadriel_12k3dw52kkhi',
      name: 'Galadriel',
      surname: 'jovial',
      email: 'Galadriel@gmail.com',
      wristbandMerged: false,
      wristband: { wristbandNumber: 235, wristbandColor: 5, active: true }
    }
  ]
}
```

4.2.4 Client response

```
{
  ok: true,
  players: [
    {
      username: 'Gilgalad_wsai1ooow3',
      name: 'Gilgalad',
      surname: 'sweet',
      email: 'Gilgalad@gmail.com',
      state: 'registered',
      wristband: { id: 232, color: 'yellow', colorCode: 4, state: 'paired' }
    },
    {
      username: 'Gandalf_deil7sv8j4c',
      name: 'Gandalf',
      surname: 'busy',
      email: 'Gandalf@gmail.com',
      state: 'registered',
      wristband: { id: 233, color: 'yellow', colorCode: 4, state: 'paired' }
    },
    {
      username: 'Galadriel_12k3dw52kkhi',
      name: 'Galadriel',
      surname: 'jovial',
      email: 'Galadriel@gmail.com',
      state: 'registered',
      wristband: { id: 235, color: 'blue', colorCode: 5, state: 'paired' }
    }
  ]
}
```

4.3 Search Player

4.3.1 Client request

```
// 1st argument
searchTerm = "string"

// 2nd argument, options, optional
{
  queue: false || true,
}
```

4.3.2 Backend request

```
{
  timestamp : 1707068032950,
  searchTerm : "l"
}
```

4.3.3 Backend response

```
{
  timestamp: 1707067665549,
  result: 'OK',
  players: [
    {
      username: 'jgtcqvlxs6',
      name: 'Tuor',
      surname: 'vigorous',
      email: 'jgtcqvlxs6@gmail.com',
      wristbandMerged: false,
      wristband: null
    },
    {
      username: 'TG96',
      name: null,
      surname: null,
      email: 'TG96@maze.com',
      wristbandMerged: false,
      wristband: null
    },
    {
      username: ’li’,
      name: ’Melian’,
      surname: ’epic’,
      email: ’ki3fc4jx7jp@gmail.com’,
      wristbandMerged: false,
      wristband: { wristbandNumber: 329, wristbandColor: null, active: true }
    },
    {
      username: ’lo’,
      name: ’Idril’,
      surname: ’brave’,
      email: ’nsevvxw4ca6@gmail.com’,
      wristbandMerged: false,
      wristband: { wristbandNumber: 111, wristbandColor: 2, active: true }
    }

  ]
}
```

4.3.4 Client response

```
{
  ok: true,
  players: [
    {
      username: ’pp’,
      name: ’pp’,
      surname: ’pp’,
      email: ’pp@gmail.com’,
      state: ’registered’,
      wristband: { id: null, color: null, colorCode: null, state: ’unpaired’ }
    }
  ]
}
```

4.4 Register Player

4.4.1 Client request

```
// 1st argument, required, PlayerCommander
```

```
{
  username: 'test',
  name: 'test',
  surname: 'test',
  email: 'testt@gmail.com',
},
// 2nd argument, required password
password: "testpass"
// 3rd argument, optional, options
{
  queue: true || false
}
```

4.4.2 Backend request

```
{
  timestamp: 1706724066778,
  username: "test",
  surname: "test",
  name: "test",
  email: "test@gmail.com",
  password: "testpass",
}
```

4.4.3 Backend response

```
{
  timestamp : 1706874481773,
  result : "OK",
  player : {
    name : "test",
    surname : "test",
    username : "test",
    email : "test@gmail.com",
    wristbandColor : 0,
  }
}
```

4.4.4 Client response

```
{
  ok: true,
  player: {
    username: 'n7rgqxbr0vn',
    name: 'Saruman',
    surname: 'serene',
    email: 'n7rgqxbr0vn@gmail.com',
    state: 'registered',
    wristband: { id: null, color: null, colorCode: null, state: 'unpaired' }
  }
}
```

5 Wristband Tasks

5.1 Register Wristband

5.1.1 Client request

```
// 1st argument, required, Player
{
  username: "test",
  name: 'test',
  surname: 'test',
  email: 'testt@gmail.com',
}
// 2nd argument, required, Wristband
{
  id: 3,
  color: 'green',
  colorCode: 3
}
// 3rd argument, optional, options
{
  queue: true || false
}
```

5.1.2 Backend request

```
{
  timestamp : 1706957679789,
  username : "diwgp3nrrtf",
  wristbandNumber : 234
}
```

5.1.3 Backend response

```
{
  timestamp : 1706957679848,
  result : "OK",
  message : "successfully registerWristbandToPlayer"
}
```

5.1.4 Client response

```
{
  ok: true,
  player: {
    username: "test",
    name: 'test',
    surname: 'test',
    email: 'testt@gmail.com',
    wristband: {
      id: 3,
      color: "green",
      colorCode: 3,
      state: "paired"
    }
  }
}
```

5.2 Deregister Wristband

5.2.1 Client request

```
// 1st argument, required, Player
{
  username: "test",
  name: 'test',
  surname: 'test',
  email: 'testt@gmail.com',
}
// 2nd argument, required, Wristband
{
  id: 3,
  color: ’green’,
  colorCode: 3
}
// 3rd argument, optional, options
{
  queue: true || false
}
```

5.2.2 Backend request

```
{
  timestamp : 1706960913052,
  username : "a39hldmki3",
  wristbandNumber : 432
}
```

5.2.3 Backend response

```
{
  timestamp : 1706960913123,
  result : "OK",
  message : "successfully unregisterWristbandToPlayer"
}
```

5.2.4 Client response

```
{
  ok: true,
  player: {
    username: "test",
    name: ’test’,
    surname: ’test’,
    email: ’testt@gmail.com’,
    wristband: {
      id: 3,
      color: "green",
      colorCode: 3,
      state: "unpaired"
    }
  }
}
```

5.3 Get Wristband Information

5.3.1 Client request

```
{
  id: 3,
  colorCode: 3,
  color: "green",
  state: "state",
}
```

5.3.2 Backend request

```
{
  timestamp: 1706879364557,
  wristbandNumber: 3
}
```

5.3.3 Backend response

```
{
  timestamp: 1706879364557,
  result: 'OK',
  wristband: { wristbandNumber: 3, wristbandColor: 2, active: false }
}
```

5.3.4 Client response

```
{
  ok: true
  wristband: {
    id: 3,
    color: 'green',
    colorCode: 3,
    state: 'state',
  },
}
```

5.4 Scan Wristband

5.4.1 Client request

```
// 1st argument, required, unsubcb
(unsub) => {...}
// 2nd argument, optional, options
{
  queue: false || true
}
```

5.4.2 Backend request

```
// null
```

5.4.3 Backend response

```
{
  timestamp: 1706880614077,
  result: 'OK',
  wristbandNumber: 3,
  wristbandColor: 3
}
```

5.4.4 Client request

```
{
  ok: true
  wristband: { id: 3, color: 'green', colorCode: 3, state: 'unpaired' },
  unsubed: false,
}
```

6 Cashier Tasks

6.1 List Cashiers

6.1.1 Client request

```
// 1st argument, optional, options
{
  queue: true | false
}
```

6.1.2 Backend request

```
{
  timestamp: 1706707779283,
}
```

6.1.3 Backend response

```
{
  timestamp: 1706707779283,
  result: 'OK',
  cashiers: [
    { id: 1, username: 'pavlos', email: 'pavlosTester123@gmail.com' },
    { id: 3, username: 'tt', email: 'tt@gmail.com' }
  ]
}
```

6.1.4 Client response

```
{
  ok: true,
  cashiers: [
    {
      id: 1,
      username: 'pavlos',
      email: 'pavlosTester123@gmail.com',
      role: 'cashier'
    },
    {
      id: 8,
      username: 'TEST',
      email: 'test@gmail.com',
      role: 'cashier'
    }
  ]
}
```

6.2 Login Cashier

6.2.1 Client request

```
// 1st argument, required, cashier
```

```
{
  id: 3,
  username: "test",
  email: "test@gmail.com",
  role: "cashier",
}
// 2nd argument, required, password
password: "testpass"
// 3nd argument, optional, options
{
  queue: true | false
}
```

6.2.2 Backend request

```
{
  username: "33rksrlppga",
  password: "7c38dir1206",
}
```

6.2.3 Backend response

```
{
  timestamp: 1706777994830,
  result: 'OK',
  jwtResponse: {
    jwt: 'eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiIzM3Jrc3JscHBnYSIsImlhdCI6MTcwNjc3Nzk5NCwiZXhw
    id: 74,
    username: '33rksrlppga',
    email: '33rksrlppga@gmail.com',
    roles: [ 'ROLE_CASHIER' ]
  }
}
```

6.2.4 Client response

```
{
  ok: true,
  cashier: {
    id: 3,
    username: "test",
    email: "test@gmail.com",
    role: "cashier",
  },
  jwt: 'eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiIzM3Jrc3JscHBnYSIsImlhdCI6MTcwNjc3OTAxMywiZXhwIj
}
```

6.3 Register Cashier

6.3.1 Client request

```
// 1st argument, required, cashier
{
  username: "test",
  email: "test@gmail.com",
  role: 'test',
}
// 2nd argument, required, password
password: "oteuheno",
// 3nd argument, optional, options
{
  queue: true | false
}
```

6.3.2 Backend request

```
{
  username: "testCashier",
  email: "testCashier@gmail.com",
  password: "testCashierPassword",
  role: ["ROLE_CASHIER"],
}
```

6.3.3 Backend response

```
{
  timestamp: 1706729341301,
  result: 'OK'
}
```

6.3.4 Client response

```
{
  ok: true,
  cashier: {
    id: 3
    username: "test",
    email: "test@gmail.com",
    role: "cashier",
  },
  password: "testpass",
}
```

6.4 Deregister Cashier

6.4.1 Client request

```
// 1st argument, required, cashier
{
  id: 3,
  username: "test",
  email: "test@gmail.com",
  role: "cashier",
}
// 2nd argument, optional, options
{
  queue: true | false
}
```

6.4.2 Backend request

```
{
  timestamp: 1706732989145,
  username: "tt",
  userId: 3,
}
```

6.4.3 Backend response

```
{
  timestamp: 1706732989145,
  result: 'OK',
  cashiers: [
    { id: 1, username: 'pavlos', email: 'pavlosTester123@gmail.com' },
    { id: 5, username: 'testCashier', email: 'testCashier@gmail.com' },
    { id: 6, username: 'testCash', email: 'testCash@gmail.com' },
    { id: 7, username: 'r9rcnpncmrf', email: 'Tom@gmail.com' },
    { id: 8, username: 'ci10l5jm4ip', email: 'Finwe@gmail.com' },
    { id: 9, username: '9r0d6jqctfp', email: 'Elrond@gmail.com' },
    { id: 10, username: 'xi87q2qgu6', email: 'Gimli@gmail.com' },
    { id: 11, username: '2b6rdbkpl6j', email: 'Gilgalad@gmail.com' },
    { id: 13, username: 'mpw14t0s9jg', email: 'Isildur@gmail.com' },
    { id: 14, username: 'qbavrn3kw7', email: 'Aragorn@gmail.com' },
    { id: 15, username: 'jq6ttl0bueg', email: 'Maedhros@gmail.com' },
    { id: 16, username: 'ko1b9haqpqh', email: 'Thorin@gmail.com' },
    { id: 17, username: 'x21gpwr0bnm', email: 'Beren@gmail.com' },
    { id: 18, username: 'face6c6oojv', email: 'Celebrimbor@gmail.com' },
    { id: 19, username: '4i4asuxctvr', email: 'Theoden@gmail.com' },
    { id: 20, username: 'jj7mvpbsco4', email: 'Earendil@gmail.com' }
  ]
}
```

6.4.4 Client response

```
{
  ok: true,
  cashier: {
    id: 3,
    username: "test",
    email: "test@gmail.com"
    role: "cashier"
  }
}
```

7 Package Tasks

7.1 List Packages

7.1.1 Client request

```
// 1st argument, optional, options
{
  queue: true | false
}
```

7.1.2 Backend request

```
// null
```

7.1.3 Backend response

```
{
  timestamp: 1706640606387,
  result: 'OK',
  packages: [
    { name: 'Per Mission 5', amount: 5, type: 'mission', cost: 50 },
    {
      name: 'Per Mission 10',
      amount: 10,
      type: 'mission',
      cost: 100
    },
    {
      name: 'Per Mission 15',
      amount: 15,
      type: 'mission',
      cost: 150
    },
    {
      name: 'Per Mission 20',
      amount: 20,
      type: 'mission',
      cost: 200
    },
    { name: 'Per Time 30', amount: 30, type: 'time', cost: 50 },
    { name: 'Per Time 60', amount: 60, type: 'time', cost: 100 },
    { name: 'Per Time 90', amount: 90, type: 'time', cost: 150 },
    { name: 'Per Time 120', amount: 120, type: 'time', cost: 200 }
  ]
}
```

7.1.4 Client response

```
{
  ok: true,
  packages: [
    {
      id: null,
      name: 'Per Mission 5',
      type: 'mission',
      amount: 5,
      cost: 50,
      t_start: null,
      t_end: null,
      remainder: null,
      state: 'registered'
    },
    {
      id: null,
      name: 'Per Mission 10',
      type: 'mission',
      amount: 10,
      cost: 100,
      t_start: null,
      t_end: null,
      remainder: null,
      state: 'registered'
    },
    {
      id: null,
      name: 'Per Mission 15',
      type: 'mission',
      amount: 15,
      cost: 150,
      t_start: null,
      t_end: null,
      remainder: null,
      state: 'registered'
    },
    {
      id: null,
      name: 'Per Mission 20',
      type: 'mission',
      amount: 20,
      cost: 200,
      t_start: null,
      t_end: null,
      remainder: null,
      state: ’registered’
    },
    {
      id: null,
      name: ’Per Time 30’,
      type: ’time’,
      amount: 30,
      cost: 50,
      t_start: null,
      t_end: null,
      remainder: null,
      state: ’registered’
    },
    {
      id: null,
      name: ’Per Time 60’,
      type: ’time’,
      amount: 60,
      cost: 100,
      t_start: null,
      t_end: null,
      remainder: null,
      state: ’registered’
    },
    {
      id: null,
      name: ’Per Time 90’,
      type: ’time’,
      amount: 90,
      cost: 150,
      t_start: null,
      t_end: null,
      remainder: null,
      state: ’registered’
    },
    {
      id: null,
      name: ’Per Time 120’,
      type: 'time',
      amount: 120,
      cost: 200,
      t_start: null,
      t_end: null,
      remainder: null,
      state: 'registered'
    }
  ]
}
```

8 Team Tasks

8.1 Register Team

8.1.1 Client request

```
// 1st argument, required, Team
{
  name: 'elated_Galadriel_cl4piph2kic',
  t_created: null,
  points: 0,
  state: 'unregistered'
  roster: [
    {
      username: 'lqplk9p1w68',
      name: 'Finwe',
      surname: 'laughing',
      email: 'lqplk9p1w68@gmail.com',
      state: 'registered',
      wristband: { id: 351, color: 'red', colorCode: 1, state: 'paired' }
    },
    {
      username: 'pgs5ssie3',
      name: 'Eowyn',
      surname: 'strange',
      email: 'pgs5ssie3@gmail.com',
      state: 'registered',
      wristband: { id: 253, color: 'orange', colorCode: 6, state: 'paired' }
    }
  ]
}
// 2nd argument, optional, Options
{
  queue: true || false
}
```

8.1.2 Backend request

```
{
  timestamp : 1706979526513,
  teamName : "testTeam",
  usernames : [ "9qqu592xhrg", "g0dh1umskej" ]
}
```

8.1.3 Backend response

```
{
  timestamp : 1706979526580,
  result : "OK",
  message : "successfully created team: tziros1"
}
```

8.1.4 Client response

```
{
  ok: true,
  team: {
    name: 'elated_Galadriel_cl4piph2kic',
    t_created: 1707028052944,
    points: 387,
    packages: [],
    roster: [
      {
        username: 'lqplk9p1w68',
        name: 'Finwe',
        surname: 'laughing',
        email: 'lqplk9p1w68@gmail.com',
        state: 'inTeam',
        wristband: { id: 351, color: 'red', colorCode: 1, state: 'paired' }
      },
      {
        username: 'pgs5ssie3',
        name: 'Eowyn',
        surname: 'strange',
        email: 'pgs5ssie3@gmail.com',
        state: 'inTeam',
        wristband: { id: 253, color: 'orange', colorCode: 6, state: 'paired' }
      }
    ],
    state: 'registered',
  }
}
```

8.2 Register Team Package

8.2.1 Client request

```
// 1st argument, Team
{
  {
    name: 'elated_Galadriel_cl4piph2kic',
    t_created: null,
    points: 0,
    state: 'unregistered'
    roster: [
      {
        username: 'lqplk9p1w68',
        name: 'Finwe',
        surname: 'laughing',
        email: 'lqplk9p1w68@gmail.com',
        state: 'registered',
        wristband: { id: 351, color: 'red', colorCode: 1, state: 'paired' }
      },
      {
        username: 'pgs5ssie3',
        name: 'Eowyn',
        surname: 'strange',
        email: 'pgs5ssie3@gmail.com',
        state: 'registered',
        wristband: { id: 253, color: 'orange', colorCode: 6, state: 'paired' }
      }
    ]
  }
}

// 2nd argument, package
{
  id: null,
  name: 'Per Mission 5',
  type: 'mission',
  cost: 50,
  t_start: null,
  t_end: null,
  amount: 5,
  remainder: null,
  state: 'unregistered'
}

// 3rd argument, options
{ queue: false || true }
```

8.2.2 Backend request

```
{
  timestamp : 1707053008561,
  teamName : "hopeful_Feanor_ng2coekx3lc",
  name : "Per Time 30"
}
```

8.2.3 Backend response

```
{
  timestamp : 1707053008626,
  result : "OK",
  team : {
    name : "hopeful_Feanor_ng2coekx3lc",
    totalPoints : 0,
    teamState : null,
    created : null,
    lastRegisterAttempt : null,
    currentRoster : {
      version : 1,
      players : [ {
        username : "c77r5w5mod2",
        wristbandNumber : 455,
        wristbandColor : null
      }, {
        username : "hndfw7wu1a",
        wristbandNumber : 347,
        wristbandColor : null
      } ]
    },
    roomType : null,
    packages : [ {
      id : 10,
      name : "Per Time 30",
      cost : null,
      started : null,
      ended : null,
      duration : 1800.000000000,
      paused : false,
      active : false
    } ]
  }
}
```

8.2.4 Client response

```
{
  ok: true,
  team: {
    ...
    packages: [
      ...,
      {
        id: 3,
        name: 'Per Mission 5',
        type: "mission",
        cost: null,
        amount: 5,
        remainder: 5,
        t_start: null,
        t_end: null,
        state: "registered",
      }
    ]
  }
}
```

8.3 Deregister Team Package

8.3.1 Client request

```
// 1st argument, Team
{
  {
    name: ’elated_Galadriel_cl4piph2kic’,
    t_created: null,
    points: 0,
    state: ’unregistered’
    roster: [
      {
        username: ’lqplk9p1w68’,
        name: ’Finwe’,
        surname: ’laughing’,
        email: ’lqplk9p1w68@gmail.com’,
        state: ’registered’,
        wristband: { id: 351, color: ’red’, colorCode: 1, state: ’paired’ }
      },
      {
        username: ’pgs5ssie3’,
        name: ’Eowyn’,
        surname: ’strange’,
        email: ’pgs5ssie3@gmail.com’,
        state: ’registered’,
        wristband: { id: 253, color: ’orange’, colorCode: 6, state: ’paired’ }
      }
    ]
  }
}
```

```
// 2nd argument, package
{
  id: null,
  name: 'Per Mission 5',
  type: 'mission',
  cost: 50,
  t_start: null,
  t_end: null,
  amount: 5,
  remainder: null,
  state: 'unregistered'
}

// 3rd argument, options
{ queue: false || true }
```

8.3.2 Backend request

```
{
  timestamp : 1707056780735,
  teamName : "affectionate_Shelob_ct4pqxcce8w",
  packageId : 17
}
```

8.3.3 Backend response

```
{
  timestamp : 1707056780791,
  result : "OK",
  team : {
    name : "affectionate_Shelob_ct4pqxcce8w",
    totalPoints : 0,
    teamState : null,
    created : null,
    lastRegisterAttempt : null,
    currentRoster : {
      version : 1,
      players : [ {
        username : "g9781e0di69",
        wristbandNumber : 401,
        wristbandColor : null
      }, {
        username : "18tw5isjpd7e",
        wristbandNumber : 421,
        wristbandColor : null
      } ]
    },
    roomType : null,
    packages : [ ]
  }
}
```

8.3.4 Client response

```
{
  ok: true,
  team: {
    ...,
    packages: []
  }
}
```

8.4 Start team

8.4.1 Client request

```
// 1st argument, team
{
  ...
},

// 2nd argument options
{ cause: true || false };
```

8.4.2 Backend request

```
{
  timestamp : 1707060079874,
  teamName : "compassionate_Melian_ktl66x5o73f"
}
```

8.4.3 Backend response

```
{
  timestamp : 1707060079952,
  result : "OK",
  team : {
    name : "compassionate_Melian_ktl66x5o73f",
    totalPoints : 0,
    teamState : null,
    created : null,
    lastRegisterAttempt : null,
    currentRoster : {
      version : 1,
      players : [ {
        username : "cdc0t3lfjfg",
        wristbandNumber : 154,
        wristbandColor : null
      }, {
        username : "97tixfvlwsp",
        wristbandNumber : 255,
        wristbandColor : null
      } ]
    },
    roomType : null,
    packages : [ {
      id : 21,
      name : "Per Mission 10",
      cost : null,
      started : 1707060079921,
      ended : null,
      missions : 10,
      missionsPlayed : 0,
      active : true
    } ]
  }
}
```

8.4.4 Client response

```
{
  ...team,
  state: "playing",
}
```

8.5 Find Team

8.5.1 Client request

```
// 1st arument
{
  ...team,
},
// 2nd argument, options
{
  cause: true || false,
}
```

8.5.2 Client response

```
{
  team: { ... },
}
```

9 Session Tasks

9.1 List Session

9.1.1 Backend response

```
{
  timestamp: 1708773525572,
  result: 'OK',
  message: 'No active session'
}

// Or
{
  timestamp: 1709248857023,
  result: 'OK',
  message: '{"session":{"current":true,"created":"2024-02-29 19:10:31.306","ended":"nu
}
```

9.1.2 Client response

```
{
  session: {
    current: true,
    created: '2024-02-29 19:10:31.306',
    ended: 'null',
    started: '2024-02-29 19:10:31.305',
    updated: '2024-02-29 19:10:31.306',
    user: {
      active: true,
      id: 8,
      email: 'test@gmail.com',
      username: 'TEST'
    },
    active: true
  },
  ok: true,
  msg: 'Successfully retrieved Session'
}
```

9.2 Start Session

9.2.1 Client request

```
// 1st argument, required, cashier
{
  id: 3,
  username: "test",
  email: "test@gmail.com",
  role: "cashier",
}
// 2st argument, required, jwt
{
  jwt: "eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiIzM3Jrc3JscHBnYSIsImlhdCI6MTcwNjc3Nzk5NCwiZXhwI
}
// 3st argument, optional, options
{
  queue: true | false
}
```

9.2.2 Backend request

```
{
  jwt: "eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiIzM3Jrc3JscHBnYSIsImlhdCI6MTcwNjc3Nzk5NCwiZXhwIj
}
```

9.2.3 Backend response

```
{
  timestamp: 1706780850379,
  result: 'OK',
}
```

9.2.4 Client request

```
{
  ok: true,
  cashier: {
    id: 3,
    username: "test",
    email: "test@gmail.com",
    role: "cashier",
  },
  jwt: "eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiIzM3Jrc3JscHBnYSIsImlhdCI6MTcwNjc3Nzk5NCwiZXhwIj
}
```

9.3 Stop Session

9.3.1 Client request

```
// 1st argument, required, cashier
{
  id: 3,
  username: "test",
  email: "test@gmail.com",
  role: "cashier",
}
// 2st argument, required, jwt
{
  jwt: "eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiIzM3Jrc3JscHBnYSIsImlhdCI6MTcwNjc3Nzk5NCwiZXhwIj
}
// 3st argument, optional, comment
```

```
comment: "Nothing unexpected ever happens!"
// 4th argument, optional, options
{
  queue: true | false
}
```

9.3.2 Backend request

```
{
  jwt: "eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiIzM3Jrc3JscHBnYSIsImlhdCI6MTcwNjc3Nzk5NCwiZXhwIj
  comment: "Nothing unexpected ever happens!"
}
```

9.3.3 Backend response

```
{
  timestamp: 1706780850379,
  result: 'OK',
}
```

9.3.4 Client response

```
{
  ok: true,
  cashier: {
    id: 3,
    username: "test",
    email: "test@gmail.com",
    role: "cashier",
  },
}
```

9.4 Stop Sesson force

10 The Command pattern

The **afmachine** or **afm** for short is the control center of the application.

It's API is consumed by invoking any of the public methods of the **afm** instance.

All API calls are stored under the directory:

/src/afmachine/tasks/* /src/afmachine/synthetic-tasks/*

Each of these is usually a wrapper for a backend API call.

The backend API calls are found at:

/src/afmachine/device/admin-screen/* /src/afmachine/device/rpi-reader/*

The primary function of **Afmachine** is to allow clients (such as a React component or the UI in general) to build middleware chains around each Task. This design model follows loosely the **command** design pattern. see (`https://en.wikipedia.org/wiki/Command_pattern`)

Each time an API call is made **afm** creates a **Command** for the invoked **Task**. A **Command** represents one instance of a running **Task**.

For example: If a client was to invoke an API call multiple times:

afm.listPkgs() afm.listPkgs() afm.listPkgs()

There would be 3 **Commands** created for the listPkgs **Task**.

Each **Command** carries with it a lot of information but fundamentally it contains a sequence of functions (the middleware chain) to run.

When a command is created it is placed into a queue by afm. The afm is responsible for invoking each the commands in the queue in sequence (meaning in order, waiting for the completion of one to carry on with the next).

For example: If a client was to invoke the API calls:

afm.loginCashier(); afm.listPkgs(); afm.logoutCashier();

There would be 3 **Commands** to run in the queue by **afm**. It would proceed by running them in these order:

1. loginCashier()
2. listPkgs()
3. logoutCashier()

Clients to afm can register functions that become part of a Command through the Eventful interface at:

/src/Eventful.js

For exmaple:

// Register a hook to run before each command in the queue. afm.on('precmd', (afm) => {});

// Register a hook to run after each command in the queue. afm.on('postcmd', (afm) => {});

// Register a hook to run before each Command for the listPkgs Task. afm.listPkgs.on('precmd', (command, next) => {});

// Register a hook to run after each Command for the listPkgs Task. afm.listPkgs.on('postcmd', (command, next) => {});

11 Conventions

11.1 Command inputs and outputs

```
{
  args: {
    ...AFM_FORM_INPUTS
  }
  req: {
    ...BACKEND_FORM_INPUTS
  }
  raw: {
    ...BACKEND_FORM_OUTPUTS
  }
  res: {
    ...AFM_FORM_OUPUTS
  }
}
```

11.2 Commands wrap their return value within an object

Commands never return an entity object, they return an object that contains the entity or entities.

```
// Instead of:
const response = {
  username: "...",
  name: "...",
}

// This:
const response = {
  player: {
    username: "...",
    name: "...",
  },
}
```

12 Normalization functions

Normalization function take an array of source objects and combines them into one object. It is similar to Object.assign(target, . . . sources) but adjusted to fit the special needs of the ENTITY being normalized. (such as a Player, Wristband etc). To be more exact, each normalization function is responsible for:

- TRANSLATION of an object in backend form to frontend form.
- DEDUCING the state of the entity.
- MERGING of multiple sources in any form.
- NORMALIZATION of the NESTED entities contained within, if any.

12.1 Inputs

All normalization functions accept the following inputs:

- an array of sources, or a single object, or null, or the empty object etc. . .
- An Options object

To pass along instructions to a NESTED normalization function, one must begin a secondary nesting in the Options object.

For example:

```
// Considering a Composite Entity such as a Team which
// contains within Players with each owning a Wristband and Packages.
// Team -> Players -> Wristband
// Team -> Packages
{
  nullSupersede: true, // Team target
  state: 'registered', // Team Target
  package: { // NESTED TARGET -> Package
    state: "playing",
  }
  player: { // NESTED TARGET -> Player
    state: "inTeam",
  }
  wristband: { // NESTED TARGET -> Player -> Wristband
    state: "paired",
    nullsSupersede: false
  }
}
```

12.2 Deducing state

Deducing state is carried on in 2 stages. The goal is to reduce ambiguity an introduce Determinism.

The first stage is about interpreting the properties that carry with them State.

For example, given a Package with a non-nil id property:

```
// Package
{
  id: 5
}
```

I could interpret the existence of a non-nil ID property as indicative of the Package being at least Registered.

But I do not (not in the 1st stage). The only properties used in the first stage in that EXACT order of a Package normalization function are:

- targetState The targetState is passed as an option to the normalization function. If it has been defined it interrupts the function and returns immediately operating under the assumption that the caller already knows what the state of the target should be.
- active This property is found in a BackendForm package. If it is defined and true it means the Package is currently active.
- state This property is found in an AfmForm package.
- defaultState The defaultState is passed as an option to the normalization function. If no state has been deduced so far in the process, it is used.

The general pattern is that, the targetState has the highest precedence. It is followed by State backendForm, then by afmForm State, then by defaultState.

In the tobject() functions which must also deduce state the order is: targetState, afmForm, defaultState.

This is the end of the 1st stage.

The 2nd stage is responsible for binding the Content of the entity to its State.

What do I mean by that?.

Some properties such as:

state or active (in a Package)

are explicitly designed to convey state.

While the other properties in a Package are about its Contents.

```
{
  id,
  t_started,
  t_ended,
  remainder,
  amount,
  ...
}
```

Therefore, one may allocate each Property as belonging to:

- State
- Content

So the 2st stage is about making sure that the target's Contents align with the State from the 1st stage.

So again carrying on with the above Example:

A Package can be in one of four states:

- Unregistered
- Registered
- Playing
- Completed

(-START NOTE-) I believe that the architecture of the backend should be based on Packages. What do I mean? Instead of having Teams with Packages and Teams having an active Package. You have Packages with a 'has-a' Team. So instead of (1)Team-(*)Packages you have a (1)Package-(1)Team. But it is not, so I make best with what I have. (-END NOTE-)

Lets say that the target after stage 1 looks like these:

```
{
  state: 'unregistered'
  id: 5,
  t_start: 100,
  t_end: 500,
}
```

This is an example of a misalignment. The State says that this Package is unregistered but the Contents say that it is Completed.

What should be done in this situation?

In order to help the developer know when a situation like this arises a state Error is thrown.

So the 2nd stage is responsible for making sure that a misalignment never occurs.

13 Schemas

13.1 Package

13.1.1 AFM Time

```
{
  id: 3,
  name: 'Per Time 90',
  amount: 99999 // milliseconds
  type: "time",
  cost: 90.99,
  amount: 888, // milliseconds,
  remainder: 123, // milliseconds,
  t_start: 1232434324, // milliseconds,
  t_end: 1234234234, // milliseconds
  state: "unregistered" | "registered" | "playing" | "completed"
}
```

13.1.2 AFM Missions

```
{
  id: 3,
  name: 'Per Mission 5',
  amount: 5 // missions
  type: "mission",
  cost: 90.99,
  amount: 5, // missions
  remainder: 1, // missions,
  t_start: 1232434324, // milliseconds,
  t_end: 1234234234, // milliseconds
  state: "unregistered" | "registered" | "playing" | "completed"
}
```

13.1.3 Backend Time

```
// team's package
{
  id: 5,
  name: 'Per Time 90',
  cost: null,
  started: 1706685129723, // milliseconds
  ended: null, // milliseconds
  duration: 5400, // seconds
  paused: false,
  active: true
}
```

13.1.4 Backend Missions

```
// team's package
{
  id: 1,
  name: 'Per Mission 5',
  cost: null,
  started: null, // milliseconds
  ended: null, // milliseconds
  missions: 5, // missions
  missionsPlayed: 0, // missions
  active: false
}
```

13.1.5 Available Backend packages

```
// Missions
{
  name: "Per Mission 5",
  amount: 5, // Missions
  type: "mission",
  cost: 150
}

// time
{
  name: "Per Time 90",
  amount: 90, // minutes
  type: 'time',
  cost: 150
}
```

13.2 Device

13.2.1 AFM rpi reader

```
{
  id: 'ADMINISTRATION1Reader',
  type: 'RPI_READER',
  room: 'ADMINISTRATION1'
}
```

13.2.2 AFM admin screen

```
{
  id: '001',
  type: 'REGISTRATION_SCREEN',
  room: 'ADMINISTRATION1'
}
```

13.2.3 Backend

```
{
  deviceId: '001',
  deviceType: 'REGISTRATION_SCREEN',
  roomType: 'ADMINISTRATION1'
}
```

13.3 Cashier

13.3.1 AFM

```
{
  id: 4394,
  username: '80teepo7fu9',
  email: '80teepo7fu9@gmail.com',
  role: 'manager'
}
```

13.4 Player

13.4.1 AFM

```
{
  username: "test"
  name: "testname",
  surname: "testsurname",
  email: "test@gmail.com",
  state: 'unregistered' || 'registered' || 'inTeam' || 'playing'
  wristband: {
    id: null || 3,
    color: null || 'green',
    colorCode: null || 3,
    state: "unpaired" || "pairing" || "unpairing" || "paired"
  }
}
```

13.4.2 Backend

```
{
  username: "test",
  name: "test",
  surname: "test",
  email: "test@gmail.com",
  wristbandMerged: true || false,
}
```

13.5 Team

13.6 Wristband

13.6.1 AFM

```
{
  id: 3,
  color: "green",
  colorCode: 2,
  state: "unpaired" || "pairing" || "unpairing" || "paired"
}
```

13.6.2 Backend

```
// wristband register
{
  timestamp : 1706957679789,
  username : "diwgp3nrrtf",
  wristbandNumber : 234
}

// wristband deregister
{
  timestamp : 1706960913123,
  result : "OK",
  message : "successfully unregisterWristbandToPlayer"
}

// wristband info
{
  timestamp: 1706879364557,
  result: 'OK',
  wristband: { wristbandNumber: 3, wristbandColor: 2, active: false }
}

// wristband scan
{
  timestamp: 1706880614077,
  result: 'OK',
  wristbandNumber: 3,
  wristbandColor: 3
}


// list registered players + search players
{
  username: 'Merry_2mpmnxcgv1s',
  name: 'Merry',
  surname: 'compassionate',
  email: 'Merry@gmail.com',
  wristbandMerged: false,
  wristband: null
},
{
  username: 'Wormtongue_klagnkjxqla',
  name: 'Wormtongue',
  surname: 'jovial',
  email: 'Wormtongue@gmail.com',
  wristbandMerged: false,
  wristband: { wristbandNumber: 230, wristbandColor: 3, active: true }
},

// list registered players with writband
{
  username: 'Gandalf_deil7sv8j4c',
  name: 'Gandalf',
  surname: 'busy',
  email: 'Gandalf@gmail.com',
  wristbandMerged: false,
  wristband: { wristbandNumber: 233, wristbandColor: 4, active: true }
},

// list teams
players: [
  {
    username: 'test1',
    wristbandNumber: null,
    wristbandColor: null
  },
  {
    username: "test3",
    wristbandNumber: 1,
    wristbandColor: null,
  },
  {
    username: "test4",
    wristbandNumber: null,
    wristbandColor: 2,
  },

  {
    username: "Sauron_0h96h9q4xixv",
    wristbandNumber: 241,
    wristbandColor: 2,
  },

]
```