

API Bilemo : Documentation

login

POST

/api/login

The request allowing a client or a user to acquire their authentication token for the Bilemo API.

Parameters :

- Name : "email"
 - Type : string
-
- Name : plainPassword
 - Type : string

Request body (JSON) :

```
{
  "email": "user@example.com",
  "plainPassword": "string"
}
```

Responses :

- Code : 200
 - Description : request was successful and the token was sent back.
-
- Code : 401
 - Message : "Invalid credentials."
 - Description : the data sent did not match any known credentials

Accessing Bilemo's cellphone catalogue

GET

/api/phones

The request allowing users to access the cellphone list proposed by Bilemo. It will send back the phone objects, linked with their respective associated entities. The elements will exhaustively described the product : brand, model, capacity and color.

Phones are sent by package of 30 items with the pagination to navigate the whole list.

Parameters :

- Name : "page"
- Description : the collection page number
- Type : integer

Responses :

- Code : 200
- Description : request was successful and sent back 30 products fully described

Accessing a specific cellphone

GET

/api/phones/{id}

The request allowing users to access a specific cellphone. It will send back the phone object, linked with its associated entities. The elements will exhaustively described the product : brand, model, capacity and color.

Parameters :

- Name : "id" (the cellPhone id)
- Type : integer

Responses :

- Code : 200
- Description : request was successful and sent back the detailed product

-
- Code : 500
 - Message : "not found"
 - Description : the product you tried to access does not exist in the database

Accessing a client's users list

GET

/api/users/clientUsers

The request allowing a client to access its user's list. The list displays the email and roles of the users. No parameters are needed, the users are selected on the base of the safety token to ensure no clients access users out of their property right.

Responses :

- Code : 200
- Description : request was successful and sent back the client's users list.

Accessing a specific user

GET

/api/users/{id}/clientUser

The request allowing a client to access the detail of one of its user. The user entity displays its email and roles.

Parameters :

- Name : "id" (the user id)
- Type : integer

Responses :

- Code : 200
 - Description : request was successful and sent back the detailed user
-
- Code : 500
 - Message : "not found"
 - Description : the product you tried to access does not exist in the database
-
- Code : 403
 - Message : "The user you are trying to access is not on your users list"
 - Description : the user the client tries to access is not on his list. He does not have the rights

Creating a new user

POST

/api/users

The request allowing a client to create a new user that will integrate its users list.

Parameters :

- Name : "email"
- Type : string

-
- Name : roles
 - Type : array

-
- Name : plainPassword
 - Type : string

Request body (JSON) :

```
{
  "email": "user@example.com",
  "roles": [
    "string"
  ],
  "plainPassword": "string"
}
```

Responses :

- Code : 200
 - Description : request was successful, the user is created in database
-

- Code : 500
 - Message : "An exception occurred while executing a query:
SQLSTATE[23000]: Integrity constraint violation: 1062 Duplicate [...] entry
for key [...]"
 - Description : The email is the account username. The API cannot tolerate two accounts with the same email.
-

- Code : 422
 - Message : "The password entered must use 6 characters minimum."
 - Description : Safety oriented demands regarded the password defined by the user.
-

- Code :422
 - Message : "The user email has to be entered"
 - Description : The email field cannot be blank
-

- Code :422
- Message : "The user password has to be entered"
- Description : The plainpassword field cannot be blank

Deleting a user

DELETE

/api/phones

The request allowing a client to delete a user of his users list from the database.

Parameters :

- Name : "id" (the user id)
- Type : integer

Responses :

- Code : 200
 - Description : request was successful, the user is deleted from database
-
- Code : 403
 - Message : "The user you are trying to access does not belong to you."
 - Description : The client tries to delete from the database a user that does not appears on his users list