

Endpoints specification

Status code en commun à toutes les réponses

Status : 401 Unauthorized

Response body :

```
{  
  "error": "Authentication token not found in headers or token has expired."  
}
```

Status : 500 Internal Service error

Response body :

```
{  
  "error": "The request was not completed due to an internal error on the server  
side."  
}
```

Status : 404 Not Found

Response body :

```
{  
  "error": "The requested resource was not found."  
}
```

Status : 503 Service unavailable

Response body :

```
{  
  "error": "The server was unavailable."  
}
```

/account

/account/inscription ✓

Method : POST

Need authentication : No

Request body :

```
{
  "username": "root",
  "password": "password",
  "email": "mail@tropbien.com",
  "birthdate": "YYYY-mm-dd",
  "lastName": "lastName",
  "name": "name",
  "bio": "bio"
  "idImage" : "idImage"
}
```

Status : 201 Created

Response body :

```
{}
```

Status : 400 Bad request

Response body :

```
{
  "error": "Missing parameter username"
}
```

/account/login ✓

Method : POST

Need authentication : No

Request body :

```
{
  "username": "root",
  "password": "password"
}
```

Status : 200 OK

Response body :

```
{
  "authToken": "revrverazfvrageqregvearqverzverz"
}
```

Status : 400 Bad request

Response body :

```
{
  "error": "Missing parameter username"
}
```

/account/logout ✓

Method : GET

Need authentication : Yes

Status 204 OK (No response body)

/account/update ✓

Method : PUT

Need authentication : Yes

Request body :

```
{
  "current_password": "current_password",
  "username": "root",
  "password": "password",
  "email": "mail@tropbien.com",
  "birthdate": "YYYY-mm-dd",
  "lastName": "lastName",
  "name": "name",
  "bio": "bio"
  "idImage": "idImage"
}
```

(tous les champs sont optionnels sauf current_password)

Status 204 OK (No response body)

Fail : same Username or email

Status : 409 Conflict

Response body :

```
{
  "error": "Username already taken !"
}
```

Fail : Wrong current_password

Status : 403 Forbidden

Response body :

```
{
  "error": "Password mismatch, you are not allowed to modify the profil !"
}
```

/account/info ✓

Method : GET

Need authentication : Yes

Request body : None

Status 200 OK

Response body :

```
{
  "username": "username",
  "lastname": "lastname",
  "name": "name",
  "idImage": "idImage",
  "biography": "biography",
  "mail": "mail",
  "birthDate": "dd/mm/YYYY",
  "groupIds": [ 1, 2, 43 ]
}
```

/account/task/all ✓

Method : GET

Need authentication : Yes

Request body : None

Status 200 OK

Response body

```
{
  "tasks" : [
    {
      "taskId": 325,
      "name": "Ménage",
      "description": "Description de tâches",
      "taskUser": "Fatima",
      "frequency": 5,
      "priority": 2,
      "datetimeStart" : "2020-12-14 11:25:36",
      "datetimeEnd": "2020-12-14 12:30:00",
      "dependancies" : [ 324,323,322 ]
    }
  ]
}
```

/account/task/upcoming ✓

Method : GET

Need authentication : Yes

Request body : None

Status 200 OK

Response body

```
{
  "tasks" : [
    {
      "taskId": 325,
      "name": "Ménage",
      "description": "Description de tâches",
      "taskUser": "Fatima",
      "frequency": 5,
      "priority": 2,
      "datetimeStart" : "2020-12-14 11:25:36",
      "datetimeEnd": "2020-12-14 12:30:00",
      "dependancies" : [ 324,323,322 ]
    }
  ]
}
```

/account/invitation ✓

Method : GET

Need authentication : Yes

Request body : None

Status 200 OK

Response body :

```
{
  "invitations" : [
    {
      "invitationId": 1,
      "sender": "Patrick",
      "group": "Le group trop cool",
      "groupId": 4,
      "idImageGroup": 45
    },
    {
      "invitationId": 4,
      "sender": "Michel",
      "group": "Le group trop cool 2",
      "groupId": 78,
      "idImageGroup": 13
    }
  ]
}
```

```
]
}
```

/account ✓

Method : GET

Need authentication : Yes

Request body : None

Request parameter : username

Status 200 OK

Response body :

```
{
  "users": [
    "user1",
    "user2"
  ]
}
```

/group

/group ✓

Method: GET ✓

Request body : None

Status 200 OK

Response body :

```
{
  "groups" :
  [
    {
      "groupId": 4,
      "group_name": "La famille",
      "idImageGroup": "1"
    },
    {
      "groupId": 6,
      "group_name": "collocation",
      "idImageGroup": "4"
    }
  ]
}
```

//create a new group and new invitations

Method: POST ✓

Request body :

```
{
  "group_name": "name",
  "description": "description"
  "guests" : [ "username1", "username2", ... ]
  "idImageGroup" : 5
}
```

// Utiliser le header pour ajouter le créateur du groupe

Status 204 OK (No response body)

Status 400 Bad request ("Make sure to send all the parameters")

Status 409 Conflict

/group/<id_group> ✓

Method: GET

Request body : None

Status 200 OK

Response body :

```
{
  "id" : 212,
  "group_name": "name",
  "description": "",
  "members" : [ "username1", "username2", ... ]
  "idImageGroup" : 5
}
```

/group/<id_group>/update ✓

METHOD : PUT

Request body :

```
{
  "group_name": "Democrateam",
  "description" : "les best"
}
```

Status 400, "Bad Request: Make sure to send all parameters !"

/group/<id_group>/quit ✓

Supprime l'utilisateur de la table membre

Method: GET

Request body : None

Status 204 OK (No response body)

/group/<id_group>/invite ✓

Crée une nouvelle invitation ✓

Method: GET

Request parameters : username

Request body : None

Status 204 OK (No response body)

Status code : 409 Conflict

Response body :

```
{
    "error" : "The request could not be completed due to a conflict. User may have
already been invited."
ou
    "error" : "The request could not be completed due to a conflict. User is already in the
group."
}
```

Method: DELETE ✓

Status 204 OK (No response body)

/group/<id_group>/accept ✓

Ici supprime l'invitation et rajoute l'utilisateur comme membre du groupe

Method: GET

Request parameters : invite_id

Request body : None

Status 204 OK (No response body)

/group/<id_group>/task

/group/<id_group>/task/all ✓

Method: GET

Request body : None

Status 200 OK

Response body :

```
{
  "tasks" :
  [
    {
      "taskId": 325,
      "name": "Ménage",
      "description": "Description de tâches",
      "taskUser": "Fatima",
      "frequency": 5,
      "priority": 2,
      "datetimeStart" : "2020-12-14 11:25:36",
      "datetimeEnd": "2020-12-14 12:30:00",
      "dependancies" : [ 324,323,322 ]
    },
    {...},
    {...}
  ]
}
```

group/<id_group>/task ✓

Je déplace la création d'un tâche car ce n'est techniquement pas le même endpoint

Method : POST ✓

Request body :

```
{
  "name" : "Ménage",
  "description" : "Une description de tache",
  "taskUser" : "Fatima",
  "frequency" : 5,
  "priority" : 1/2/3/...,
  "dependencies" : [ 324,323,322],
  "datetimeStart" : "2020-01-01 01:10:01",
  "datetimeEnd" : "2020-01-01 12:12:21",
  "duration": 14
}
```

duration est en minutes

Si datetimeStart et datetimeEnd sont renseigné, alors la valeur de duration est automatique calculer par le backend. Sinon il faut obligatoirement fournir une valeur pour duration.

(datetimeStart et datetimeEnd sont des champs optionnels)

Status 201 Created

Response body : {}

Fail to get value of request body

Status : 400 Bad request

Response body :

```
{
  "error": "Missing parameter taskUser"
}
```

Fail to find username or group to affect task

Status : 400 Bad request

Response body :

```
{
  "error": "User or group not found !"
}
```

/task/<id_task>✓

Method: GET✓

Status 200 OK

Response body :

```
{
  "taskId": 325,
  "name": "Ménage",
  "description": "Description de tâches",
  "taskUser": "Fatima",
  "frequency": 5,
  "priority": 2,
  "datetimeStart" : "2020-12-14 11:25:36",
  "datetimeEnd": "2020-12-14 12:30:00",
  "dependancies" : [ 324,323,322 ],
  "duration": 14,
  "idGroup" : 2
}
```

(datetimeStart et datetimeEnd peuvent être null)

Method : PUT✓

Request body :

```
{
  "taskId": 1,
  "name": "Ménage",
  "description": "Description de tâches",
  "taskUser": "Fatima",
  "frequency": 5,
  "priority": 2,
  "datetimeStart" : "2020-12-14 11:25:36",
  "datetimeEnd": "2020-12-14 12:30:00",
  "dependancies" : [ 324,323,322 ],
  "duration": 14
}
```

(l'ensemble des champs sont optionnels, seuls ceux présent seront mis à jour)

Le Front renvoie l'ensemble des éléments attention sur les champs pouvant être vide

Status 200 OK

Fail : Task doesn't exists:

Status 404 Not Found

Response body :

```
{
  "error": "Task not found !"
}
```

```
}
```

Method : DELETE ✓

Status 204 OK (No response body)

Status : 404 Not Found

Response body :

```
{
    "error": "The requested resource was not found or was already deleted."
}
```

/group/<id_group>/order

Method: GET ✓

Request parameter : startHour, startMinute, endHour, endMinute

Status 200 OK

Response body :

```
{
    "tasks_modified": [
        {
            "taskId": 2,
            "name": "Ménage",
            "description": "Description de tâches",
            "taskUser": "Fatima",
            "frequency": 5,
            "priority": 2,
            "datetimeStart" : "2020-12-14 11:25:36",
            "datetimeEnd": "2020-12-14 12:30:00",
            "dependancies" : [ 324,323,322 ],
            "duration": 14
        },
        {
            "taskId": 1,
            "name": "Ménage",
            "description": "Description de tâches",
            "taskUser": "Fatima",
            "frequency": 5,
            "priority": 2,
            "datetimeStart" : "2020-12-14 11:25:36",
            "datetimeEnd": "2020-12-14 12:30:00",
            "dependancies" : [ 324,323,322 ],
            "duration": 14
        }
    ]
}
```

Fail : Missing parameters:

Status 400 Bad Request

Response body :

```
{  
    "error": "You must send those parameters : startHour, startMinute, endHour,  
endMinute"  
}
```