

# API BileMo 1.0.0 OAuth2

This is documentation for the API BileMo.

This API works with **Google Authentication System**. It use **OAuth2 Protocol**. [Official documentation](#)

For implement this system, follow these steps :

- 1 - You must to register your application as a client on the [Google API SERVICE](#) and define a redirect URI.
- 2 - Google then gives you 2 keys : **ID\_client** and **ID\_secret**. You must store these informations.
- 3 - Then, you can connect to the google authentication page with the **Authorization URL** wich can be found in the **Authorize** section.
- 4 - Once you have logged in with your account, Google send you a **code** in query parameters with the **redirect URI** previously specified.
- 5 - Finally, for get an **acces\_token**, you must put this code in query parameter of the **Token URL** wich can be found in the **Authorize** section and send this.
- 6 - Google then gives you an **access\_token** and a **refresh\_token** in json response. This token is valid for **1 hour**. After, you can get another acces\_token with the **refresh\_token** (see the documentation)

You can automate these operations so that your application connects to your google account itself.

Servers

[https://127.0.0.1:8000 - API over HTTPS](#)

[Authorize](#)

## phones

GET /phones

Try it out

Name	Description
brand string (query)	Filter results by brand <input type="text" value="#/phones?brand=samsung"/>
availale {0, 1} (query)	Filter results by availability <input type="text" value="#/phones?availale=1"/>
minprice integer (query)	Get results greater than a number <input type="text" value="#/phones?minprice=700"/>
maxprice integer (query)	Get results inferior than a number <input type="text" value="#/phones?maxprice=1200"/>
byprice {ASC, DESC} (query)	Try results by price <input type="text" value="#/phones?byprice=DESC"/>
search string (query)	Search a phone by model with a string of characters <input type="text" value="#/phones?search=foo"/>
perpage integer (query)	Define the number of element per page - default = 10 <input type="text" value="#/phones?perpage=8"/>
page integer (query)	Define the page - default = 1 <input type="text" value="#/phones?page=2"/>

Responses

Code	Description	Links
200	Returns the list of phones with pagination system  Media type <a href="#">application/json</a> Controls Accept header. Example Value   Schema <pre>{   "id": 112,   "model": "Samsung Galaxy S8",   "price": 854.54,   "color": "dark grey",   "availability": true,   "brand": {     "brand": "Samsung"   } }</pre>	No links
204	When returns an empty result  Media type <a href="#">application/json</a> Example Value   Schema <pre>{   "status": "204: No Content",   "message": "Aucun resultat pour cette requête." }</pre>	No links
400	When existing a problem in a request  Media type <a href="#">application/json</a>	No links

Example Value | Schema

```
{
  "status": "400: Bad Request",
  "message": "{ message for invalid request }"
}
```

GET /phones/{id}



#### Parameters

Try it out

Name	Description
------	-------------

**id** required  
integer  
(path)

The id of one phone

#/phones/123

#### Responses

Code	Description	Links
------	-------------	-------

200	Returns details of one phone	No links
-----	------------------------------	----------

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{
  "id": 112,
  "model": "Samsung Galaxy S9",
  "catchphrase": "Qui aut asse sit dolores.",
  "description": "Perspiciatis velit aut quia consequatur similique recusandae. Dicta voluptatem est aut recusandae quaerat quibusdam.",
  "price": 854.54,
  "color": "dark grey",
  "size": "144 mm, 76 mm, 8 mm",
  "batteryPower": "4241 mAh",
  "osName": "PhoneOS",
  "weight": "198 g",
  "memory": 28,
  "availability": true,
  "brand": {
    "brand": "Samsung"
  },
  "photos": [
    {
      "name": "public/photos/Huawei ad-tenetur_back.jpg"
    },
    {
      "name": "public/photos/Huawei ad-tenetur_micro.jpg"
    }
  ]
}
```

400	When existing a problem in a request	No links
-----	--------------------------------------	----------

Media type

application/json

Example Value | Schema

```
{
  "status": "400: Bad Request",
  "message": "{ message for invalid request }"
}
```

404	When {id} doesn't exist	No links
-----	-------------------------	----------

Media type

application/json

Example Value | Schema

```
{
  "status": "404: Not Found",
  "message": "Cette ressource n'existe pas"
}
```

## users



GET /users



#### Parameters

Try it out

Name	Description
------	-------------

**byusername**  
{ASC, DESC}  
(query)  
Try results by username of users in alphabetical order

#/users?byusername=DESC

**id**  
{ASC, DESC}  
(query)  
Try results by the id of users

#/users?byid=DESC

**search**  
string  
(query)  
Search a user by username with a string of characters

#/users?search=foo

**perpage**  
integer  
(query)  
Define the number of element per page - default = 10

#/users?perpage=8

**page**  
integer  
(query)  
Define the page - default = 1

#/users?page=2

#### Responses

Code	Description	Links
------	-------------	-------

200	Returns the list of users linked to a client with pagination system	No links
<div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value   Schema</div> <div><pre>{  "id": 853,  "username": "johndoe654",  "email": "johndoe654@orange.fr"}</pre></div>		
204	When returns an empty result	No links
<div>Media type</div> <div>application/json</div> <div>Example Value   Schema</div> <div><pre>{  "status": "204: No Content",  "message": "Aucun resultat pour cette requête."}</pre></div>		
400	When existing a problem in a request	No links
<div>Media type</div> <div>application/json</div> <div>Example Value   Schema</div> <div><pre>{  "status": "400: Bad Request",  "message": "{ message for invalid request }"}</pre></div>		

POST

/users

Parameters

Try it out

Name	Description
<div>in_request_body <div><div>required</div></div></div> <div>object</div> <div>(header)</div>	<div>Add a new user in the database - Json format</div> <div><pre>{  "username": "johndoe654",  "email": "johndoe654@orange.fr",  "tel": "9654586987",  "profilPicture": "my_picture.jpg",  "adress": "154, rue des bouleaux 44450 Paris"}</pre></div>

Responses

Code	Description	Links
201	<div>For create a new user linked to a client. Returns Success Message with the just created element</div> <div><div>Media type</div><div>application/json</div><div>Controls Accept header.</div><div>Example Value   Schema</div><div><pre>{  "status": "201: Created",  "message": "Un utilisateur a été ajouté."}</pre></div></div>	<div>No links</div>
400	<div>When the request is not valid</div> <div><div>Media type</div><div>application/json</div><div>Example Value   Schema</div><div><pre>{  "status": "400: Bad Request",  "message": "{ message for invalid request }"}</pre></div></div>	<div>No links</div>

GET

/users/{id}

Parameters

Try it out

Name	Description
<div><div>id</div><div><div>required</div></div><div>integer</div><div>(path)</div></div>	The id of one user
<div>#/users/695</div>	

Responses

Code	Description	Links
200	Returns details of one user linked to a client	No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{
  "id": 853,
  "username": "johndoe654",
  "email": "johndoe654@orange.fr",
  "roles": [
    "ROLE_USER"
  ],
  "dateAtCreated": "2021-01-14T10:27:38.251Z",
  "tel": "0645785859",
  "profilPicture": "my_picture.jpg",
  "adress": "154, rue des bouleaux 44450 Paris"
}
```

400

When existing a problem in a request

No links

Media type

application/json

Example Value | Schema

```
{
  "status": "400: Bad Request",
  "message": "{ message for invalid request }"
}
```

404

When {id} dosen't exist

No links

Media type

application/json

Example Value | Schema

```
{
  "status": "404: Not Found",
  "message": "Cette ressource n'existe pas"
}
```

PUT

/users/{id}

Parameters

Try it out

Name	Description
<b>in_request_body</b> <span>required</span> object (header)	Add properties for replace or update a user - Json format - Warning : For PUT method, the body require all the properties of the User. For PATCH method, it's not important. In other words, for replace a user, use PUTCH method and for update a user, use PATCH method. <div><pre>{   "username": "johndoe654",   "email": "johndoe654@orange.fr",   "tel": "064586987",   "profilPicture": "my_picture.jpg",   "adress": "154, rue des bouleaux 44450 Paris" }</pre></div>
<b>id</b> <span>required</span> integer (path)	The id of one user <div>#/users/695</div>

Responses

Code	Description	Links
200	For update a user linked to a client. Returns Success Message with the just updated element <div><div>Media type</div><div>application/json</div><div>Controls Accept header.</div><div>Example Value   Schema</div><div><pre>{   "status": "200: Success",   "message": "L'utilisateur {user_id} a été modifié." }</pre></div></div>	No links
400	When the request is not valid <div><div>Media type</div><div>application/json</div><div>Example Value   Schema</div><div><pre>{   "status": "400: Bad Request",   "message": "{ message for invalid request }" }</pre></div></div>	No links
404	When {id} dosen't exist <div><div>Media type</div><div>application/json</div><div>Example Value   Schema</div><div><pre>{   "status": "404: Not Found",   "message": "Cette ressource n'existe pas" }</pre></div></div>	No links

DELETE

/users/{id}

Parameters

Try it out

Name	Description
<b>id</b> <span>required</span>	

integer (path)	The id of one user	
	<input type="text" value="#/users/695"/>	
Responses		
Code	Description	Links
200	For remove a user linked to a client. Returns Success Message.  Media type <input type="text" value="application/json"/> Controls Accept header. Example Value   Schema <pre>{   "status": "200: Success",   "message": "L'utilisateur {user_id} a été supprimé." }</pre>	No links
400	When the request is not valid  Media type <input type="text" value="application/json"/> Example Value   Schema <pre>{   "status": "400: Bad Request",   "message": "{ message for invalid request }" }</pre>	No links
404	When {id} doesn't exist  Media type <input type="text" value="application/json"/> Example Value   Schema <pre>{   "status": "404: Not Found",   "message": "Cette ressource n'existe pas" }</pre>	No links

PATCH	/users/{id}	
Parameters		
Try it out		
Name	Description	
in_request_body * required object (header)	Add properties for replace or update a user - Json format - Warning : For PUT method, the body require all the properties of the User. For PATCH method, it's not important. In other words, for replace a user, use PUTCH method and for update a user, use PATCH method. <pre>{   "username": "johndoe654",   "email": "johndoe54@orange.fr",   "tel": "065456997",   "profilPicture": "my_picture.jpg",   "adress": "154, rue des bouleaux 44450 Paris" }</pre>	
id * required integer (path)	The id of one user	
	<input type="text" value="#/users/695"/>	
Responses		
Code	Description	Links
200	For update a user linked to a client. Returns Success Message with the just updated element  Media type <input type="text" value="application/json"/> Controls Accept header. Example Value   Schema <pre>{   "status": "200: Success",   "message": "L'utilisateur {user_id} a été modifié." }</pre>	No links
400	When the request is not valid  Media type <input type="text" value="application/json"/> Example Value   Schema <pre>{   "status": "400: Bad Request",   "message": "{ message for invalid request }" }</pre>	No links
404	When {id} doesn't exist  Media type <input type="text" value="application/json"/> Example Value   Schema <pre>{   "status": "404: Not Found",   "message": "Cette ressource n'existe pas" }</pre>	No links

```
status: 404: not found ,
"message": "Cette ressource n'existe pas"
}
```

## Schemas

```
UsersDetails {
  id          integer
  username*   string
              maxLength: 0
              minLength: 3
  email       string
  roles       > [...]
  dateAtCreated string($date-time)
              nullable: true
  tel         string
              nullable: true
  profilPicture string
              nullable: true
  adress      string
              nullable: true
}
```

```
UsersList {
  id          integer
  username*   string
              maxLength: 0
              minLength: 3
  email       string
}
```

```
PhoneDetails {
  id          integer
  model       string
  catchPhrase string
              nullable: true
  description  string
  price       number($float)
  color       string
              nullable: true
  size        string
              nullable: true
  batteryPower string
              nullable: true
  osName      string
              nullable: true
  weight      string
              nullable: true
  memory      integer
              nullable: true
  availability boolean
              nullable: true
  brand       > {...}
  photos      > [...]
}
```

```
PhonesList {
  id          integer
  model       string
  price       number($float)
  color       string
              nullable: true
  size        string
              nullable: true
  availability boolean
              nullable: true
  brand       > {...}
}
```

```
PhoneBrand {
  brand string
}
```

```
Photo {
  name string
}
```

```
PhoneBrand2 {
  brand string
}
```