

Manual de programador (API REST)



ROAD RUNNER

Contenido

Extensiones PHP de la API REST	4
Firebase	4
PHPGPX.....	4
Llamadas.....	5
GET /api/checkuser/?username=XXXX	5
Headers	5
Query string.....	5
Respuestas.....	5
Ejemplo.....	5
POST /api/login/	6
Headers	6
Body.....	6
Respuesta	6
Ejemplo.....	6
POST /api/register/.....	7
Headers	7
Body.....	7
Respuesta	7
Ejemplo.....	7
GET /api/route/?id=XXXX	9
Headers	9
Query string.....	9
Respuestas.....	9
Ejemplo.....	9
GET /api/routes/[?name=XXXX] [?min_dist=XXXX] [?max_dist=XXXX] [?min_slope=XXXX] [?max_slope=XXXX] [?circular=(T F)] [?intensity=XXXX] [?activity=XXXX] [?user=XXXX].....	10
Headers	10
Query string.....	10
Respuesta	10
Ejemplo.....	10
POST /api/routes/.....	11
Headers	11
Body.....	11
Respuesta	11
Ejemplo.....	11

GET /api/user/?id=XXXX.....	13
Headers	13
Query string.....	13
Respuesta	13
Ejemplo.....	13
PUT /api/user/?id=XXXX.....	14
Headers	14
Query string.....	14
Body.....	14
Respuesta	14
Ejemplo.....	14
DELETE /api/user/?id=XXXX	16
Headers	16
Query string.....	16
Respuesta	16
Ejemplo.....	16
Otras Carpetas.....	17
Clases.....	17
GPX	17

Extensiones PHP de la API REST

Firebase

Versión: ^6.4

Documentación: <https://github.com/firebase/php-jwt>

PHPGPX

Versión 1.2.1

Documentación: <https://github.com/Sibyx/phpGPX>

Llamadas

GET /api/checkuser/?username=XXXX

Se le pasa en el query string un nombre de usuario y devuelve un JSON indicando si ya existe o no un usuario en el sistema con dicho nombre

Headers

Ninguno en especial

Query string

Nombre del usuario de la forma: ?username=XXXX

Respuestas

Cada status siempre devuelve un json con los posibles siguientes campos

- success (T|F): indica si la consulta se ha podido realizar en la Base de datos
- exists (T|F): si success es true indica si el nombre existe en la base de datos
- msg (String): si success es false indica el mensaje de error

Estados

- Status 200 OK
Devuelve un Json con: success: true y exists: true|false
- Status 400 Bad Request
Devuelve un Json con: success: false y msg: "Método incorrecto para llamada"
- Status 403 Error
Devuelve un Json con: success: false y msg: "Nombre no indicado"
- Status 404 Not Found
Devuelve un Json con: success: false y msg: "Error en la comprobación de usuarios"

Ejemplo

/api/checkuser/?username=llagui

Method: 'GET'

POST /api/login/

Inicia sesión para obtener un JWT.

Headers

'Content-Type': 'application/json; charset=utf-8'

Body

```
{  
  username: 'string',  
  pass: 'string'  
}
```

Respuesta

Cada status siempre devuelve un json con los posibles siguientes campos

- success (T|F): indica si la consulta se ha podido realizar en la Base de datos
- id (int): si success es true indica el id del usuario registrado
- token (String): si success es true indica el Token del usuario registrado para la sesión a iniciar
- msg (String): si success es false indica el mensaje de error

Estados

- Status 200 OK
Devuelve un Json con: success: true, con el id del usuario y el token para la sesión
- Status 400 Bad Request
Devuelve un Json con: success: false y msg: "Método incorrecto para llamada"
- Status 401 Error
Devuelve un Json con: success: false y msg: "Error en la comprobación de usuarios"
- Status 403 Error
Devuelve un Json con: success: false y msg: "Campo requerido vacío"
- Status 404 Not Found
Devuelve un Json con: success: false y msg: "No se encuentra el usuario. Inténtelo de nuevo más tarde"

Ejemplo

/api/login/

Method: 'POST'

Headers {

'Content-Type': 'application/json; charset=utf-8'
}

Body {

username: 'llagui',
pass: '1234'
}

POST /api/register/

Registra un usuario con los datos del body de la petición y devuelve un jwt si el usuario se ha registrado con éxito. Solo fullname, username, email y pass son obligatorios y height, weight, birthday, activities son opcionales al hacer la cuenta

Headers

'Content-Type': 'application/json; charset=utf-8'

Body

```
{
  fullname: 'string',
  username: 'string',
  email: 'string',
  pass: 'string',
  height: 'string',
  weight: 'string',
  birthday: 'string',
  activities: [array]
}
```

Respuesta

Cada status siempre devuelve un json con los posibles siguientes campos

- success (T|F): indica si la consulta se ha podido realizar en la Base de datos
- id (int): si success es true indica el id del usuario registrado
- token (String): si success es true indica el Token del usuario registrado para la sesión a iniciar
- msg (String): si success es false indica el mensaje de error

Estados

- Status 200 OK
Devuelve un Json con: success: true, con el id del usuario y el token para la sesión
- Status 400 Bad Request
Devuelve un Json con: success: false y msg: "Método incorrecto para llamada"
- Status 403 Error
Devuelve un Json con: success: false y msg: "Campo requerido vacío"
- Status 404 Not Found
Devuelve un Json con: success: false y msg: "No se puede registrar el usuario. Inténtelo de nuevo más tarde"

Ejemplo

/api/login/

Method: 'POST'

Headers {

'Content-Type': 'application/json; charset=utf-8'
}

Body {

fullname: 'Yago Nieto Garrido',

```
username: 'llagui',  
email: 'yago@mail.es',  
pass: '1234',  
height: '182',  
weight: '',  
birthday: '',  
activities: ['Senderismo',,,]  
}
```


GET /api/route/?id=XXXX

Devuelve un array con todos los datos de una ruta en específico

Headers

Ninguno en especial

Query string

Id de ruta de la forma: ?id=XXXX

Respuestas

Cada status siempre devuelve un json con los posibles siguientes campos

- success (T|F): indica si la consulta se ha podido realizar en la Base de datos
- 'ruta' (Array): si success es true devuelve los datos principales de la consulta a la base de datos (route_name, distance, slope, circular, start_lat, start_lon, user, desc, intensity, activity, gpx)
- 'estadisticas' (Array): si success es true devuelve los datos secundarios recogidos al recorrer el gpx asociado (distance, realDistance, avgSpeed, avgPace, minAltitude, maxAltitude, cumulativeElevationGain, cumulativeElevationLoss, startedAt, finishedAt, duration)
- 'puntos' (Array): si success es true devuelve con todos los puntos recogidos del gpx asociado. Array muy grande con puntos [[lat, lon], [lat, lon],...]
- 'usuario' (String): si success es true devuelve el nombre del usuario que subió la ruta
- msg (String): si success es false indica el mensaje de error

Estados

- Status 200 OK
Devuelve un Json con: success: true, 'ruta' Array con datos principales de la ruta, 'estadisticas' Array con datos secundarios, 'puntos' Array con latitud y longitud de todos los puntos de la ruta, 'usuario' el nombre de usuario del creador
- Status 400 Bad Request
Devuelve un Json con: success: false y msg: "Método incorrecto para llamada"
- Status 401 Route Error
Devuelve un Json con: success: false y msg: "La ruta no existe"
- Status 403 Error
Devuelve un Json con: success: false y msg: "Id no enviado"
- Status 404 Not Found
Devuelve un Json con: success: false y msg: "Error en la comprobación de usuarios"

Ejemplo

/api/checkuser/?username=llagui

Method: 'GET'

GET /api/routes/[?name=XXXX] [?min_dist=XXXX] [?max_dist=XXXX]
[?min_slope=XXXX] [?max_slope=XXXX] [?circular=(T|F)] [?intensity=XXXX]
[?activity=XXXX] [?user=XXXX]

Devuelve un array con todas las rutas que hay en la base de datos junto con sus datos principales filtrando según lo que se ponga

Headers

Ninguno en especial

Query string

Se pueden combinar

Nombre de ruta de la forma: [?name=XXXX]

Distancia mínima/máxima de la forma: [?min_dist=XXXX] [?max_dist=XXXX]

Distancia mínima/máxima de la forma: [?min_slope=XXXX] [?max_slope=XXXX]

Si es circular o no de la forma: [?circular=(T|F)]

La intensidad de la ruta(0-4): [?intensity=XXXX]

Tipo de actividad: [?activity=XXXX]

Las rutas del usuario(id): [?user=XXXX]

Respuesta

Cada status siempre devuelve un json con los posibles siguientes campos

- success (T|F): indica si la consulta se ha podido realizar en la Base de datos
- 'rutas' (Array): si success es true array con todas las rutas y sus datos principales (route_name, distance, slope, circular, start_lat, strat_lon, user, desc, intensity, activity, gpx)
- msg (String): si success es false indica el mensaje de error

Estados

- Status 200 OK
- Devuelve un Json con: success: true, array con todas las rutas y sus datos principales
- Status 400 Bad Request
Devuelve un Json con: success: false y msg: "Método incorrecto para llamada"
- Status 401 Route Error
Devuelve un Json con: success: false y msg: "Datos de ruta incorrectos"
- Status 404 Not Found
Devuelve un Json con: success: false y msg: "No se encuentran las rutas. Inténtelo de nuevo más tarde"

Ejemplo

/api/checkuser/?user=1&activity=Senderismo

Method: 'GET'

POST /api/routes/

Sube una nueva ruta a la base de datos y almacena en esta los datos principales leyendo el fichero y se sube a la carpeta /api/gpx

Headers

'Authorization': Token de JavaScript

'Data': Array con los datos "route_name", "user", "desc", "intensity", "activity, "circular"

Body

FormData con el archivo gpx

Respuesta

Cada status siempre devuelve un json con los posibles siguientes campos

- success (T|F): indica si la consulta se ha podido realizar en la Base de datos
- msg (String): si success es false indica el mensaje de error, si success es true indica que se ha creado el elemento

Estados

- Status 200 OK
Devuelve un Json con: success: true y msg: "Se ha creado la ruta"
- Status 400 Bad Request
Devuelve un Json con: success: false y msg: "Método incorrecto para llamada"
- Status 403 Error
Devuelve un Json con: success: false y msg: "Token no valido. Inténtelo de nuevo más tarde"
- Status 404 Not Found
Devuelve un Json con: success: false y msg: "No se puede registrar la ruta. Inténtelo de nuevo más tarde"

Ejemplo

```
let input = document.querySelector('input[type="file"]');  
let data = new FormData();  
data.append('file', input.files[0]);
```

/api/routes/

method: 'POST',

headers: {

'Authorization': jwt

'Data': JSON.stringify({

 "route_name": 'Hola',

 "user": '1',

 "desc": 'Ruta muy bonita',

 "intensity": 1,

```
        "activity": 'Senderismo',  
        "circular": true,  
    })  
}
```

```
body: data
```

GET /api/user/?id=XXXX

Devuelve un array con todos los datos de un usuario menos la contraseña

Headers

'Authorization': Token de JavaScript

Query string

Id del usuario de la forma: ?id=XXXX

Respuesta

Cada status siempre devuelve un json con los posibles siguientes campos

- success (T|F): indica si la consulta se ha podido realizar en la base de datos
- user (Array): si success es true Datos de un usuario menos la contraseña (fullname, username, email, birthday, height, weight, activities)
- msg (String): si success es false indica el mensaje de error

Estados

- Status 200 OK
Devuelve un Json con: success: true y user array con los datos de usuario
- Status 400 Bad Request
Devuelve un Json con: success: false y msg: "Método incorrecto para llamada"
- Status 401 User Error
Devuelve un Json con: success: false y msg: "Datos de usuario incorrectos"
- Status 403 Error
Devuelve un Json con: success: false y msg: "Id no enviado"
- Status 404 Not Found
Devuelve un Json con: success: false y msg: "No se encuentra el usuario. Inténtelo de nuevo más tarde"
- Status 405 Not Found
Devuelve un Json con: success: false y msg: "Token incorrecto"

Ejemplo

../api/user/?id=3

method: 'GET',

headers: Authorization: jwt

PUT /api/user/?id=XXXX

Actualizar los datos de un usuario en la base de datos

Headers

'Authorization': Token de JavaScript

Query string

Id del usuario de la forma: ?id=XXXX, es necesario para la comprobación del token

Body

```
{
  id: id,
  email: 'string',
  [pass: 'string'],
  height: 'string',
  weight: 'string',
  birthday: 'string',
  activities: [array]
}
```

Respuesta

Cada status siempre devuelve un json con los posibles siguientes campos

- success (T|F): indica si la consulta se ha podido realizar en la base de datos
- msg (String): si success es false indica el mensaje de error y si success es true da la confirmación de que se ha creado el usuario

Estados

- Status 200 OK
Devuelve un Json con: success: true y msg: "Usuario actualizado"
- Status 400 Bad Request
Devuelve un Json con: success: false y msg: "Método incorrecto para llamada"
- Status 401 User Error
Devuelve un Json con: success: false y msg: "Campos requerido vacío"
- Status 404 Not Found
Devuelve un Json con: success: false y msg: "No se puede actualizar el usuario. Inténtelo de nuevo más tarde "
- Status 405 Not Found
Devuelve un Json con: success: false y msg: "Token incorrecto"

Ejemplo

```
let elementos = {
  "id": 1,
  "email": "hoal@mail.es",
  "height": "",
  "weight": "",
  "birthday": "",
  "activities": ["Senderismo",,,,],
};
../api/user/?id=1
```

```
method: 'PUT',  
headers: {  
  'Content-Type': 'application/json;charset=utf-8',  
  'Authorization': `${localStorage.getItem('token')}`,  
},  
body: JSON.stringify(elementos)
```

DELETE /api/user/?id=XXXX

Actualizar los datos de un usuario en la base de datos

Headers

'Authorization': Token de JavaScript

Query string

Id del usuario de la forma: ?id=XXXX, es necesario para la comprobación del token

Respuesta

Cada status siempre devuelve un json con los posibles siguientes campos

- success (T|F): indica si la consulta se ha podido realizar en la base de datos
- msg (String): si success es false indica el mensaje de error y si success es true da la confirmación de que se ha creado el usuario

Estados

- Status 200 OK
Devuelve un Json con: success: true y msg: "Usuario actualizado"
- Status 400 Bad Request
Devuelve un Json con: success: false y msg: "Método incorrecto para llamada"
- Status 403 Error
Devuelve un Json con: success: false y msg: "Id no enviado"
- Status 404 Not Found
Devuelve un Json con: success: false y msg: "No se puede eliminar el usuario. Inténtelo de nuevo más tarde"
- Status 405 Not Found
Devuelve un Json con: success: false y msg: "Token incorrecto"

Ejemplo

```
../api/user/?id=1  
method: 'DELETE',  
headers: {  
  'Authorization': `${sessionStorage.getItem('token')}`,  
},
```


Otras Carpetas

Clases

Carpeta donde está la conexión a la base de datos

GPX

Carpeta donde se guardan todos los GPX que se suben a la API