# 

**A R T E M I S**

Documento de endpoints

# 

# Index

[**Index**](#_g5amffuf5vrf) **1**

[API: User](#_gco3v1i0vjo6) **7**

[**Register**](#_kwb3l1ukgmro) **7**

[Entrada](#_op9jeqz434h7) 7

[Recurso URI](#_6lr9kp9xr9) 7

[Cuerpo](#_vtqk6ajb1obi) 7

[Salida](#_jm13thmqdmum) 8

[Respuesta de petición](#_uyt0r6k5i56z) 8

[Campos de respuesta](#_y4maibdk71kv) 8

[Códigos HTTP de estado](#_jtke3jsrd9du) 8

[Ejemplos](#_576wl2r2h0do) 8

[Ejemplo 1](#_nt77execwhbm) 8

[Entrada](#_wos8nssdo2jc) 8

[Salida](#_7m0x3vdp7dni) 9

[**Login**](#_lcszqbfznxcf) **9**

[Entrada](#_q3mjtnuoq5r4) 9

[Recurso URI](#_89don6ngxspp) 9

[Cuerpo](#_grd6yupeds3f) 9

[Salida](#_pmkp11663pbj) 10

[Respuesta de petición](#_qy75e6mwqwq7) 10

[Campos de respuesta](#_ufiveqq1kyia) 10

[Códigos HTTP de estado](#_8qxpncfiwa0x) 10

[Ejemplos](#_bdqbiy55xkap) 11

[Ejemplo 1](#_f9v1d49chvdw) 11

[Entrada](#_ratb5dvkyu98) 11

[Salida](#_ven0yruluz8h) 11

[**Google Login**](#_2n2mdtrc1nc0) **11**

[Entrada](#_6jgix4q69q2u) 11

[Recurso URI](#_5j1w4sgv8rnu) 12

[Cuerpo](#_bwqvdfg4e6yc) 12

[Salida](#_hb3htfwfeayj) 12

[Respuesta de petición](#_3wj7czuj8erz) 12

[Campos de respuesta](#_9in1e96rm12p) 12

[Códigos HTTP de estado](#_l7ttq1pt5c97) 12

[Ejemplos](#_c3csd9l89voe) 13

[Ejemplo 1](#_r2izx4z20woo) 13

[Entrada](#_w22yrx2e0bto) 13

[Salida](#_e3xdfggxjhll) 13

[**Logout**](#_jqspjrmn59hi) **14**

[Entrada](#_acc3qhd1ytwd) 14

[Recurso URI](#_91fjswkm6qc6) 14

[Cuerpo](#_5laug8c3iegm) 14

[Salida](#_j0piin4lrv9y) 14

[Respuesta de petición](#_9aibpbcwcg6m) 14

[Campos de respuesta](#_orra49nvfus1) 14

[Códigos HTTP de estado](#_evnruv35vd76) 15

[Ejemplos](#_djp47fxcwgbo) 15

[Ejemplo 1](#_rv1q18bgmazx) 15

[Entrada](#_wrtelfv3r95l) 15

[Salida](#_gcd0z0esxjga) 15

[GetUser](#_9ns3tw80224r) 15

[Entrada](#_ya2bjw35xkrv) 15

[Recurso URI](#_k3vi43nzx7zn) 16

[Cuerpo](#_acffnjmtaff5) 16

[Salida](#_ql19favkvrg5) 16

[Respuesta de petición](#_ussnlvz014wm) 16

[Campos de respuesta](#_iq4v86xur2c8) 16

[Códigos HTTP de estado](#_invks3q1nwv2) 17

[Ejemplo](#_cslcffst4oyo) 17

[Ejemplo 1](#_9o7r73pg0a3v) 17

[Entrada](#_gyewnfksbkzk) 17

[Salida](#_emohgeqazspb) 18

[GetUsers](#_shlftuw3m1gv) 18

[Entrada](#_xvmbpmew2je) 18

[Recurso URI](#_cbr2r9eu8lse) 18

[Parámetros URI](#_cpmv3nf6a1cv) 18

[Cuerpo](#_rad6sjy0e0it) 19

[Salida](#_cqpqx3ht6f7) 19

[Respuesta de petición](#_hf1rsaltsoma) 19

[Campos de respuesta](#_j67xqg2qu7ah) 20

[Códigos HTTP de estado](#_wr7nyuc4g6b8) 21

[Ejemplo 1](#_gcd6yfbz5nfm) 21

[Entrada](#_v1iz7aewnexg) 21

[Salida](#_6sgmhsfb7e) 21

[SearchUsers](#_spnbhyn9lkzc) 22

[Entrada](#_dt88qoae8vnb) 22

[Recurso URI](#_wvdf9r54g8jf) 23

[Parámetros URI](#_qwh8like6aaq) 23

[Cuerpo](#_xfaoywfqzbmv) 23

[Salida](#_w8y46an09dqe) 23

[Respuesta de petición](#_o5whxp6eozvg) 23

[Campos de respuesta](#_zc7tzvx3t95d) 24

[Códigos HTTP de estado](#_mo50h2917t7m) 25

[Ejemplo 1](#_hzqveeikj882) 25

[Entrada](#_fe8b342siz9r) 25

[Salida](#_mc6w7lcbwl5p) 25

[PutUser](#_dexmaijcspb) 25

[Entrada](#_2bqufkg6idgb) 25

[Entrada](#_2cx6t65e0vf0) 26

[Recurso URI](#_3hrugd3kp4qm) 26

[Cuerpo](#_uon6afvi2liu) 26

[Salida](#_t7o9akx77se7) 27

[Respuesta de petición](#_m77c2c4wxtm0) 27

[Campos de respuesta](#_rjyry92ub4df) 27

[Códigos HTTP de estado](#_oo7huyycgmf4) 28

[Ejemplos](#_9yx45k5txe48) 28

[Ejemplo 1](#_578m4hxt1xhu) 28

[Entrada](#_9l54wslufhoo) 28

[Salida](#_z94gye3ro77t) 29

[DeleteUser](#_at8urut90he9) 29

[Entrada](#_5gka9dwh8udx) 29

[Entrada](#_ar4eng9l0du8) 29

[Recurso URI](#_8txpj2rmagl3) 29

[Cuerpo](#_ftj743i6b666) 29

[Salida](#_h6a73272ztu6) 30

[Respuesta de petición](#_drhg82wcflpk) 30

[Campos de respuesta](#_5eznc4coozv0) 30

[Códigos HTTP de estado](#_xq8892j41a9s) 30

[Ejemplos](#_u1vg3z2wn65c) 30

[Ejemplo 1](#_6en4x7vl1u4n) 30

[Entrada](#_wb7tkc7cpq8y) 30

[Salida](#_8khf234dig8r) 30

[FollowUser](#_pqbwlcwnsbt8) 31

[Entrada](#_cbtk3x5xl5c2) 31

[Recurso URI](#_rd7anklreai4) 31

[Parámetros URI](#_kexcio5y7zop) 31

[Cuerpo](#_2cmfn2kied2q) 31

[Salida](#_ufiv55lvyg3y) 32

[Respuesta de petición](#_tpmd93f4m7lr) 32

[Campos de respuesta](#_p9pu1usy6tr9) 32

[Códigos HTTP de estado](#_s5x7sabeeutu) 32

[Ejemplo 1](#_9njxzvfg0bdt) 32

[Entrada](#_1hzcivte0227) 32

[Salida](#_dx9neif2uq6c) 32

[**API: Folder**](#_9tiq2jmbjr3p) **33**

[**CreateFolder**](#_awv2h6lpeinx) **34**

[Entrada](#_klnwrakg4v7t) 34

[Recurso URI](#_yp8kqasplubm) 34

[Cuerpo](#_6ro1im1coj25) 34

[Salida](#_dfn2af9gpnkq) 34

[Respuesta de petición](#_4ex77n4b9xd2) 35

[Campos de respuesta](#_vq8qujorr9tj) 35

[Códigos HTTP de estado](#_1zt4nmg1q6ew) 35

[Ejemplos](#_2dep25pemk75) 35

[Ejemplo 1](#_73n5olkjdxxc) 35

[Entrada](#_pguc5ny0g2p4) 35

[Salida](#_6752ofjlfopw) 36

[**GetFolder**](#_r8hywj9v52) **36**

[Entrada](#_keh9l9389bv8) 36

[Recurso URI](#_n3m0e2l14jqr) 36

[Parámetros URI](#_83s5nsyvdwlz) 36

[Salida](#_bs3wukw8i4wr) 37

[Respuesta de petición](#_jocdplhnn2kv) 37

[Campos de respuesta](#_pg03dp5bcxmj) 37

[Códigos HTTP de estado](#_yosrozidgw3x) 37

[Ejemplos](#_jbhkfhcup37s) 38

[Ejemplo 1](#_khs36k7zh5ib) 38

[Entrada](#_74i94x13pbkq) 38

[Salida](#_ntkekbzbwis8) 38

[**GetFolders**](#_b7vykd6n8vsk) **38**

[Entrada](#_sedvjfvi14lu) 39

[Recurso URI](#_vm7w1bb223q) 39

[Salida](#_lwfzubgvygxm) 39

[Respuesta de petición](#_bikvzjct0jxj) 39

[Campos de respuesta](#_geb19drask1r) 39

[Códigos HTTP de estado](#_tkmrzz8xrr9e) 40

[Ejemplos](#_80af3ocoygfn) 40

[Ejemplo 1](#_9e3b09bpmdwd) 40

[Entrada](#_pmgzjrqdq30j) 40

[Salida](#_3srarlc7c03k) 40

[**PutFolder**](#_uzg1mn3pzgvr) **41**

[Entrada](#_2yyaa0mbh5fl) 41

[Recurso URI](#_c643p6cggzpe) 41

[Parámetros URI](#_7pq2chyrv94s) 41

[Cuerpo](#_vo42rlg1riyp) 41

[Salida](#_8ntpfiwx007s) 42

[Respuesta de petición](#_kdh5p5pe2jzc) 42

[Campos de respuesta](#_h2hekmfcurbi) 42

[Códigos HTTP de estado](#_lu9scwtgzg3t) 42

[Ejemplos](#_u4zvt4z2qhen) 42

[Ejemplo 1](#_k60zccr5nt1e) 42

[Entrada](#_4573fbkjnehy) 43

[Salida](#_9yybw1et4ke6) 43

[**DeleteFolder**](#_o5lzsrz7lrj8) **43**

[Entrada](#_oqgq0p1w8rzy) 43

[Recurso URI](#_sk2m96nzc3h0) 43

[Parámetros URI](#_5rmelwwemvjn) 44

[Salida](#_ucvdbhx4txpl) 44

[Respuesta de petición](#_cs71tkfbbxsv) 44

[Campos de respuesta](#_w1wbd9ajvpkr) 44

[Códigos HTTP de estado](#_jt1sal5nw5xy) 44

[Ejemplos](#_zcycqohz3r1l) 45

[Ejemplo 1](#_zcywnt17956x) 45

[Entrada](#_wisnyckxfu6k) 45

[Salida](#_izpaaivz6ven) 45

[**API: Work**](#_qncxwxsme28o) **45**

[CreateWork](#_m7btwarxzped) 45

[Entrada](#_a47axxbk44yx) 46

[Recurso URI](#_oaw0xcmhh12g) 46

[Parámetros URI](#_xnnu76fz6akj) 46

[Cuerpo](#_jaxp5lxf7fzc) 46

[Salida](#_pz6o8wbbwdbq) 47

[Respuesta de petición](#_nerar5gigbqw) 48

[Campos de respuesta](#_qr1bmj4toz1p) 48

[Códigos HTTP de estado](#_lco6f3470jtp) 48

[Ejemplos](#_q1pkjcetjnsn) 48

[Ejemplo 1](#_ju8o6229mwb1) 48

[Entrada](#_lg385duqgvqn) 48

[Salida](#_m3uacyhron3q) 49

[GetWork](#_knsiw1oc7fx6) 49

[Entrada](#_k2dh2nu5nglu) 49

[Recurso URI](#_n3tasrp8l5vi) 49

[Parámetros URI](#_d4z5dfmnry89) 49

[Salida](#_ikkgak1ccnq1) 50

[Respuesta de petición](#_wzp1qmuam59) 50

[Campos de respuesta](#_20u7qgx26vzo) 50

[Códigos HTTP de estado](#_q7spcowinok) 50

[Ejemplos](#_rh83g86246am) 51

[Ejemplo 1](#_2j9n97an72h0) 51

[Entrada](#_purv1cx6rk1c) 51

[Salida](#_eswm6vdjey9r) 51

# 

# API: User

## Register

|  |  |
| --- | --- |
| POST | /user |

Este método crea un usuario en base de datos.

### Entrada

#### Recurso URI

|  |  |
| --- | --- |
| POST | https://artemis-back.herokuapp.com/user |

#### Cuerpo

|  |  |  |  |
| --- | --- | --- | --- |
| Parámetro | Tipo | Descripción | Reglas de petición |
| artistic\_name | String | Este parámetro representa el nombre artístico de un usuario de Artemis. | Requerido |
| nick\_name | String | Este parámetro representa el nickname de un usuario de Artemis. | Requerido |
| password | String | Este parámetro representa la contraseña de un usuario de Artemis, la cual es encriptada luego. | Requerido |
| email | String | Este parámetro representa el correo electrónico de un usuario de Artemis. | Requerido |

### Salida

#### Respuesta de petición

|  |
| --- |
| {  "ok": true,  "msg": "Usuario Creado con éxito", } |

#### Campos de respuesta

|  |  |  |
| --- | --- | --- |
| Parámetro | Tipo | Descripción |
| ok | Bool | Representa el éxito de la operación. |
| msg | String | Mensaje de respuesta de la operación. |

#### Códigos HTTP de estado

|  |  |
| --- | --- |
| Estado | Significado |
| 200 | Éxito |
| 400 | Mala petición |
| 401 | Autorización no válida |
| 500 | Error interno del servidor |

#### Ejemplos

##### Ejemplo 1

Registrar a un usuario.

###### Entrada

Enviamos la siguiente información por POST

|  |  |
| --- | --- |
| POST | https://artemis-back.herokuapp.com/user |

Junto con el siguiente body:

|  |
| --- |
| {  "artistic\_name": "RammyBob-Omb",  "nick\_name": "RamonaFlowers",  "email": "ramonaflowers@gmail.com",   "password": "12345" } |

###### *Salida*

|  |
| --- |
| {  "ok": true,  "msg": "Usuario creado con éxito", } |

## 

## Login

|  |  |
| --- | --- |
| POST | /users/login |

Este método permite a un usuario iniciar sesión. Si un usuario está logueado al momento de realizar el inicio de sesión entonces es deslogueado, la cookie es destruida, y se procede al inicio de sesión del usuario actual.

### Entrada

#### Recurso URI

|  |  |
| --- | --- |
| POST | https://artemis-back.herokuapp.com/user/login |

#### Cuerpo

|  |  |  |  |
| --- | --- | --- | --- |
| Parámetro | Tipo | Descripción | Reglas de petición |
| email | String | Este parámetro representa el correo electrónico de un usuario de Artemis. | Requerido |
| password | String | Este parámetro representa la contraseña de un usuario de Artemis, la cual es encriptada luego. | Requerido |

### Salida

#### Respuesta de petición

|  |
| --- |
| {  "ok": true,  "msg": "Usuario logueado con éxito", } |

Asimismo, se genera una cookie la cual contiene un jsonwebtoken que almacena toda la información del usuario.

#### Campos de respuesta

|  |  |  |
| --- | --- | --- |
| Parámetro | Tipo | Descripción |
| ok | Bool | Representa el éxito de la operación. |
| msg | String | Mensaje de respuesta de la operación. |

#### Códigos HTTP de estado

|  |  |
| --- | --- |
| Estado | Significado |
| 200 | Éxito |
| 400 | Mala petición |
| 401 | Autorización no válida |
| 500 | Error interno del servidor |

#### 

#### Ejemplos

##### Ejemplo 1

Iniciar sesión con un usuario

###### Entrada

Enviamos la siguiente información por POST

|  |  |
| --- | --- |
| POST | https://artemis-back.herokuapp.com/user/login |

Junto con el siguiente body:

|  |
| --- |
| {  "email": "ramonaflowers@gmail.com",   "password": "12345" } |

###### Salida

|  |
| --- |
| {  "ok": true,  "msg": "Usuario logueado con éxito" } |

Asimismo, se crea una cookie que permite autenticar al usuario durante la sesión.

## Google Login

|  |  |
| --- | --- |
| POST | /user/login/google |

Este método permite a un usuario iniciar sesión. Si un usuario está logueado al momento de realizar el inicio de sesión entonces es deslogueado, la cookie es destruida, y se procede al inicio de sesión del usuario actual.

### Entrada

#### Recurso URI

|  |  |
| --- | --- |
| POST | https://artemis-back.herokuapp.com/user/login/google |

#### Cuerpo

|  |  |  |  |
| --- | --- | --- | --- |
| Parámetro | Tipo | Descripción | Reglas de petición |
| idtoken | String | Este parámetro representa el token obtenido del login de Google | Requerido |

### Salida

#### Respuesta de petición

|  |
| --- |
| {  "ok": true,  "msg": "Usuario logueado con éxito", } |

Asimismo, se genera una cookie la cual contiene un jsonwebtoken que almacena toda la información del usuario.

#### Campos de respuesta

|  |  |  |
| --- | --- | --- |
| Parámetro | Tipo | Descripción |
| ok | Bool | Representa el éxito de la operación. |
| msg | String | Mensaje de respuesta de la operación. |

#### Códigos HTTP de estado

|  |  |
| --- | --- |
| Estado | Significado |
| 200 | Éxito |
| 400 | Mala petición |
| 401 | Autorización no válida |
| 500 | Error interno del servidor |

#### Ejemplos

##### Ejemplo 1

Iniciar sesión con un usuario con Google

###### Entrada

Enviamos la siguiente información por POST

|  |  |
| --- | --- |
| POST | https://artemis-back.herokuapp.com/user/login/google |

Junto con el siguiente body:

|  |
| --- |
| { "idtoken":"eyJhbGciOiJSUzI1NiIsImtpZCI6IjI1MmZjYjk3ZGY1YjZiNGY2ZDFhODg1ZjFlNjNkYzRhOWNkMjMwYzUiLCJ0eXAiOiJKV1QifQ..b56OA21vc3ugiFVtSvWtHYQ4jWY8aCninvr\_vjFFXj840Pl9nYPSSjByPzudHWVK5SDuePLy6jvIh6eJkk37Eu\_m6a3C1q7t3LMT63tpDD1ZvzjYJ6ktLNa9BxrZixgeABB357JavbpDfnacVh3n0qlWkwC3pwcrRnAz8q\_WtDgNrHLjFs26q0qKUJckxW8v8h4BoOiPT28u4mm3xcriz3hGllWwRmX4p9ZzUh4AS3A4jiBZSkR5xpu6OhOqO3G5MipaHtNc3\_3wlnsnZmqu-RcQfXVYZncTT4L6zjhVtd94IFoQUv7XvFhjCBpPnlYMI92\_F0akISQRVa9zOFpkXA"  } |

###### Salida

|  |
| --- |
| {  "ok": true,  "msg": "Usuario logueado con éxito" } |

Asimismo, se crea una cookie que permite autenticar al usuario durante la sesión.

## Logout

|  |  |
| --- | --- |
| GET | /user/logout |

Este método permite a un usuario cerrar sesión. Antes se verifica que el usuario esté logueado y el token de la cookie sea correcto.

### Entrada

#### Recurso URI

|  |  |
| --- | --- |
| GET | https://artemis-back.herokuapp.com/user/logout |

#### Cuerpo

No hay cuerpo

### Salida

#### Respuesta de petición

|  |
| --- |
| {  "ok": true,  "msg": "Cierre sesión exitoso", } |

Asimismo, se destruye la cookie.

#### Campos de respuesta

|  |  |  |
| --- | --- | --- |
| Parámetro | Tipo | Descripción |
| ok | Bool | Representa el éxito de la operación. |
| msg | String | Mensaje de respuesta de la operación. |

#### Códigos HTTP de estado

|  |  |
| --- | --- |
| Estado | Significado |
| 200 | Éxito |
| 400 | Mala petición |
| 401 | Autorización no válida |
| 500 | Error interno del servidor |

#### Ejemplos

##### Ejemplo 1

Cerrar sesión con un usuario

###### Entrada

Enviamos la siguiente información por GET

|  |  |
| --- | --- |
| GET | https://artemis-back.herokuapp.com/user/logout |

###### Salida

|  |
| --- |
| {  "ok": true,  "msg": "Cierre sesión exitoso" } |

Asimismo, se destruye la cookie.

## GetUser

### Entrada

|  |  |
| --- | --- |
| GET | /user |

Este método permite obtener la información del usuario logueado en formato JSON. Requiere que la cookie de usuario logueado esté puesta, de lo contrario el método no será accedido.

#### Recurso URI

|  |  |
| --- | --- |
| GET | https://artemis-back.herokuapp.com/user |

#### Cuerpo

No hay cuerpo

### Salida

#### Respuesta de petición

|  |
| --- |
| {  "followers": [],  "folders": [],  "signed\_google": false,  "signed\_facebook": false,  "\_id": "5ff66fb5cf4b720860729842",  "artistic\_name": "MikoChan",  "nick\_name": "IinoMiko",  "email": "iinomiko@gmail.com",  “profile\_img”: “url”  "\_\_v": 0 } |

#### Campos de respuesta

|  |  |  |
| --- | --- | --- |
| Parámetro | Tipo | Descripción |
| followers | Array | Array dónde están los ObjectId de los usuarios que siguen al usuario logueado. |
| folders | Array | Array dónde están los ObjectId de los folders que posee el usuario logueado. |
| signed\_google | Boolean | Valor booleano que indica si el usuario se ha registrado usando google. |
| signed\_facebook | Boolean | Valor booleano que indica si el usuario se ha registrado usando facebook. |
| \_id | ObjectId | Id que identifica al usuario en la base de datos. |
| artistic\_name | String | String que contiene el nombre artístico del usuario logueado. |
| nick\_name | String | String que contiene el nickname de usuario logueado. |
| email | String | String que contiene el email del usuario logueado. |
| profile\_img | String | String que contiene el url donde está almacenada la imagen de perfil del usuario |

#### Códigos HTTP de estado

|  |  |
| --- | --- |
| Estado | Significado |
| 200 | Éxito |
| 400 | Mala petición |
| 401 | Autorización no válida |
| 500 | Error interno del servidor |

### Ejemplo

#### Ejemplo 1

Obtener la información del usuario logueado

###### Entrada

Enviamos la siguiente información por GET

|  |  |
| --- | --- |
| GET | https://artemis-back.herokuapp.com/user |

###### Salida

|  |
| --- |
| {  "followers": [],  "folders": [],  "signed\_google": false,  "signed\_facebook": false,  "\_id": "5ff66fb5cf4b720860729842",  "artistic\_name": "MikoChan",  "nick\_name": "IinoMiko",  "email": "iinomiko@gmail.com",  "\_\_v": 0 } |

## GetUsers

### Entrada

|  |  |
| --- | --- |
| GET | /users?from={from}&limit={limit} |

Este método permite obtener la información de varios usuarios en formato JSON de acuerdo a dos parámetros: from, desde donde se va a buscar y que por defecto es 0, y limit, número de resultados y que por defecto es 0 para obtener todos los documentos de base de datos. Requiere que la cookie de usuario logueado esté puesta, de lo contrario el método no será accedido.

#### Recurso URI

|  |  |
| --- | --- |
| GET | https://artemis-back.herokuapp.com/users?from={from}&limit={limit} |

#### 

#### Parámetros URI

|  |  |  |  |
| --- | --- | --- | --- |
| Parámetro | Tipo | Descripción | Reglas de petición |
| from | integer | Este parámetro representa desde que elemento de la base de datos se va a empezar a entregar documentos | Opcional |
| limit | integer | Este parámetro representa el número de documentos a enviar por la base de datos | Opcional |

### 

#### Cuerpo

No hay cuerpo

### Salida

#### Respuesta de petición

|  |
| --- |
| [  {  "followers": [],  "folders": [],  "signed\_google": false,  "signed\_facebook": false,  "\_id": "5ff519b2a89695001717c43b",  "artistic\_name": "John Salsichón",  "nick\_name": "warevil",  “profile\_img”: “url”,  "email": "tubatheriafina@hotmail.com",  "\_\_v": 0  },  {  "followers": [],  "folders": [],  "signed\_google": false,  "signed\_facebook": false,  "\_id": "5ff541d63862e30017132cde",  "artistic\_name": "brasilball",  "nick\_name": "Brasil Ball",  “profile\_img”: “url”,  "email": "huehuehue@mail.com",  "\_\_v": 0  },  {  "followers": [],  "folders": [],  "signed\_google": false,  "signed\_facebook": false,  "\_id": "5ff66fb5cf4b720860729842",  "artistic\_name": "MikoChan",  "nick\_name": "IinoMiko",  “profile\_img”: “url”,  "email": "iinomiko@gmail.com",  "\_\_v": 0  } ] |

#### Campos de respuesta

La respuesta es un Array que contendrá los datos de todos los usuarios solicitados

|  |  |  |
| --- | --- | --- |
| Parámetro | Tipo | Descripción |
| followers | Array | Array dónde están los ObjectId de los usuarios que siguen al usuario logueado. |
| folders | Array | Array dónde están los ObjectId de los folders que posee el usuario logueado. |
| signed\_google | Boolean | Valor booleano que indica si el usuario se ha registrado usando google. |
| signed\_facebook | Boolean | Valor booleano que indica si el usuario se ha registrado usando facebook. |
| \_id | ObjectId | Id que identifica al usuario en la base de datos. |
| artistic\_name | String | String que contiene el nombre artístico del usuario logueado. |
| nick\_name | String | String que contiene el nickname de usuario logueado. |
| email | String | String que contiene el email del usuario logueado. |
| profile\_img | String | String que contiene el url donde está almacenada la imagen de perfil del usuario |

#### Códigos HTTP de estado

|  |  |
| --- | --- |
| Estado | Significado |
| 200 | Éxito |
| 400 | Mala petición |
| 401 | Autorización no válida |
| 500 | Error interno del servidor |

##### Ejemplo 1

Obtener todos

###### Entrada

Enviamos la siguiente información por GET

|  |  |
| --- | --- |
| GET | https://artemis-back.herokuapp.com/users?from=14&limit=3 |

###### Salida

|  |
| --- |
| [  {  "followers": [],  "folders": [],  "signed\_google": false,  "signed\_facebook": false,  "\_id": "5ff519b2a89695001717c43b",  "artistic\_name": "John Salsichón",  "nick\_name": "warevil",  “profile\_img”: “url”,  "email": "tubatheriafina@hotmail.com",  "\_\_v": 0  },  {  "followers": [],  "folders": [],  "signed\_google": false,  "signed\_facebook": false,  "\_id": "5ff541d63862e30017132cde",  "artistic\_name": "brasilball",  "nick\_name": "Brasil Ball",  “profile\_img”: “url”,  "email": "huehuehue@mail.com",  "\_\_v": 0  },  {  "followers": [],  "folders": [],  "signed\_google": false,  "signed\_facebook": false,  "\_id": "5ff66fb5cf4b720860729842",  "artistic\_name": "MikoChan",  "nick\_name": "IinoMiko",  “profile\_img”: “url”,  "email": "iinomiko@gmail.com",  "\_\_v": 0  } ] |

## SearchUsers

### Entrada

|  |  |
| --- | --- |
| GET | /users/search?{term} |

Este método permite obtener información de todos los usuarios cuyo nick\_name sea parecido al enviado por la búsqueda.

#### Recurso URI

|  |  |
| --- | --- |
| GET | https://artemis-back.herokuapp.com/users/search/{term} |

#### 

#### Parámetros URI

|  |  |  |  |
| --- | --- | --- | --- |
| Parámetro | Tipo | Descripción | Reglas de petición |
| term | string | Este parámetro representa una expresión regular utilizada para buscar en la base de datos por el nick\_name | Requerido |

#### Cuerpo

No hay cuerpo

### Salida

#### Respuesta de petición

###### 

|  |
| --- |
| [  {  "followers": [],  "folders": [],  "signed\_google": false,  "signed\_facebook": false,  "\_id": "5ff4e2e579acf009584fa0cd",  "artistic\_name": "LangleyShino",  "nick\_name": "AsukaTorres",  "email": "asutorres@gmail.com",  "\_\_v": 0  },  {  "followers": [],  "folders": [],  "signed\_google": true,  "signed\_facebook": false,  "\_id": "5ff278dd15800a330475076a",  "artistic\_name": "Leonardo Torres Briones",  "nick\_name": "Leonardo Torres Briones",  "email": "dleonardotb2001@gmail.com",  "\_\_v": 0  } ] |

#### Campos de respuesta

La respuesta es un Array que contendrá los datos de todos los usuarios encontrados

|  |  |  |
| --- | --- | --- |
| Parámetro | Tipo | Descripción |
| followers | Array | Array dónde están los ObjectId de los usuarios que siguen al usuario logueado. |
| folders | Array | Array dónde están los ObjectId de los folders que posee el usuario logueado. |
| signed\_google | Boolean | Valor booleano que indica si el usuario se ha registrado usando google. |
| signed\_facebook | Boolean | Valor booleano que indica si el usuario se ha registrado usando facebook. |
| \_id | ObjectId | Id que identifica al usuario en la base de datos. |
| artistic\_name | String | String que contiene el nombre artístico del usuario logueado. |
| nick\_name | String | String que contiene el nickname de usuario logueado. |
| email | String | String que contiene el email del usuario logueado. |
| profile\_img | String | String que contiene el url donde está almacenada la imagen de perfil del usuario |

#### Códigos HTTP de estado

|  |  |
| --- | --- |
| Estado | Significado |
| 200 | Éxito |
| 400 | Mala petición |
| 401 | Autorización no válida |
| 500 | Error interno del servidor |

##### Ejemplo 1

Obtener todos

###### Entrada

Enviamos la siguiente información por GET

|  |  |
| --- | --- |
| GET | https://artemis-back.herokuapp.com/users/search/Torres |

###### Salida

|  |
| --- |
| [  {  "followers": [],  "folders": [],  "signed\_google": false,  "signed\_facebook": false,  "\_id": "5ff4e2e579acf009584fa0cd",  "artistic\_name": "LangleyShino",  "nick\_name": "AsukaTorres",  "email": "asutorres@gmail.com",  "\_\_v": 0  },  {  "followers": [],  "folders": [],  "signed\_google": true,  "signed\_facebook": false,  "\_id": "5ff278dd15800a330475076a",  "artistic\_name": "Leonardo Torres Briones",  "nick\_name": "Leonardo Torres Briones",  "email": "dleonardotb2001@gmail.com",  "\_\_v": 0  } ] |

## UpdateUser

### Entrada

|  |  |
| --- | --- |
| PUT | /user |

Este método permite actualizar la información del usuario logueado.

### Entrada

#### Recurso URI

|  |  |
| --- | --- |
| PUT | https://artemis-back.herokuapp.com/user |

#### Cuerpo

|  |  |  |  |
| --- | --- | --- | --- |
| Parámetro | Tipo | Descripción | Reglas de petición |
| artistic\_name | String | Este parámetro representa el nombre artístico de un usuario de Artemis. | Requerido |
| nick\_name | String | Este parámetro representa el nickname de un usuario de Artemis. | Requerido |
| password | String | Este parámetro representa la contraseña de un usuario de Artemis, la cual es encriptada luego. | Requerido |
| email | String | Este parámetro representa el correo electrónico de un usuario de Artemis. | Requerido |
| image | File | Este parámetro representa la nueva imagen de perfil del usuario | Requerido |

### 

Asimismo, los parámetros de todas maneras deberán ser únicos en la base de datos.

### Salida

#### Respuesta de petición

|  |
| --- |
| {  "ok": true,  "msg": "Usuario actualizado con éxito" } |

#### Campos de respuesta

|  |  |  |
| --- | --- | --- |
| Parámetro | Tipo | Descripción |
| ok | Bool | Representa el éxito de la operación. |
| msg | String | Mensaje de respuesta de la operación. |

#### Códigos HTTP de estado

|  |  |
| --- | --- |
| Estado | Significado |
| 200 | Éxito |
| 400 | Mala petición |
| 401 | Autorización no válida |
| 500 | Error interno del servidor |

#### Ejemplos

##### Ejemplo 1

###### Entrada

Enviamos la siguiente información por PUT

|  |  |
| --- | --- |
| PUT | https://artemis-back.herokuapp.com/users |

Junto con el siguiente body:

|  |
| --- |
| {  "artistic\_name": "RammyBob-Omb",  "nick\_name": "RamonaFlowers",  "email": "ramonaflowers@gmail.com",   "password": "12345", } |

###### Salida

|  |
| --- |
| {  "ok": true,  "msg": "Usuario actualizado con éxito" } |

## DeleteUser

### Entrada

|  |  |
| --- | --- |
| DELETE | /user |

Este método permite eliminar la información del usuario logueado junto con toda su información en otras colecciones como carpetas, trabajos, comentarios, comentarios en sus obras, estadísticas y versiones de sus trabajos.

### Entrada

#### Recurso URI

|  |  |
| --- | --- |
| DELETE | https://artemis-back.herokuapp.com/user |

#### Cuerpo

No tiene cuerpo

### Salida

#### Respuesta de petición

|  |
| --- |
| {  "ok": true,  "msg": "Usuario borrado exitosamente" } |

#### Campos de respuesta

|  |  |  |
| --- | --- | --- |
| Parámetro | Tipo | Descripción |
| ok | Bool | Representa el éxito de la operación. |
| msg | String | Mensaje de respuesta de la operación. |

#### Códigos HTTP de estado

|  |  |
| --- | --- |
| Estado | Significado |
| 200 | Éxito |
| 400 | Mala petición |
| 401 | Autorización no válida |
| 500 | Error interno del servidor |

#### Ejemplos

##### Ejemplo 1

###### Entrada

Enviamos la siguiente información por DELETE

|  |  |
| --- | --- |
| DELETE | https://artemis-back.herokuapp.com/user |

###### Salida

#### 

|  |
| --- |
| {  "ok": true,  "msg": "Usuario borrado exitosamente" } |

## FollowUser

### Entrada

Enviamos la siguiente información por PUT

|  |  |
| --- | --- |
| PUT | /user/follow/{nick\_name} |

Este método permite seguir a un usuario en la plataforma

#### Recurso URI

|  |  |
| --- | --- |
| PUT | https://artemis-back.herokuapp.com/user/follow/{nick\_name} |

#### 

#### Parámetros URI

|  |  |  |  |
| --- | --- | --- | --- |
| Parámetro | Tipo | Descripción | Reglas de petición |
| nick\_name | string | Este parámetro representa el nick\_name del usuario a seguir | Requerido |

#### Cuerpo

No hay cuerpo

### Salida

#### Respuesta de petición

###### 

|  |
| --- |
| {  "ok": true,  "msg": "Usuario <nick\_name> seguido con éxito" } |

#### Campos de respuesta

La respuesta es un Array que contendrá los datos de todos los usuarios encontrados

|  |  |  |
| --- | --- | --- |
| Parámetro | Tipo | Descripción |
| ok | Boolean | Representa el éxito o fracaso de la consulta |
| msg | String | Mensaje de descripción de la consulta |

#### Códigos HTTP de estado

|  |  |
| --- | --- |
| Estado | Significado |
| 200 | Éxito |
| 400 | Mala petición |
| 401 | Autorización no válida |
| 500 | Error interno del servidor |

##### Ejemplo 1

Seguir al usuario con nick\_name “americaBall”

###### Entrada

Enviamos la siguiente información por PUT

|  |  |
| --- | --- |
| PUT | https://artemis-back.herokuapp.com/user/follow/americaBall |

###### Salida

|  |
| --- |
| {  "ok": true,  "msg": "Usuario americaBall seguido con éxito"  } |

# 

# API: Folder

## CreateFolder

|  |  |
| --- | --- |
| POST | /folder |

Este método crea un folder en base de datos para el usuario logueado.

### Entrada

#### Recurso URI

|  |  |
| --- | --- |
| POST | https://artemis-back.herokuapp.com/folder |

#### Cuerpo

|  |  |  |  |
| --- | --- | --- | --- |
| Parámetro | Tipo | Descripción | Reglas de petición |
| name | string | Nombre del folder a crearse | Requerido |

### 

### Salida

#### Respuesta de petición

|  |
| --- |
| {  "ok": true,  "msg": "Folder creado con éxito" } |

#### Campos de respuesta

|  |  |  |
| --- | --- | --- |
| Contenedor exterior/ campo | Tipo | Descripción |
| ok | bool | Representa el éxito de la operación |
| msg | string | Mensaje de respuesta de la operación |

#### Códigos HTTP de estado

Este endpoint puede devolver uno de los siguientes códigos de estado HTTP

|  |  |
| --- | --- |
| Estado | Significado |
| 200 | Éxito |
| 400 | Mala petición |
| 401 | Autorización no válida |
| 500 | Error interno del servidor |

#### Ejemplos

##### Ejemplo 1

Crear un folder para el usuario logueado.

###### Entrada

Enviamos la siguiente petición **POST**:

|  |  |
| --- | --- |
| POST | /folder |

Junto con el siguiente **Body:**

|  |
| --- |
| {  "name": "Chroniverse Universe" } |

###### 

###### Salida

Obtenemos la siguiente respuesta:

|  |
| --- |
| {  "ok": true,  "msg": "Folder creado con éxito" } |

## GetFolder

|  |  |
| --- | --- |
| GET | /folder?folder\_name={folder\_name} |

Este método obtiene un folder en base al usuario logueado y el parámetro folder\_name.

### Entrada

#### Recurso URI

|  |  |
| --- | --- |
| GET | https://artemis-back.herokuapp.com/folder?folder\_name={folder\_name} |

#### 

#### Parámetros URI

|  |  |  |  |
| --- | --- | --- | --- |
| Parámetro | Tipo | Descripción | Reglas de petición |
| folder\_name | string | Este parámetro representa el nombre del folder a obtenerse. | Requerido |

### Salida

#### Respuesta de petición

|  |
| --- |
| {  "works": [],  "\_id": "5ff24b1692e3ee264cce035f",  "owner": "5fef75da5ee9e733f84b9c9e",  "name": "Chroniverse Universe",  "\_\_v": 0 } |

#### Campos de respuesta

|  |  |  |
| --- | --- | --- |
| Contenedor exterior/ campo | Tipo | Descripción |
| works | array | Contiene las obras del folder obtenido |
| \_id | string | Representa el id en base de datos del folder obtenido |
| owner | string | Representa el id en base de datos del propietario del folder obtenido |
| name | string | Representa el nombre del folder obtenido |
| \_\_v | integer | Representa la versión del documento |

#### Códigos HTTP de estado

Este endpoint puede devolver uno de los siguientes códigos de estado HTTP

|  |  |
| --- | --- |
| Estado | Significado |
| 200 | Éxito |
| 400 | Mala petición |
| 401 | Autorización no válida |
| 500 | Error interno del servidor |

#### Ejemplos

##### Ejemplo 1

Obtener el folder llamado “Chroniverse Universe” del usuario logueado, en este caso con nickname “LoveShadyKing”.

###### Entrada

Enviamos la siguiente petición **GET**:

|  |  |
| --- | --- |
| GET | /folder?folder\_name=Chroniverse Universe |

###### 

###### Salida

Obtenemos la siguiente respuesta:

|  |
| --- |
| {  "works": [],  "\_id": "5ff24b1692e3ee264cce035f",  "owner": "5fef75da5ee9e733f84b9c9e",  "name": "Chroniverse Universe",  "\_\_v": 0 } |

## GetFolders

|  |  |
| --- | --- |
| GET | /folders |

Este método devuelve todos los folders en base de datos para el usuario con el nickname ingresado

### Entrada

#### Recurso URI

|  |  |
| --- | --- |
| GET | https://artemis-back.herokuapp.com/folders |

### Salida

#### Respuesta de petición

|  |
| --- |
| [  {  "works": [],  "\_id": "5ff24b1692e3ee264cce035f",  "owner": "5fef75da5ee9e733f84b9c9e",  "name": "Chroniverse Universe",  "\_\_v": 0  },  {  "works": [],  "\_id": "5ff3795355ef9924904f8037",  "owner": "5fef75da5ee9e733f84b9c9e",  "name": "Anécdotas personales",  "\_\_v": 0  } ] |

#### Campos de respuesta

|  |  |  |
| --- | --- | --- |
| Contenedor exterior/ campo | Tipo | Descripción |
| works | array | Contiene las obras del folder obtenido |
| \_id | string | Representa el id en base de datos del folder obtenido |
| owner | string | Representa el id en base de datos del propietario del folder obtenido |
| name | string | Representa el nombre del folder obtenido |
| \_\_v | integer | Representa la versión del documento |

#### Códigos HTTP de estado

Este endpoint puede devolver uno de los siguientes códigos de estado HTTP

|  |  |
| --- | --- |
| Estado | Significado |
| 200 | Éxito |
| 400 | Mala petición |
| 401 | Autorización no válida |
| 500 | Error interno del servidor |

#### Ejemplos

##### Ejemplo 1

Obtener todos los folders para logueado, en este caso con el nickname “LoveShadyKing”

###### Entrada

Enviamos la siguiente petición **GET**:

|  |  |
| --- | --- |
| GET | /folders |

###### Salida

Obtenemos la siguiente respuesta:

|  |
| --- |
| [  {  "works": [],  "\_id": "5ff24b1692e3ee264cce035f",  "owner": "5fef75da5ee9e733f84b9c9e",  "name": "Chroniverse Universe",  "\_\_v": 0  },  {  "works": [],  "\_id": "5ff3795355ef9924904f8037",  "owner": "5fef75da5ee9e733f84b9c9e",  "name": "Anécdotas personales",  "\_\_v": 0  } ] |

## PutFolder

|  |  |
| --- | --- |
| PUT | /folder?folder\_name={folder\_name} |

Este método modifica un folder en base de datos con el folder\_name ingresado para el usuario logueado

### Entrada

#### Recurso URI

|  |  |
| --- | --- |
| PUT | https://artemis-back.herokuapp.com/folder?&folder\_name={folder\_name} |

#### Parámetros URI

|  |  |  |  |
| --- | --- | --- | --- |
| Parámetro | Tipo | Descripción | Reglas de petición |
| folder\_name | string | Este parámetro representa el nombre del folder a actualizarse | Requerido |

#### Cuerpo

|  |  |  |  |
| --- | --- | --- | --- |
| Parámetro | Tipo | Descripción | Reglas de petición |
| name | string | Nuevo nombre del folder | Requerido |

### Salida

#### Respuesta de petición

|  |
| --- |
| {  "ok": true,  "msg": "Folder actualizado con éxito" } |

#### Campos de respuesta

|  |  |  |
| --- | --- | --- |
| Contenedor exterior/ campo | Tipo | Descripción |
| ok | bool | Representa el éxito de la operación |
| msg | string | Mensaje de respuesta de la operación |

#### Códigos HTTP de estado

Este endpoint puede devolver uno de los siguientes códigos de estado HTTP

|  |  |
| --- | --- |
| Estado | Significado |
| 200 | Éxito |
| 400 | Mala petición |
| 401 | Autorización no válida |
| 500 | Error interno del servidor |

#### Ejemplos

##### Ejemplo 1

Actualizar un folder para el usuario logueado y el folder “Chroniverse Universe”

###### Entrada

Enviamos la siguiente petición **PUT**:

|  |  |
| --- | --- |
| PUT | /folder?folder\_name=Chroniverse Universe |

Junto con el siguiente **Body:**

|  |
| --- |
| {  "name": "Las estrepitosas aventuras de Panchito" } |

###### 

###### Salida

Obtenemos la siguiente respuesta:

|  |
| --- |
| {  "ok": true,  "msg": "Folder actualizado con éxito" } |

## DeleteFolder

|  |  |
| --- | --- |
| DELETE | /folder?folder\_name={folder\_name} |

Este método elimina un folder en base de datos con el folder\_name ingresado para el usuario logueado

### Entrada

#### Recurso URI

|  |  |
| --- | --- |
| DELETE | https://artemis-back.herokuapp.com/folder?folder\_name={folder\_name} |

#### Parámetros URI

|  |  |  |  |
| --- | --- | --- | --- |
| Parámetro | Tipo | Descripción | Reglas de petición |
| folder\_name | string | Este parámetro representa el nombre del folder a eliminarse | Requerido |

### Salida

#### Respuesta de petición

|  |
| --- |
| {  "ok": true,  "msg": "Folder borrado con éxito" } |

#### Campos de respuesta

|  |  |  |
| --- | --- | --- |
| Contenedor exterior/ campo | Tipo | Descripción |
| ok | bool | Representa el éxito de la operación |
| msg | string | Mensaje de respuesta de la operación |

#### Códigos HTTP de estado

Este endpoint puede devolver uno de los siguientes códigos de estado HTTP

|  |  |
| --- | --- |
| Estado | Significado |
| 200 | Éxito |
| 400 | Mala petición |
| 401 | Autorización no válida |
| 500 | Error interno del servidor |

#### Ejemplos

##### Ejemplo 1

Borrar el folder “Chroniverse Universe” para el usuario logueado, en este caso con nickname el “LoveShadyKing”

###### Entrada

Enviamos la siguiente petición **PUT**:

|  |  |
| --- | --- |
| DELETE | /folder?folder\_name=Chroniverse Universe |

###### 

###### Salida

Obtenemos la siguiente respuesta:

|  |
| --- |
| {  "ok": true,  "msg": "Folder borrado con éxito" } |

# API: Work

## CreateWork

|  |  |
| --- | --- |
| POST | /work?folder\_name={folder\_name} |

Este método crea una obra en base de datos en el folder designado para el usuario logueado

### Entrada

#### Recurso URI

|  |  |
| --- | --- |
| POST | https://artemis-back.herokuapp.com/work?folder\_name={folder\_name} |

#### Parámetros URI

|  |  |  |  |
| --- | --- | --- | --- |
| Parámetro | Tipo | Descripción | Reglas de petición |
| folder\_name | string | Este parámetro representa el nombre del folder a obtenerse. | Requerido |

### 

#### Cuerpo

|  |  |  |  |
| --- | --- | --- | --- |
| Parámetro | Tipo | Descripción | Reglas de petición |
| name | String | Nombre de la obra | Requerido |
| private | Boolean | Si la obra es privada o pública | Requerido |
| private views | Array | Arreglo de espectadores privados (Personas que pueden ver la obra aunque sea privada) | Opcional |
| tag | String | Cadena que representa los tags de la obra | Requerido |
| version\_name | String | Nombre de la versión | Requerido |
| file | File | Archivo que representa la obra | Requerido |
| image | File | Imagen de la obra | Requerido |

### Salida

#### Respuesta de petición

|  |
| --- |
| {  "ok": true,  "msg": "Obra creada con éxito" } |

#### Campos de respuesta

|  |  |  |
| --- | --- | --- |
| Contenedor exterior/ campo | Tipo | Descripción |
| ok | bool | Representa el éxito de la operación |
| msg | string | Mensaje de respuesta de la operación |

#### Códigos HTTP de estado

Este endpoint puede devolver uno de los siguientes códigos de estado HTTP

|  |  |
| --- | --- |
| Estado | Significado |
| 200 | Éxito |
| 400 | Mala petición |
| 401 | Autorización no válida |
| 500 | Error interno del servidor |

#### Ejemplos

##### Ejemplo 1

Crear una obra con el nombre “Obra 1” para el folder llamado “Folder 1”

###### Entrada

Enviamos la siguiente petición **POST**:

|  |  |
| --- | --- |
| POST | /work?folder\_name=Folder 1 |

Junto con el siguiente **Body:**

|  |
| --- |
| {  "name": "Obra 1",  "private": false,  "tag": "emocionante",  "version\_name": "v0.1",  "file": obra.txt,  "image": obra.jpg } |

###### 

###### Salida

Obtenemos la siguiente respuesta:

|  |
| --- |
| {  "ok": true,  "msg": "Obra creada con éxito" } |

## GetWork

|  |  |
| --- | --- |
| GET | /work?folder\_name={folder\_name}&work\_name={work\_name} |

Este método obtiene una obra de nombre work\_name en el folder de nombre folder\_name

### Entrada

#### Recurso URI

|  |  |
| --- | --- |
| GET | https://artemis-back.herokuapp.com/work?folder\_name={folder\_name}&work\_name={work\_name} |

#### Parámetros URI

|  |  |  |  |
| --- | --- | --- | --- |
| Parámetro | Tipo | Descripción | Reglas de petición |
| folder\_name | string | Nombre del folder que contiene a la obra | Requerido |
| work\_name | string | Nombre de la obra | Requerido |

### Salida

#### Respuesta de petición

|  |
| --- |
| {  "tag": [  "emocionante"  ],  "private": false,  "private\_viewers": [],  "past\_versions": [],  "\_id": "5fff8b2769d9202cc05c4efd",  "name": "Obra 1",  "owner": "5ffe98fa7a7a8f36207cf51c",  "folder": "5ffe99c1912234318ccfa8c6",  "img\_url": "http://res.cloudinary.com/dc2gg0bh1/image/upload/v1610582824/yxicfrv2asda1.jpg",  "img\_public\_id": "yxicfrv2i1gw5jhfays2",  "\_\_v": 0,  "stats": "5fff8b3269d9202cc05c4eff",  "current\_version": "5fff8b2869d9202cc05c4efe" } |

#### Campos de respuesta

|  |  |  |
| --- | --- | --- |
| Contenedor exterior/ campo | Tipo | Descripción |
| tag | array | Tags de la obra |
| private | boolean | Si la obra es privada o no |
| private\_viewers | array | Arreglo de ids de espectadores privados |
| past\_versions | array | Arreglo de ids de versiones pasadas |
| \_id | string | Id del documento en base de datos |
| name | string | Nombre de la obra |
| owner | string | Id del propietario de la obra |
| folder | string | Id del folder de la obra |
| img\_url | string | Url de la imagen guardada en cloudinary |
| img\_public\_id | string | Id de la imagen guardada en cloudinary |
| \_\_v | string | Versión del documento en base de datos |
| stats | string | Id de las estadísticas de la obra en base de datos |
| current\_version | string | Id de la versión actual de la obra en base de datos |

#### Códigos HTTP de estado

Este endpoint puede devolver uno de los siguientes códigos de estado HTTP

|  |  |
| --- | --- |
| Estado | Significado |
| 200 | Éxito |
| 400 | Mala petición |
| 401 | Autorización no válida |
| 500 | Error interno del servidor |

#### Ejemplos

##### Ejemplo 1

Obtener una obra llamada Obra 1 desde el folder llamado Folder 1

###### Entrada

Enviamos la siguiente petición **GET**:

|  |  |
| --- | --- |
| GET | /work?folder\_name=Folder 1&work\_name=Obra 1 |

###### 

###### Salida

Obtenemos la siguiente respuesta:

|  |
| --- |
| {  "tag": [  "emocionante"  ],  "private": false,  "private\_viewers": [],  "past\_versions": [],  "\_id": "5fff8b2769d9202cc05c4efd",  "name": "Obra 1",  "owner": "5ffe98fa7a7a8f36207cf51c",  "folder": "5ffe99c1912234318ccfa8c6",  "img\_url": "http://res.cloudinary.com/dc2gg0bh1/image/upload/v1610582824/yxicfrv2asda1.jpg",  "img\_public\_id": "yxicfrv2i1gw5jhfays2",  "\_\_v": 0,  "stats": "5fff8b3269d9202cc05c4eff",  "current\_version": "5fff8b2869d9202cc05c4efe" } |

## GetWorks

|  |  |
| --- | --- |
| GET | /works |

Este método devuelve todos las obras en base de datos del usuario logueado

### Entrada

#### Recurso URI

|  |  |
| --- | --- |
| GET | https://artemis-back.herokuapp.com/works |

### Salida

#### Respuesta de petición

|  |
| --- |
| [  {  "tag": [  "emocionante"  ],  "private": false,  "private\_viewers": [],  "past\_versions": [],  "\_id": "5fff9d6869d9202cc05c4f01",  "name": "Obra 1",  "owner": "5ffe98fa7a7a8f36207cf51c",  "folder": "5ffe99c1912234318ccfa8c6",  "img\_url": "http://res.cloudinary.com/dc2gg0bh1/image/upload/v1610587497/mindue1aduxogn4qdasdaf.jpg",  "img\_public\_id": "mindue1aduxogn4qdasda",  "\_\_v": 0,  "stats": "5fff9d6a69d9202cc05c4f03",  "current\_version": "5fff9d6969d9202cc05c4f02"  },  {  "tag": [  "emocionante"  ],  "private": false,  "private\_viewers": [],  "past\_versions": [],  "\_id": "5fffa6c869d9202cc05c4f05",  "name": "Obra 2",  "owner": "5ffe98fa7a7a8f36207cf51c",  "folder": "5ffe99c1912234318ccfa8c6",  "img\_url": "http://res.cloudinary.com/dc2gg0bh1/image/upload/v1610589900/rxqkvndsoddokxwxjasd.jpg",  "img\_public\_id": "rxqkvndsoddokxwxasdd",  "\_\_v": 0,  "stats": "5fffa6ce69d9202cc05c4f07",  "current\_version": "5fffa6cd69d9202cc05c4f06"  } ] |

#### Campos de respuesta

|  |  |  |
| --- | --- | --- |
| Contenedor exterior/ campo | Tipo | Descripción |
| tag | array | Tags de la obra |
| private | boolean | Si la obra es privada o no |
| private\_viewers | array | Arreglo de ids de espectadores privados |
| past\_versions | array | Arreglo de ids de versiones pasadas |
| \_id | string | Id del documento en base de datos |
| name | string | Nombre de la obra |
| owner | string | Id del propietario de la obra |
| folder | string | Id del folder de la obra |
| img\_url | string | Url de la imagen guardada en cloudinary |
| img\_public\_id | string | Id de la imagen guardada en cloudinary |
| \_\_v | string | Versión del documento en base de datos |
| stats | string | Id de las estadísticas de la obra en base de datos |
| current\_version | string | Id de la versión actual de la obra en base de datos |

#### Códigos HTTP de estado

Este endpoint puede devolver uno de los siguientes códigos de estado HTTP

|  |  |
| --- | --- |
| Estado | Significado |
| 200 | Éxito |
| 400 | Mala petición |
| 401 | Autorización no válida |
| 500 | Error interno del servidor |

#### Ejemplos

##### Ejemplo 1

Obtener todas las obras del folder llamado Folder 1

###### Entrada

Enviamos la siguiente petición **GET**:

|  |  |
| --- | --- |
| GET | /works?folder\_name=Folder 1 |

###### Salida

Obtenemos la siguiente respuesta:

|  |
| --- |
| [  {  "tag": [  "emocionante"  ],  "private": false,  "private\_viewers": [],  "past\_versions": [],  "\_id": "5fff9d6869d9202cc05c4f01",  "name": "Obra 1",  "owner": "5ffe98fa7a7a8f36207cf51c",  "folder": "5ffe99c1912234318ccfa8c6",  "img\_url": "http://res.cloudinary.com/dc2gg0bh1/image/upload/v1610587497/mindue1aduxogn4qdasdaf.jpg",  "img\_public\_id": "mindue1aduxogn4qdasda",  "\_\_v": 0,  "stats": "5fff9d6a69d9202cc05c4f03",  "current\_version": "5fff9d6969d9202cc05c4f02"  },  {  "tag": [  "emocionante"  ],  "private": false,  "private\_viewers": [],  "past\_versions": [],  "\_id": "5fffa6c869d9202cc05c4f05",  "name": "Obra 2",  "owner": "5ffe98fa7a7a8f36207cf51c",  "folder": "5ffe99c1912234318ccfa8c6",  "img\_url": "http://res.cloudinary.com/dc2gg0bh1/image/upload/v1610589900/rxqkvndsoddokxwxjasd.jpg",  "img\_public\_id": "rxqkvndsoddokxwxasdd",  "\_\_v": 0,  "stats": "5fffa6ce69d9202cc05c4f07",  "current\_version": "5fffa6cd69d9202cc05c4f06"  } ] |

## UpdateWork

|  |  |
| --- | --- |
| PUT | /work?folder\_name={folder\_name}&work\_name={work\_name} |

Este método permite actualizar la información del usuario logueado.

### Entrada

#### Recurso URI

|  |  |
| --- | --- |
| PUT | https://artemis-back.herokuapp.com/work?folder\_name={folder\_name}&work\_name={work\_name} |

#### Parámetros URI

|  |  |  |  |
| --- | --- | --- | --- |
| Parámetro | Tipo | Descripción | Reglas de petición |
| folder\_name | string | Nombre del folder que contiene a la obra | Requerido |
| work\_name | string | Nombre de la obra | Requerido |

#### Cuerpo

|  |  |  |  |
| --- | --- | --- | --- |
| Parámetro | Tipo | Descripción | Reglas de petición |
| name | String | Este parámetro representa el nuevo nombre de la obra | Opcional |
| description | String | Este parámetro representa la nueva descripción de la obra | Opcional |
| private | String | Este parámetro representa si la obra será privada o no | Opcional |
| image | File | Este parámetro representa la nueva imagen de la obra | Opcional |
| tag | String | Este parámetro representa los nuevos tags de la obra | Opcional |
| private\_viewers | String | Este parámetro representa las referencias a los ids de los espectadores privados de las obras | Opcional |

### Salida

#### Respuesta de petición

|  |
| --- |
| {  "ok": true,  "msg": "Obra actualizada con éxito" } |

#### Campos de respuesta

|  |  |  |
| --- | --- | --- |
| Parámetro | Tipo | Descripción |
| ok | Bool | Representa el éxito de la operación. |
| msg | String | Mensaje de respuesta de la operación. |

#### Códigos HTTP de estado

|  |  |
| --- | --- |
| Estado | Significado |
| 200 | Éxito |
| 400 | Mala petición |
| 401 | Autorización no válida |
| 500 | Error interno del servidor |

#### Ejemplos

##### Ejemplo 1

###### Entrada

Enviamos la siguiente información por PUT

|  |  |
| --- | --- |
| PUT | https://artemis-back.herokuapp.com/work?folder\_name=Folder 1&work\_name=Obra 1 |

Junto con el siguiente body:

|  |
| --- |
| {  "name": "Obra 1 modificada",  "description": "Descripción 1 modificada",  "private": false,   "tag": "6.5/10 IGN",  "image": "obra1\_modificado.jpg" } |

###### Salida

|  |
| --- |
| {  "ok": true,  "msg": "Usuario actualizado con éxito" } |

## DeleteWork

### Entrada

|  |  |
| --- | --- |
| DELETE | /work?folder\_name={folder\_name}&work\_name={work\_name} |

Este método permite eliminar la información del usuario logueado junto con toda su información en otras colecciones como carpetas, trabajos, comentarios, comentarios en sus obras, estadísticas y versiones de sus trabajos.

### Entrada

#### Recurso URI

|  |  |
| --- | --- |
| DELETE | https://artemis-back.herokuapp.com/work?folder\_name={folder\_name}&work\_name={work\_name} |

#### Cuerpo

No tiene cuerpo

### Salida

#### Respuesta de petición

|  |
| --- |
| {  "ok": true,  "msg": "Obra borrada con exito" } |

#### Campos de respuesta

|  |  |  |
| --- | --- | --- |
| Parámetro | Tipo | Descripción |
| ok | Bool | Representa el éxito de la operación. |
| msg | String | Mensaje de respuesta de la operación. |

#### Códigos HTTP de estado

|  |  |
| --- | --- |
| Estado | Significado |
| 200 | Éxito |
| 400 | Mala petición |
| 401 | Autorización no válida |
| 500 | Error interno del servidor |

#### Ejemplos

##### Ejemplo 1

###### Entrada

Enviamos la siguiente información por DELETE

|  |  |
| --- | --- |
| DELETE | https://artemis-back.herokuapp.com/work?folder\_name=Folder 1&work\_name=Obra 1 |

###### Salida

#### 

|  |
| --- |
| {  "ok": true,  "msg": "Obra borrada exitosamente" } |