

Routes DogWalker_API

Content

Api wide errors	2
User authorization type.....	2
Rate Limits	2
Schema validations	3
Mascotas.....	4
Clientes	6
Paseadores.....	8
Admin.....	10
queriesMascotas	12
Area	13
Location	14
FranjaHoraria	15
Servicio	16
Paseo.....	18

Api wide errors

- 401 Unauthorized: {error: "Token not valid "} or {error: "No token"} or {error: "Token information not valid"}
- 403 Forbidden: { error: 'Auth not valid' }
- 404 Page not found: { error: "Page not found" }
- 422 Unprocessable Entity {error: <automatic message with errors in format or/and contend of the data send in the request>}
- 429 Too Many Requests {error: <custom over rate message>}
- 500 Internal Server Error { error: "Something Broke", message: error message}

User authorization type

They are 3 types of user that will use the API via a Front End. Each type of user will have access to different routes of the API. To verify the identity for each request , the user must send a JWT token in the auth field of the request header. This token will be provided by the API when a post */log request is executed, and it will have an expiration time of 1h.

In the route tables the authorise tokens for each route will be indicate by the "User authorization type" column. These are:

- Admin: only users with tokens form the /admin/log route
- Cliente: only users with tokens from the /cliente/log route
- Paseador: only users with tokens from the /paseador/log route
- Anyone: no token must be provided

Rate Limits

The API will have a limit rate for each kind of user if they provide a JWT

- Admin: 60 req/ 1 min
- Cliente: 25 req 1 /min
- Paseador: 35 req/ 1min
- Anyone: 10 req/ 1 min

Each main route will be provided its own rate limit and some specific routes like the */add , */update and */log will have a more restrictive rate limit indicated in the "Rate limit per IP" column or None if they do not require it

Schema validations

For every single data send by body or params , a middleware using Joi (<https://www.npmjs.com/package/joi>). The rules for every single field send are specified in detail at ./DogWalker_API/services/schemas.js in the program source file . They are divided by schemes that are used in one or many routes of the API , also on combine with others.

For more information about the rules applies for it each field of the data send, check out the Joi documentation : <https://joi.dev/api/?v=17.9.1>

Mascotas

Rate limit: 30 req/ 1 min

Route	Send	Function	Response	User authorization type	Rate limit per IP
GET /mascotas/all	None	Return Json with list of all content in “mascotas” table	200: Json list of Mascotas (empty if none) 400: error message	Admin	None
GET /mascotas/:pet_token	None	Return Json with row in “mascotas” table with pet_token send by param	200: Json list of Mascotas (empty if none) 422: {error: “message with format mistakes in the pet_token send by params”} 400: error message	Admin Cliente Paseador	
GET /mascotas/cliente/:client_ID	None	Return Json with rows in “mascotas” table with the client_ID send by param	200: Json list of Mascotas (empty if none) 422: {error: “message with format mistakes in the client_ID send by params”} 400: error message	Admin Cliente	
GET /mascotas/paseador/:walker_ID	None	Return Json with rows in “mascotas” table with the walker_ID send by param	200: Json list of Mascotas (empty if none) 422: {error: “message with format mistakes in the walker_ID send by params”} 400: error message	Admin Paseador	
GET /mascotas/date/:intDate/:endDate	None	Return Json with rows in “mascotas” table with renovation_date field between intDate and endDate	200: Json list of Mascotas (empty if none) 422: {error: “message with format mistakes in the intDate and endDate send by params”} 400: error message	Admin	
POST /mascotas/add	Req.body={ client_ID: client_ID, pet_name: pet_name,	Add a “Mascota” to the database.	201: {pet_token: pet_token} 422: {error : “message with the format mistakes in the req.body”}	Admin Cliente	8 req / 1min

	pet_breed:, pet_breed, service:, service, renovation_date: renovation_date }		400: { error: "Unable to add" }		
PUT /mascotas/update	Req.body={ pet_token: pet_token, client_ID: client_ID, pet_name: pet_name, pet_breed:, pet_breed, service:, service, renovation_date: renovation_date }	Update a "Mascota" with pet_token in the database.	201: {pet_token: pet_token} 422: {error: "message with the format mistakes in the req.body"} 404: { error: "No data has been update"} 400: { error: "Unable to update" }	Admin Cliente	5 req / 1min
DELETE /mascotas/ delete /:pet_token	None	Delete "Mascota" with pet_token in the database	200: 1 404: 0 422: {error: "message with format mistakes in the pet_token send by params"} 400: { error: "Unable to delete" }	Admin	4 req / 1min

Clientes

Rate limit: 25 req/ 1 min

Route	Send	Function	Response	Autorization type	Rate limit per IP
GET /clientes/all	None	Return Json with list of all content in "clientes" table	200: Json list of Clientes (empty if none) 400: error message	Admin	None
GET /clientes/:client_ID	None	Return Json with row in "clientes" table with client_ID send by param	200: Json list of Clientes (empty if none) 422: error: "message with format mistakes in the client_ID send by params". 400: error message	Admin Paseador Cliente	
POST /clientes/add	Req.body={ client_tel: client_tel, client_ID: client_ID, client_name: client_name, start_date: start_date, client_email: client_email, client_user: client_user, client_password : client_password, location_id : location_id, address_link: address_link }	Add a "Cliente" to the database (password is hash)	201: {token: encrypted jwt token , client_ID: client_ID} 409: { error: "User not valid" } 422: {error : "message with the format mistakes in the req.body"} 400: { error: "Unable to add" }	Admin	8 req / 1min
POST /cliente/log	Req.body={ client_user: client_user, client_password : client_password,	Check if credentials are valid for "clientes" table	200: {token: encrypted jsu token , client_ID: client_ID} 404: { error: "Data not valid" } 422: {error: "message with the format mistakes in the req.body]	Anyone	10 req / 5min

	}		400: error message		
PUT /clientes/update	Req.body={ client_tel: client_tel, client_ID: client_ID, client_name: client_name, start_date: start_date, client_email: client_email, location_id : location_id, address_link: address_link, }	Update a “Cliente” to the database with client_ID	201: {client_tel: client_tel} 404: { error: "No data has been update" } 422: {error : “message with the format mistakes in the req.body”} 400: { error: "Unable to update" }	Admin	5 req / 1min
PUT /clientes/update/auth	Req.body={ client_user: client_user, client_password : client_password, new_user: new_user, new_password : new_ password }	Update old “Cliente” credentials with new in the database (password is hash)	201: { user : new_user} 404: { error: "No data has been update" } 409: { error: "Credentials not valid" } or { error: "New user not valid" } 422: {error: “message with the format mistakes in the req.body”} 400: { error: "Unable to update" }	Cliente Admin	1 req / 1h
DELETE /clientes/delete /:client_ID	None	Delete “Cliente” with client_ID in the database	200: 1 404: 0 422: error: “message with format mistakes in the client_ID send by params”. 400: { error: "Unable to delete" }	Admin	4 req / 1min

Paseadores

Rate limit: 25 req/ 1 min

Route	Send	Function	Response	User authorization type	Rate limit per IP
GET /paseadores/all	None	Return Json with list of all content in "paseadores" table	200: Json list of Paseadores (empty if none) 400: error message	Admin	None
GET /paseadores/:walker_ID	None	Return Json with row in "paseadores" table with walker_ID send by param	200: Json list of Paseadores (empty if none) 422: {error: "message with format mistakes in the walker_ID send by params"} 400: error message	Admin Cliente Paseador	
POST /paseadores/add	Req.body={ walker_ID: walker_ID, walker_tel , walker_tel, walker_name: walker_name, start_date: start_date, walker_user: walker_user, walker_password : walker_password, walker_address: walker_address, walker_linkaddress,: walker_linkaddress, walker_photoURL: walker_photoURL, walker_bloodtype : walker_bloodtype }	Add a "Paseador" to the database (password is hash)	201: {token: encrypted jwt token , walker_ID: walker_ID} 409: { error: "User not valid" } 422: {error: "message with the format mistakes in the req.body"} 400: { error: "Unable to add" }	Admin	8 req / 1min
POST /paseadores /log	Req.body={ walker_user: walker_user,	Check if credentials are valid for "paseadores" table	200: {token: encrypted jsu token, walker_ID: walker_ID}	Anyone	10 req / 5min

	walker_password : walker_password,		404: { error: "Data not valid" } 422: {error: "message with the format mistakes in the req.body"} 400: error message		
PUT /paseadores/update	Req.body={ walker_ID: walker_ID, walker_tel , walker_tel, walker_name: walker_name, start_date: start_date, alker_address: walker_address, walker_linkaddress,: walker_linkaddress, walker_photoURL: walker_photoURL, walker_bloodtype : walker_bloodtype }	Update a "Paseador" to the database with client_tel	201: {walker_ID: walker_ID} 404: { error: "No data has been update" } 422: {error: "message with the format mistakes in the req.body"} 400: { error: "Unable to update" }	Admin	5 req / 1min
PUT /paseadores/update/auth	Req.body={ walker_user: walker_user, walker_password : walker_password, new_user: new_user, new_password : new_password }	Update old "Paseador" credentials with new in the database	201: { user : new_user} 404: "Unable to update" 409: { error: "Credentials not valid" } or { error: "New user not valid" } 422: {error: "message with the format mistakes in the req.body"} 400: { error: "Unable to update" }	Admin Paseador	1 req / 1h
DELETE /paseadores / delete /:walker_ID	None	Delete "Pasedor" with pet_token in the database	200: 1 404: 0 422: {error: "message with format mistakes in the walker_ID send by params"} 400: { error: "Unable to delete" }	Admin	4 req / 1min

Admin

Rate limit: 25 req/ 1 min

Route	Send	Function	Response	User authorization type	Rate limit per IP
GET /admin/all	None	Return Json with list of all content in "admin" table	200: Json list of Admin (empty if none) 400: error message	admin	None
POST /admin/add	Req.body={ admin_username: admin_username , admin_password: admin_password }	Add a "Admin" to the database (password is hash)	201: {token: encrypted jsa token} 409: {error:"User not valid"} 422: {error: "message with the format mistakes in the req.body"} 400: { error: "Unable to add" }		8 req / 1min
POST /admin/log	Req.body={ admin_username: admin_username, admin_password : admin_password }	Check if credentials are valid for "pasedores" table	200: {token: encrypted jsa token} 404: { error: "Data not valid" } 422: {error : "message with the format mistakes in the req.body"} 400: error message	anyone	10 req / 5min
PUT /admin/update/auth	Req.body={ admin_username: admin_username, admin_password : admin_password, new_user: new_user, new_password : new_password }	Update old "Admin" credentials with new in the database	201: {user: new_user} 404: { error: "No data has been update" } 409: { error: "Credentials not valid" } or { error: "New user not valid" } 422: {error : "message with the format mistakes in the req.body"} 400: { error: "Unable to update" }	admin	1 req / 1h
DELETE /admin/delete /:admin_username	None	Delete "Admin" with pet_token in the database	200: 1 404: 0 422: {error: "message with format mistakes in the admin_username send by params"} 400: { error: "Unable to delete" }		4 req / 1min
PUT /admin/restore /cliente	Req.body={ client_ID: client_ID }	Restore the Cliente password to its client_ID	201: 1	admin	1 req / 1h

			422: {error: "message with format mistakes in client_tel send by req.body"} 404: { error: 'No data has been update' } 400: { error: 'Unable to update' }		
PUT /admin/restore /cliente	Req.body={ walker_ID: walker_ID: }	Restore the Paseador password to its walker_ID	201: 1 422: {error: "message with format mistakes in walker_ID send by req.body"} 404: { error: 'No data has been update' } 400: { error: 'Unable to update' }		1 req / 1h

queriesMascotas

Rate limit: 40 req/ 1 min

Route	Send	Function	Response	User authorization type	Rate limit per IP
GET /mascotasQueries/clientes/paseadores/all	None	Return list of “clientes” table join with “mascotas” table join with table “paseadores”	200: [{client_name , walker_name , pet_name , service , pet_breed , location , area } ,{...} ...,{...}] 400: error message	Admin	None
GET /mascotasQueries /clientes/all	None	Return list of “clientes” table join with “mascotas” table join	200: [{area, location, client_name , pet_name , service , pet_breed , } ,{...} ...,{...}] 400: error message		
GET /mascotasQueries /clientes/area/:area	None	Return list of “clientes” table join with “mascotas” table join with area equal to req.param.area	422: error: “message with format mistakes in the params”.		
GET /mascotasQueries/clientes/location/:location	None	Return list of “clientes” table join with “mascotas” with location equal to req.param.location			
GET /mascotasQueries /paseadores/all	None	Return list of “paseadores” table join with “mascotas”	200: [{ walker_name , pet_name , service , pet_breed , location , area } ,{...} ...,{...}] 400: error message		
GET /mascotasQueries/paseadores/:walker_name	None	Return list of “paseadores” table join with “mascotas” with walker_name equal to req.param. walker_name			

Area

Rate limit: 40 req/ 1 min

Route	Send	Function	Response	User authorization type	Rate limit per IP
GET /areas/all	None	Return Json with list of all content in “area” table	200: Json list of Area (empty if none) 400: error message	Admin	None
GET /areas/:area_id	None	Return Json with row in “area” table with area _id send by param	200: Json list of Area(empty if none) 422: error: “message with format mistakes in the area _id send by params”. 400: error message	Admin, Cliente, Paseador	
POST /areas/add	Req.body={ area_name: area_name }	Add a “area” to the database	201: {area_id} 422: {error : “message with the format mistakes in the req.body”} 400: { error: "Unable to add" }	Admin	8 req / 1min

Location

Rate limit: 40 req/ 1 min

Route	Send	Function	Response	User authorization type	Rate limit per IP
GET /locations/all	None	Return Json with list of all content in "location" table	200: Json list of Location (empty if none) 400: error message	Admin	None
GET /locations/:location_id	None	Return Json with row in "location" table with location_id send by param	200: Json list of Location (empty if none) 422: error: "message with format mistakes in the area_id send by params". 400: error message	Admin, Cliente, Paseador	
POST /locations/add	Req.body={ location_name : location_name, area_id:area_id }	Add a "location" to the database	201: { location_id } 422: {error : "message with the format mistakes in the req.body"} 400: { error: "Unable to add" }	Admin	8 req / 1min

FranjaHoraria

Rate limit: 40 req/ 1 min

Route	Send	Function	Response	User authorization type	Rate limit per IP
GET /franjaHoraria/all	None	Return Json with list of all content in "franjaHoraria" table	200: Json list of franjaHoraria (empty if none) 400: error message	Admin	None
GET /franjaHoraria/:franja_id	None	Return Json with row in "franjaHoraria" table with franja_id send by param	200: Json list of franjaHoraria (empty if none) 422: error: "message with format mistakes in the area _id send by params". 400: error message	Admin, Cliente, Paseador	
POST /franjaHoraria/add	Req.body={ start_minute: start_minute, end_minute: end_minute}	Add a "franjaHoraria" to the database	201: {convertido , franja_id} 422: {error : "message with the format mistakes in the req.body"} 409: { error: " Time range not valid " } 400: { error: "Unable to add" }	Admin	8 req / 1min

Servicio

Rate limit: 25 req/ 1 min

Route	Send	Function	Response	User authorization type	Rate limit per IP
GET /servicios/all	None	Return Json with list of all content in "servicio" table	200: Json list of Servicio (empty if none) 400: error message	Admin	None
GET /servicios/:servicio_ID	None	Return Json with row in "servicio" table with servicio_ID send by param	200: Json list of Servicio (empty if none) 422: error: "message with format mistakes in the servicio_ID send by params". 400: error message	Admin Paseador Cliente	
GET / servicios /mascotas/:pet_token	None	Return Json with the last row in "servicio" table with pet_token send by param	200: Json list of Servicio (empty if none) 422: error: "message with format mistakes in contend send by params". 400: error message	Admin Cliente Paseador	None
GET / servicios /mascotas/all/:pet_token	None	Return Json with rows in "servicio" table with pet_token send by param		Admin Cliente	
GET /servicios /paseadores/:walker_ID	None	Return Json with rows in "servicio" table with walker_ID send by param		Admin Paseador	
Get /servicios /mascota/paseador/:pet_token/:walker_ID	None	Return Json with rows in "servicio" table with walker_ID and pet_token send by param		Admin Paseador	
POST /servicios/add	Req.body={ pet_token: pet_token , franja_id: franja_id, walker_ID, walker_ID}	Add a "servicio" to the database	201: {servicio_ID} 422: {error : "message with the format mistakes in the req.body"} 409: { error: "Pet Token does not exist" } 409: { error: "walker_ID does not exist" } 409: { error: " walker_ID not valid" }	Admin	8 req / 1min

			400: { error: "Unable to add" }		
PUT /servicios/update	Req.body={ servicio_ID: servicio_ID , franja_id: franja_id, walker_ID, walker_ID}	Update a "Servicio" to the database with servicio_ID	201: { servicio_ID, servicio_ID} 404: { error: "No data has been update" } 422: {error: "message with the format mistakes in the req.body"} 409: { error: "walker_ID does not exist" } 409: { error: " walker_ID not valid" } 400: { error: "Unable to update" }	Admin	5 req / 1min
DELETE /servicios/delete/:servicio_ID	None	Delete "servicio" with servicio_ID in the database	200: 1 404: 0 422: {error: message with format mistakes in the servicio_ID send by params} 400: { error: "Unable to delete" }	Admin	4 req / 1min

Paseo

Rate limit: 30 req/ 1 min

Route	Send	Function	Response	User authorization type	Rate limit per IP
GET /paseos/all	None	Return Json with list of all content in “paseo” table	200: Json list of Paseo(empty if none) 400: error message	Admin Paseador must be able to manage their own Cliente	None
GET / paseos/:paseo_ID	None	Return Json with row in “paseo” table with paseo_ID send by param	200: Json list of Paseo(empty if none) 422: error: “message with format mistakes in the paseos_id send by params”. 400: error message		
GET / paseos /servicio/:servicio_ID	None	Return all rows in “paseo” table with that “servicio_ID” , that have already ended	200: Json list of Paseo(empty if none) 422: error: “message with format mistakes in the servicio_ID send by params”. 400: error message	Admin Cliente Paseador	
GET /paseos paseador/current/:walker_ID	None	Return all rows in “paseo” table with that “walker_ID” , that have not ended	200: Json list of Paseo(empty if none) 422: error: “message with format mistakes in the walker_ID send by params”. 400: error message	Admin Paseador	
GET / paseos/count /paseador/:walker_ID	None	Return total of “paseo” with that “walker_ID” that have already ended	200: {total: int} 400: error message 422: error: “message with format mistakes in the params”. 400: error message	Admin Paseador must be able to manage their own Cliente	
GET / paseos/count/ /mascota/:pet_token	None	Return total of “paseo” with that “pet_token” that have already ended			
GET / paseos/count /mascota/paseador/:pet token/:walker ID	None	Return total of “paseo” with that “walker ID” and			

		"pet_token" that have already ended			
GET /paseos/count /paseador/servicio/:walker_ID/:servicio_ID	None	Return total of "paseo" with that "walker_ID" and "servicio_ID" that have already ended		Admin	
POST / paseos /start	Req.body={ walker_ID: walker_ID servicio_ID: servicio_ID }	Add "paseo" to the database and set its star_date to the local datetime	201: {paseo_ID} 422: {error : "message with the format mistakes in the req.body"} 400: { error: "Unable to start" }	Admin Paseador	8 req / 1min
PUT / paseos /end	Req.body={ paseo_ID: paseo_ID, evidenceURL: evidenceURL }	Update "paseo" with end_date to the local datetime. If the row already has an end_date, the "paseo" is not updated	200: 1 404: { error: 'Any paseo has not been ended' } 400: { error: "Unable to end" }		5 req / 1min