

Vamo' API

Documentación del API de Vamo'.

A continuación, se describen todos los endpoints definidos por el API, categorizados según la principal clase con la que interactúa cada endpoint (puede que internamente interactúe con otras clases).

Table of Contents

Table of Contents

Main

- Vamo
- Login user

Client

- Register client
- Get all clients
- Find clients by keyword
- Get client
- Update client
- [Admin] Delete client
- Get reviews from client
- Get all "pending" from client
- Add place to "pending"
- Remove place from "pending"
- Get all "liked" from client
- Add place to "liked"

- Remove place from "liked"
- Get all posts from client
- Publish post
- Delete post
- Get all missions from a client
- Report location to accomplish missions

Business

- Get business
- Register business
- Update business
- [Admin] Delete business

Place

- Create a prospect of place
- [Admin] Validate a prospect of place
- [Admin] Create a place
- Update a place
- [Admin] Delete a place
- Add tag to place
- Remove tag from place
- [Admin] Associate a place to a business
- [Admin] Dissociate a place to a business
- Search place by keywords
- Get place
- Get posts from place
- Get reviews from place
- Publish review to place
- Update review from place

- Remove review from place
- Get events of place
- Create event
- Update event
- Delete event
- Get photos/videos from place
- Add photo/video to place
- Delete photo/video from place

Event

- Get photos/videos from event
- Add photo/video to event
- Delete photo/video from event

MissionClient

- [Admin] Disassociate a mission from all clients that haven't completed it
- [Admin] Disassociate a mission from a client

Mission

- [Admin] Create a mission
- [Admin] Update a mission

Rank

- Get all ranks
- [Admin] Create a rank
- [Admin] Update a rank
- [Admin] Delete a rank

Tag

- Get all tags
- [Admin] Create a tag

- [Admin] Update a tag
- [Admin] Delete a tag

NOTAS de desarrollador

Tipos de Tokens

Consideraciones importantes al momento de desarrollar

Main

▪ Vamo

Endpoint URL	localhost:3000/api/vamo
HTTP Method	GET
Authorization	Required ANY CLIENT token

Description:

Recibe los ID de los clientes y otros datos de búsqueda y retorna todos los lugares relacionados a los gustos de todos los clientes y a los datos de búsqueda.

Body:

Una lista con los IDs de uno o más clientes, una dirección (longitud y latitud) y un radio (en km) a la redonda de la dirección dada.

```
{
  "clients": [
    "0a0a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a",
    "1a0a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a",
    ...
  ],
  "longitude": "...",
  "latitude": "...",
  "radius": 5
}
```

```
}
```

Response:

Lista de lugares recomendados según los gustos de todos los clientes, y que estén al número de km dado, a la redonda de la dirección dada.

```
[
  {
    "id": "0p0a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a",
    "name": "Default Place",
    "picture": "...",
    "address": "...",
    "longitude": "...",
    ...
  },
  {
    ...
  }
]
```

HTTP status responses:

- ∴ 200: Se retornaron los lugares correctamente.
- ∴ 204: Se resolvieron los clientes, pero no se encontró ningún lugar para recomendar, con los criterios dados.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: Alguno de los IDs enviados no está registrado.

■ Login user

Endpoint URL	localhost:3000/api/login
HTTP Method	GET
Authorization	Required UNREGISTERED token

Description:

Recibe la información de login de un usuario (**ya sea un cliente o un negocio**) y se encarga de identificar si el usuario está o no registrado en la aplicación.

Body:

Información del usuario a loguear; email y contraseña.

```
{  
  "email": "user@email.com",  
  "password": "password"  
}
```

Response:

En caso de que el usuario esté registrado en la aplicación, se retornará un payload con la información principal del cliente o negocio.

```
{  
  "id": "0a0a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a",  
  "type": "client",  
  "email": "user@email.com",  
  "name": "Default User",  
  ...  
}
```

HTTP status responses:

- ∴ 200: El usuario es logueado correctamente.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: En caso de que el usuario no se encuentre registrado.

Client

▪ Register client

Endpoint URL	localhost:3000/api/clients
HTTP Method	POST

Authorization	Required UNREGISTERED token
---------------	-----------------------------

Description:

Recibe la información de registro de un cliente y se encarga de validar el cliente y persistirlo a la base de datos.

Body:

Toda la información del cliente junto con unos tags iniciales de los gustos del cliente.

```
{
  "name": "Default Client",
  "email": "user@email.com",
  "description": "This is a Vamo' account for a default client",
  "tags": [
    "colorful",
    "rural",
    "videogames",
    "pet friendly",
    ...
  ],
  "picture": "https://i.picsum.photos/id/242/536/354.jpg?hmac=eBmLX_mGK8UV50pe3GydmYjbGebCFjH6SiSTZUNd944"
}
```

Response:

Información del cliente con sus tags, su ID, un contador de XP = 0 y el rango del cliente.

```
{
  "id": "0a0a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a",
  "name": "Default Client",
  "email": "user@email.com",
  "password": "password",
  "description": "This is a Vamo' account for a default client",
  "tags": [
    "colorful",
    "rural",
    "videogames",
    "pet friendly",
    ...
  ],
  "xp": 0,
  "rank": 1
}
```

```
    "picture": "https://i.picsum.photos/id/242/536/354.jpg?hmac=eBmLX_mGK8UV50pe3GydmYjbGebCFjH6SiSTZUNd944",
    "xp": 0,
    "rank": {
      "name": "wood",
      "level": 0,
      "xpNext": 2500
    }
  }
}
```

HTTP status responses:

- ∴ 201: El cliente es registrado correctamente.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 412: El usuario no validó todas las reglas de negocio.

■ Get all clients

Endpoint URL	localhost:3000/api/clients
HTTP Method	GET
Authorization	Required BASIC token

Description:

Retorna la información básica de todos los clientes.

Response:

Información básica de todos los clientes, para mostrarla en una lista sin detalles.

```
[
  {
    "id": "0a0a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a",
    "name": "Default Client",
    "description": "This is a Vamo' account for a default client",
    "picture": "...",
    "rank": {
      "name": "wood",
      "level": 0,
      "xpNext": 2500
    }
  }
]
```



```
    }  
  },  
  {  
    ...  
  }  
]
```

HTTP status responses:

- ∴ 200: Se obtiene la lista de clientes.
- ∴ 204: No hay ningún cliente registrado.
- ∴ 401: El token dado no posee los permisos para este endpoint.

■ Find clients by keyword

Endpoint URL	localhost:3000/api/clients?q={q}
HTTP Method	GET
Authorization	Required BASIC token

Description:

Retorna la información básica de todos los clientes que contengan parcialmente alguna de las palabras clave.

Parameters:

- ∴ {q}: Palabras clave para la búsqueda de los clientes.

Response:

Información básica de todos los clientes, para mostrarla en una lista sin detalles.

```
[  
  {  
    "id": "0a0a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a",  
    "name": "Default Client",  
    "description": "This is a Vamo' account for a default client",  
    "picture": "...",  
    "rank": {  
      "name": "wood",
```

```

        "level":0,
        "xpNext":2500
    },
    {
        ...
    }
]

```

HTTP status responses:

- ∴ 200: Se obtiene la lista de clientes.
- ∴ 204: No hay ningún cliente para mostrar.
- ∴ 401: El token dado no posee los permisos para este endpoint.

■ Get client

Endpoint URL	localhost:3000/api/clients/{id}
HTTP Method	GET
Authorization	Required BASIC token

Description:

Retorna toda la información del perfil de un cliente.

Parameters:

- ∴ {id}: ID del cliente.

Response:

Información completa del cliente.

```

{
  "id":"0a0a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a",
  "name":"Default Client",
  "email":"user@email.com",
  "description":"This is a Vamo' account for a default client",
  "tags":[
    "colorful",
    "rural",
    "videogames",
  ]
}

```

```
        "pet friendly",
        ...
    ],
    "picture": "...",
    "xp": 0,
    "rank": {
        "name": "wood",
        "level": 0,
        "xpNext": 2500
    }
}
```

HTTP status responses:

- ∴ 200: Se obtiene el cliente con el ID dado.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: El cliente con dicho ID no fue encontrado.

■ Update client

Endpoint URL	localhost:3000/api/clients/{id}
HTTP Method	PUT
Authorization	Required CLIENT X token

Description:

Permite actualizar la información de un cliente.

Parameters:

- ∴ {id}: ID del cliente.

Body:

Información por modificar del cliente, no se debe especificar toda la información.

```
{
  "description": "Updated description for the client"
}
```

Response:

Información completa del cliente, actualizada.

```
{
  "id": "0a0a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a",
  "name": "Default Client",
  "email": "user@email.com",
  "description": " Updated description for the client",
  "tags": [
    "colorful",
    "rural",
    "videogames",
    "pet friendly",
    ...
  ],
  "picture": "https://i.picsum.photos/id/242/536/354.jpg?hmac=eBmLX_mGK8UV50pe3GydmYjbGebCFjH6SiSTZUNd944",
  "xp": 0,
  "rank": {
    "name": "wood",
    "level": 0,
    "xpNext": 2500
  }
}
```

HTTP status responses:

- ∴ 200: El usuario es actualizado correctamente.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 403: Se está intentando modificar información que no se puede modificar, como el ID.
- ∴ 404: El usuario con dicho ID no fue encontrado.
- ∴ 412: El usuario no validó todas las reglas de negocio.

▪ [Admin] Delete client

Endpoint URL	localhost:3000/api/clients/{id}
HTTP Method	DELETE
Authorization	Required ADMIN token

Description:

Elimina por completo a un cliente de la aplicación.

Parameters:

∴ {id}: ID del cliente.

Response:

No se obtiene cuerpo de respuesta, únicamente un mensaje OK o de error con el mensaje correspondiente.

HTTP status responses:

- ∴ 204: El usuario fue eliminado correctamente.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: El usuario con dicho ID no fue encontrado.

■ Get reviews from client

Endpoint URL	localhost:3000/api/clients/{id}/reviews
HTTP Method	GET
Authorization	Required BASIC token

Description:

Retorna todas las reseñas que un cliente ha hecho.

Parameters:

∴ {id}: ID del cliente.

Response:

Lista de las reseñas del cliente, con la información básica del lugar sobre el que se hizo la reseña.

```
[
  {
    "text": "Such an amazing place! I would come back for sure",
    "stars": 5,
    "place": {
```

```

        "id": "0p1a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a",
        "name": "Great Place",
        "picture": "...",
        ...
    }
},
{
    ...
}
]

```

HTTP status responses:

- ∴ 200: Se obtienen las reseñas correctamente.
- ∴ 204: Se resuelve el cliente, pero no ha hecho ninguna reseña.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: El usuario con dicho ID no fue encontrado.

■ Get all “pendig” from client

Endpoint URL	localhost:3000/api/clients/{id}/pending
HTTP Method	GET
Authorization	Required CLIENT X token

Description:

Retorna todos los lugares pendientes de un cliente.

Parameters:

- ∴ {id}: ID del cliente.

Response:

Lista con la información principal de todos los lugares pendientes.

```

[
  {
    "id": "0p1a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a",
    "name": "Great Place",

```

```

        "picture": "...",
        ...
    }, {
        ...
    }
]

```

HTTP status responses:

- ∴ 200: Se obtienen los lugares pendientes correctamente.
- ∴ 204: Se resuelve el cliente, pero no tiene pendientes.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: El usuario con dicho ID no fue encontrado.

■ Add place to “pending”

Endpoint URL	localhost:3000/api/clients/{id}/pending/{placeId}
HTTP Method	POST
Authorization	Required CLIENT X token

Description:

Agrega un lugar a la lista de pendientes de un cliente.

Parameters:

- ∴ {id}: ID del cliente.
- ∴ {placeId}: ID del lugar

Response:

Información principal del lugar que acaba de ser agregado.

```

{
  "id": "0p1a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a",
  "name": "Great Place",
  "picture": "...",
  ...
}

```

HTTP status responses:

- ∴ 200: Se agregó el lugar correctamente.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: El usuario con dicho ID no fue encontrado o el lugar con dicho ID no fue encontrado.

■ Remove place from “pending”

Endpoint URL	localhost:3000/api/clients/{id}/pending/{placeId}
HTTP Method	DELETE
Authorization	Required CLIENT X token

Description:

Elimina un lugar de la lista de pendientes de un cliente.

Parameters:

- ∴ {id}: ID del cliente.
- ∴ {placeId}: ID del lugar

Response:

Información principal del lugar que acaba de ser eliminado.

```
{  
  "id": "0p1a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a",  
  "name": "Great Place",  
  "picture": "...",  
  ...  
}
```

HTTP status responses:

- ∴ 204: Se eliminó el lugar correctamente de la lista de pendientes.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: El usuario con dicho ID no fue encontrado o el lugar con dicho ID no fue encontrado.

■ Get all “liked” from client

Endpoint URL	localhost:3000/api/clients/{id}/liked
HTTP Method	GET
Authorization	Required CLIENT X token

Description:

Retorna todos los lugares a los que un cliente dio “like”.

Parameters:

∴ {id}: ID del cliente.

Response:

Lista con la información principal de todos los lugares.

```
[
  {
    "id": "0p2a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a",
    "name": "Very Good Place",
    "picture": "...",
    ...
  }, {
    ...
  }
]
```

HTTP status responses:

- ∴ 200: Se obtienen los lugares correctamente.
- ∴ 204: Se resuelve el cliente, pero no ha dado “likes”.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: El usuario con dicho ID no fue encontrado.

▪ Add place to “liked”

Endpoint URL	localhost:3000/api/clients/{id}/liked/{placeId}
HTTP Method	POST
Authorization	Required CLIENT X token

Description:

Agrega un lugar a lista de lugares “gustados” de un cliente.

Parameters:

- ∴ {id}: ID del cliente.
- ∴ {placeId}: ID del lugar

Response:

No se obtiene cuerpo de respuesta, únicamente un mensaje OK o de error con el mensaje correspondiente.

HTTP status responses:

- ∴ 200: Se agregó el lugar correctamente.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: El usuario con dicho ID no fue encontrado o el lugar con dicho ID no fue encontrado.

▪ Remove place from “liked”

Endpoint URL	localhost:3000/api/clients/{id}/liked/{placeId}
HTTP Method	DELETE
Authorization	Required CLIENT X token

Description:

Elimina un lugar de la lista de “gustados” de un cliente.

Parameters:

- ∴ {id}: ID del cliente.
- ∴ {placeId}: ID del lugar

Response:

No se obtiene cuerpo de respuesta, únicamente un mensaje OK o de error con el mensaje correspondiente.

HTTP status responses:

- ∴ 204: Se eliminó el lugar correctamente de la lista de "gustados".
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: El usuario con dicho ID no fue encontrado o el lugar con dicho ID no fue encontrado.

■ Get all posts from client

Endpoint URL	localhost:3000/api/clients/{id}/posts
HTTP Method	GET
Authorization	Required BASIC token

Description:

Retorna todos los posts que un cliente ha subido.

Parameters:

- ∴ {id}: ID del cliente.

Response:

Lista con la información principal de todos los posts, y la información principal del lugar relacionado a cada post.

```
[
  {
    "id": "0poa0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a",
    "description": "A photo on my favorite place!",
    "visible": true,
    "picture": "...",
    "place": {
      "id": "0p1a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a",
      "name": "Great Place",
      "picture": "...",
      ...
    }
  },
]
```

```
{
    ...
}
]
```

HTTP status responses:

- ∴ 200: Se obtienen los posts correctamente.
- ∴ 204: Se resuelve el cliente, pero el cliente no tiene posts.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: El usuario con dicho ID no fue encontrado.

■ Publish post

Endpoint URL	localhost:3000/api/clients/{id}/posts
HTTP Method	POST
Authorization	Required CLIENT X token

Description:

Publica un post de un cliente.

Parameters:

- ∴ {id}: ID del cliente.

Body:

Toda la información del post a crear, únicamente con el ID del lugar asociado.

```
{
  "id": "0p1a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a",
  "description": "A photo on my favorite place!",
  "visible": true,
  "picture": "...",
  "place": {
    "id": "0p1a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a"
  }
}
```

Response:

Información principal del post que se acaba de publicar, incluyendo la información principal del lugar asociado.

```
{
  "id": "0p1a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a",
  "description": "A photo on my favorite place!",
  "visible": true,
  "picture": "...",
  "place": {
    "id": "0p1a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a",
    "name": "Great Place",
    "picture": "...",
    ...
  }
}
```

HTTP status responses:

- ∴ 201: Se publicó el post correctamente.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: El usuario con dicho ID no fue encontrado o el lugar con el ID dado no fue encontrado.
- ∴ 412: El post a publicar no validó todas las reglas de negocio.

■ Delete post

Endpoint URL	localhost:3000/api/clients/{id}/posts/{postId}
HTTP Method	DELETE
Authorization	Required CLIENT X token

Description:

Elimina un post de un cliente.

Parameters:

- ∴ {id}: ID del cliente.
- ∴ {postId}: ID del post.

Response:

Información del post que se acaba de eliminar.

```
{
  "id": "0p1a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a",
  "description": "A photo on my favorite place!",
  "visible": true,
  "picture": "...",
  "place": {
    "id": "0p1a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a",
    "name": "Great Place",
    "picture": "...",
    ...
  }
}
```

HTTP status responses:

- ∴ 204: Se eliminó el post correctamente.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: El usuario con dicho ID no fue encontrado o el post con dicho ID no fue encontrado.

■ Get all missions from a client

Endpoint URL	localhost:3000/api/clients/{id}/missions
HTTP Method	GET
Authorization	Required CLIENT X token

Description:

Retorna todas las instancias de misiones de un cliente, con su porcentaje de completitud.

Parameters:

- ∴ {id}: ID del cliente.

Response:

Listado de todas las misiones del cliente (de cualquiera de los 3 tipos), junto con su porcentaje de completitud, se muestra también la información principal de los lugares, en caso de haber lugares relacionados a las misiones.

```
[
  {
    "id": "0m1a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a",
    "percentage": 0.75,
    "mission": {
      "id": "0m110a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a",
      "name": "Upload your first 4 posts!",
      "description": "",
      "prizeXp": 5000,
      "base": true,
      "type": "PUBLISH",
      "required": 4,
      "tag": null,
      "places": []
    }
  },
  {
    "id": "0m2a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a",
    "percentage": 0.0,
    "mission": {
      "id": "0m120a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a",
      "name": "Review a restaurant!",
      "description": "",
      "prizeXp": 800,
      "base": false,
      "type": "REVIEW",
      "required": 1,
      "tag": "restaurant",
      "places": []
    }
  },
  {
    "id": "0m3a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a",
    "percentage": 0.0,
    "mission": {
      "id": "0m130a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a",
      "name": "Visit two bars!",
      "description": "",
      "prizeXp": 2000,
      "base": false,
      "type": "VISIT",
```

```

        "required":2,
        "tag":"bar",
        "places":[]
    }
},
{
    "id":"0m4a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a",
    "percentage":0.0,
    "mission":{
        "id":"0mi40a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a",
        "name":"Visit Dakiti bar!",
        "description":"",
        "prizeXp":3000,
        "base":false,
        "type":"VISIT",
        "required":1,
        "tag":null,
        "places":[
            {
                "id":"0p9a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a",
                "name":"Dakiti",
                "picture":"...",
                ...
            }
        ]
    }
},
{
    ...
}
]

```

HTTP status responses:

- ∴ 200: Se obtuvieron todas las misiones del cliente correctamente.
- ∴ 204: Se resolvió el cliente, pero no tiene instancias de misiones.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: El usuario con dicho ID no fue encontrado.

▪ Report location to accomplish missions

Endpoint URL	localhost:3000/api/clients/{id}/missions/report
--------------	---

HTTP Method	POST
Authorization	Required CLIENT X token

Description:

Recibe la ubicación (longitud y latitud) de un cliente y se encarga de identificar en qué lugar está el cliente y con base en esto, actualiza todas las instancias de misiones "visit" relacionadas con este lugar o con alguno de los tags de este lugar.

Parameters:

∴ {id}: ID del cliente.

Body:

La longitud y latitud actual del cliente.

```
{
  "longitude": "...",
  "latitude": "..."
}
```

Response:

Listado de todas las misiones que fueron actualizadas en el proceso.

```
[
  {
    "id": "0m4a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a",
    "percentage": 1.0,
    "mission": {
      "id": "0mi40a0a-0a0a-0a0a-0a0a-0a0a0a0a0a",
      "name": "Visit Dakiti bar!",
      "description": "",
      "type": "VISIT",
      "required": 1,
      "prizeXp": 3000,
      "base": false,
      "tag": null,
      "places": [
        {
```

```

        "id": "0p9a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a",
        "name": "Dakiti",
        "picture": "...",
        ...
    }
]
}
},
{
    ...
}
]

```

HTTP status responses:

- ∴ 200: Se actualizó al menos una misión con la ubicación dada.
- ∴ 204: Se resolvió el cliente, pero no fue actualizada ninguna misión.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: El usuario con dicho ID no fue encontrado.

Business

▪ Get business

Endpoint URL	localhost:3000/api/business/{id}
HTTP Method	GET
Authorization	Required BASIC token

Description:

Retorna toda la información del perfil de un negocio.

Parameters:

- ∴ {id}: ID del negocio.

Response:

Información completa del negocio, incluyendo la información principal de los lugares asociados a dicho negocio.

```
{
  "id": "0b0a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a",
  "name": "Default Business",
  "email": "business@email.com",
  "picture": "...",
  "type": "Restaurant Chain",
  "places": [
    {
      "name": "Default Store 01",
      "picture": "...",
      "address": "...",
      ...
    },
    {
      ...
    }
  ]
}
```

HTTP status responses:

- ∴ 200: Se obtiene el negocio con el ID dado.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: El negocio con dicho ID no fue encontrado.

■ Register business

Endpoint URL	localhost:3000/api/business
HTTP Method	POST
Authorization	Required REGISTERED token

Description:

Recibe la información de registro de un negocio y se encarga de validar el negocio y persistirlo a la base de datos.

Body:

Toda la información del negocio.

```
{
  "name": "Default Business",
  "email": "business@email.com",
  "password": "password",
  "picture": "...",
  "type": "Restaurant Chain"
}
```

Response:

Información del negocio con su ID y con una lista inicialmente vacía de lugares.

```
{
  "id": "0b0a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a",
  "name": "Default Business",
  "email": "business@email.com",
  "picture": "...",
  "type": "Restaurant Chain",
  "places": []
}
```

HTTP status responses:

- ∴ 201: El negocio es registrado correctamente.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 412: El negocio no validó todas las reglas de negocio.

■ Update business

Endpoint URL	localhost:3000/api/business/{id}
HTTP Method	PUT
Authorization	Required BASIC token

Description:

Permite actualizar la información de un negocio.

Parameters:

∴ {id}: ID del negocio.

Body:

La información a actualizar del negocio, no se debe enviar toda la información.

```
{
  "name": "My new business name",
  "email": "business@mybusiness.com",
}
```

Response:

Información completa del negocio, actualizada.

```
{
  "id": "0b0a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a",
  "name": "My new business name",
  "email": "business@mybusiness.com",
  "picture": "...",
  "type": "Restaurant Chain",
  "places": []
}
```

HTTP status responses:

- ∴ 200: El negocio es actualizado correctamente.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 403: Se está intentando modificar información que no se puede modificar, como el ID.
- ∴ 404: El negocio con dicho ID no fue encontrado.
- ∴ 412: El negocio no validó todas las reglas de negocio.

▪ [Admin] Delete business

Endpoint URL	localhost:3000/api/business/{id}
HTTP Method	DELETE

Authorization	Required ADMIN token
---------------	----------------------

Description:

Elimina por completo a un negocio de la aplicación, **y consigo a todos los lugares asociados.**

Parameters:

∴ {id}: ID del negocio.

Response:

No se obtiene cuerpo de respuesta, únicamente un mensaje OK o de error con el mensaje correspondiente.

HTTP status responses:

- ∴ 204: El negocio fue eliminado correctamente.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: El negocio con dicho ID no fue encontrado.

Place

▪ Create a prospect of place

Endpoint URL	localhost:3000/api/places
HTTP Method	POST
Authorization	Required BASIC token

Description:

Recibe la información de un negocio nuevo que se desea crear, y lo persiste con el atributo "prospect" que indica que es un prospecto de lugar, pero aún no ha sido validado por un administrador.

Body:

Toda la información inicial del lugar obligatoriamente debe tener un nombre, imagen, dirección (la longitud y latitud se extraerán internamente en el backend) y una lista de máximo 5 tags iniciales, los demás campos pueden ir vacíos.

Adicionalmente, si el lugar fue creado por un negocio, esto se indicará al momento de crear el lugar.

```
{
  "name": "Default place",
  "picture": "...",
  "address": "...",
  "phone": "+57...",
  "url": "defaultplace.com.co",
  "facebook": "facebook.com/default-place",
  "instagram": "instagram.com/default-place",
  "tiktok": "tiktok.com/@defaultplace",
  "business": {
    "id": "0b0a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a"
  },
  "tags": [
    "restaurant",
    "colorful",
    "eco friendly",
    "pet friendly",
    ...
  ]
}
```

Response:

Información del lugar con su ID, la longitud y latitud, con un promedio por defecto de 0, con una lista de reviews inicialmente vacía, con una lista de eventos inicialmente vacía, con una lista de media inicialmente vacía, con la lista de tags indicados y en caso de que se haya creado por un negocio, asociado al negocio, en caso contrario, sin negocio.

```
{
  "id": "0p0a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a",
  "name": "Default place",
  "picture": "...",
  "address": "...",
  "longitude": "...",
```

```

    "latitude": "...",
    "phone": "+57...",
    "url": "defaultplace.com.co",
    "average": 0,
    "facebook": "facebook.com/default-place",
    "instagram": "instagram.com/default-place",
    "tiktok": "tiktok.com/@defaultplace",
    "prospect": true,
    "reviews": [],
    "events": [],
    "media": [],
    "business": {
      "id": "0b0a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a"
    },
    "tags": [
      "restaurant",
      "colorful",
      "eco friendly",
      "pet friendly",
      ...
    ]
  }
}

```

HTTP status responses:

- ∴ 202: El prospecto de lugar es registrado correctamente, pero todavía no será publicado hasta ser aceptado por un administrador.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 412: El prospecto de lugar no validó todas las reglas de negocio.

▪ [Admin] Validate a prospect of place

Endpoint URL	localhost:3000/api/places/{id}/validate?b={bool}
HTTP Method	PUT
Authorization	Required ADMIN token

Description:

Valida o deniega un prospecto de lugar para convertirlo en un lugar cambiando el atributo "prospect", dejándolo visible a los clientes; o elimina el prospecto, en caso de que sea denegado.

Parameters:

- ∴ {id}: ID del lugar.
- ∴ {bool}: Booleano indicando si sí se valida el lugar (true) o no (false).

Body:

Toda la información por modificar/agregar al lugar al momento de publicarlo ya a los clientes, la información que no se indique en el cuerpo, permanecerá como se haya especificado en el prospecto, si se quiere eliminar algún aspecto del lugar, se debe indicar acá directamente, poniéndose nulo. En caso de que el lugar se vaya a denegar, el cuerpo no es necesario.

```
{
  "name": "This is the real name",
  "phone": "+57...",
  "tags": [
    "restaurant",
    "colorful",
    "eco friendly",
    "baby chairs",
    ...
  ]
}
```

Response:

Información completa del lugar que se acaba de publicar (si es validado).

```
{
  "id": "0p0a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a",
  "name": "This is the real name",
  "picture": "...",
  "address": "...",
  "longitude": "...",
  "latitude": "...",
}
```

```

    "phone":"+57...",
    "url":"defaultplace.com.co",
    "average":0,
    "facebook":"facebook.com/default-place",
    "instagram":"instagram.com/default-place",
    "tiktok":"tiktok.com/@defaultplace",
    "prospect":false,
    "reviews":[],
    "events":[],
    "media":[],
    "business":{
      "id":"0b0a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a"
    },
    "tags":[
      "restaurant",
      "colorful",
      "eco friendly",
      "baby chairs",
      ...
    ]
  }
}

```

HTTP status responses:

- ∴ 200: El prospecto de lugar es validado correctamente.
- ∴ 204: El prospecto de lugar es denegado correctamente.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: El lugar con dicho ID no fue encontrado.
- ∴ 412: El lugar con los datos actualizados no validó todas las reglas de negocio.

▪ [Admin] Create a place

Endpoint URL	localhost:3000/api/places/admin
HTTP Method	POST
Authorization	Required ADMIN token

Description:

Recibe la información de un negocio nuevo que un administrador va a crear, y lo persiste directamente listo para publicarse a los clientes.

Body:

Toda la información inicial del lugar obligatoriamente debe tener un nombre, imagen, dirección (la longitud y latitud se extraerán internamente en el backend) y una lista de tags iniciales, los demás campos pueden ir vacíos, puede tener o no un negocio asociado.

```
{
  "name": "Default place",
  "picture": "...",
  "address": "...",
  "phone": "+57...",
  "url": "defaultplace.com.co",
  "facebook": "facebook.com/default-place",
  "instagram": "instagram.com/default-place",
  "tiktok": "tiktok.com/@defaultplace",
  "tags": [
    "restaurant",
    "colorful",
    "eco friendly",
    "pet friendly",
    ...
  ]
}
```

Response:

Información del lugar con su ID, la longitud y latitud, con un promedio por defecto de 0, con una lista de reviews inicialmente vacía, con una lista de eventos inicialmente vacía, con una lista de media inicialmente vacía, con la lista de tags indicados y un negocio asociado si fue indicado, en caso contrario, sin negocio.

```
{
  "id": "0p0a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a",
  "name": "Default place",
  "picture": "...",
```

```

    "address": "...",
    "longitude": "...",
    "latitude": "...",
    "phone": "+57...",
    "url": "defaultplace.com.co",
    "average": 0,
    "facebook": "facebook.com/default-place",
    "instagram": "instagram.com/default-place",
    "tiktok": "tiktok.com/@defaultplace",
    "prospect": false,
    "reviews": [],
    "events": [],
    "media": [],
    "business": null,
    "tags": [
      "restaurant",
      "colorful",
      "eco friendly",
      "pet friendly",
      ...
    ]
  }
}

```

HTTP status responses:

- ∴ 200: El lugar es registrado correctamente.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 412: El lugar no validó todas las reglas de negocio.

■ Update a place

Endpoint URL	localhost:3000/api/places/{id}
HTTP Method	PUT
Authorization	Required BUSINESS X token

Description:

Permite a un negocio actualizar la información de sus lugares, al igual que a los administradores.

Parameters:

∴ {id}: ID del lugar.

Body:

La información por actualizar del lugar, no se debe enviar toda la información.

```
{
  "name": "Name changed",
  "facebook": "facebook.com/new-updated-place",
}
```

Response:

Información completa del lugar, actualizada.

```
{
  "id": "0p0a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a",
  "name": "Name changed",
  "picture": "...",
  "address": "...",
  "longitude": "...",
  "latitude": "...",
  "phone": "+57...",
  "url": "defaultplace.com.co",
  "average": 0,
  "facebook": "facebook.com/new-updated-place",
  "instagram": "instagram.com/default-place",
  "tiktok": "tiktok.com/@defaultplace",
  "prospect": false,
  "reviews": [],
  "events": [],
  "media": [],
  "business": {
    "id": "0b0a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a"
  },
  "tags": [
    "restaurant",
    "colorful",
    "eco friendly",
    "baby chairs",
    ...
  ]
}
```

HTTP status responses:

- ∴ 200: El lugar es actualizado correctamente.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 403: Se está intentando modificar información que no se puede modificar, como el atributo "prospect", "average", "id", etc.
- ∴ 404: El lugar con dicho ID no fue encontrado.
- ∴ 412: El lugar no validó todas las reglas de negocio.

▪ [Admin] Delete a place

Endpoint URL	localhost:3000/api/places/{id}
HTTP Method	DELETE
Authorization	Required ADMIN token

Description:

Elimina por completo a un lugar de la aplicación.

Parameters:

- ∴ {id}: ID del lugar.

Response:

No se obtiene cuerpo de respuesta, únicamente un mensaje OK o de error con el mensaje correspondiente.

HTTP status responses:

- ∴ 204: El lugar fue eliminado correctamente.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: El lugar con dicho ID no fue encontrado.

▪ Add tag to place

Endpoint URL	localhost:3000/api/places/{id}/tags/{tag}
HTTP Method	POST

Authorization	Required BUSINESS X token
---------------	---------------------------

Description:

Agrega un tag específico a la lista de tags de un lugar, si el tag a agregar no existía, se crea un nuevo tag.

Parameters:

- ∴ {id}: ID del lugar.
- ∴ {tag}: Tag a agregar.

Response:

No se obtiene cuerpo de respuesta, únicamente un mensaje OK o de error con el mensaje correspondiente.

HTTP status responses:

- ∴ 200: Se agregó el tag correctamente.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: El lugar con dicho ID no fue encontrado.

■ Remove tag from place

Endpoint URL	localhost:3000/api/places/{id}/tags/{tag}
HTTP Method	DELETE
Authorization	Required BUSINESS X token

Description:

Remueve un tag específico de la lista de tags de un lugar.

Parameters:

- ∴ {id}: ID del lugar.
- ∴ {tag}: Tag a remover.

Response:

No se obtiene cuerpo de respuesta, únicamente un mensaje OK o de error con el mensaje correspondiente.

HTTP status responses:

- ∴ 204: El tag fue removido correctamente.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: El lugar con dicho ID no fue encontrado o dicho tag no se encuentra asociado a el lugar.

▪ [Admin] Associate a place to a business

Endpoint URL	localhost:3000/api/places/{id}/business/{businessId}
HTTP Method	POST
Authorization	Required ADMIN token

Description:

Asociar un lugar a un negocio.

Parameters:

- ∴ {id}: ID del lugar.
- ∴ {businessId}: ID del negocio a asociar.

Response:

No se obtiene cuerpo de respuesta, únicamente un mensaje OK o de error con el mensaje correspondiente.

HTTP status responses:

- ∴ 200: Se asoció el negocio al lugar correctamente.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: El lugar con dicho ID no fue encontrado o el negocio con dicho ID no fue encontrado.

▪ [Admin] Dissociate a place to a business

Endpoint URL	localhost:3000/api/places/{id}/business
HTTP Method	DELETE
Authorization	Required ADMIN token

Description:

Desasociar un negocio de un lugar.

Parameters:

∴ {id}: ID del lugar.

Response:

No se obtiene cuerpo de respuesta, únicamente un mensaje OK o de error con el mensaje correspondiente.

HTTP status responses:

- ∴ 204: Se desasoció el lugar correctamente.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: El lugar con dicho ID no fue encontrado o el lugar no estaba asociado a ningún negocio.

▪ Search place by keywords

Endpoint URL	localhost:3000/api/places?q={q}
HTTP Method	GET
Authorization	Required BASIC token

Description:

Retorna la información básica de todos los lugares que contengan parcialmente alguna de las palabras clave.

Parameters:

∴ {q}: Palabras clave para la búsqueda de los lugares.

Response:

Información principal de los lugares, junto con la lista de tags, para mostrarlos en una lista sin detalles (tener en cuenta que aquellos que tengan "prospect"=true, no se mostrarán al cliente).

```
[
  {
    "id": "0p0a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a",
    "name": "Default place",
    "picture": "...",
    "address": "...",
    "average": 0,
    "prospect": true,
    "tags": [
      "restaurant",
      "colorful",
      "eco friendly",
      "pet friendly",
      ...
    ]
  },
  {
    ...
  }
]
```

HTTP status responses:

- ∴ 200: Se obtiene la lista de lugares.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404:

■ Get place

Endpoint URL	localhost:3000/api/places/{id}
HTTP Method	GET
Authorization	Required BASIC token

Description:

Retorna la información completa de un lugar.

Parameters:

∴ {id}: ID del lugar.

Response:

Información completa del lugar.

```
{
  "id": "0p0a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a",
  "name": "Default place",
  "picture": "...",
  "address": "...",
  "average": 0,
  "prospect": true,
  "tags": [
    "restaurant",
    "colorful",
    "eco friendly",
    "pet friendly",
    ...
  ]
}
```

HTTP status responses:

- ∴ 200: Se obtiene la lista de lugares.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: El lugar con dicho ID no fue encontrado.

■ Get posts from place

Endpoint URL	localhost:3000/api/places/{id}/posts
HTTP Method	GET
Authorization	Required BASIC token

Description:

Retorna la información completa de todos los posts asociados a un lugar (que todos los usuarios han publicado de ese lugar).

Parameters:

∴ {id}: ID del lugar.

Response:

Lista con la información completa de los posts del lugar, y la información básica del usuario que hizo el post.

```
[
  {
    "id": "0post00a-0a0a-0a0a-0a0a-0a0a0a0a0a0a",
    "description": "My first photo in this place :P",
    "picture": "...",
    "visible": true,
    "client": {
      "id": "0a0a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a",
      "name": "Default Client",
      "description": "This is a Vamo' account for a default client",
      "picture": "...",
      "rank": {
        "name": "wood",
        "level": 0,
        "xpNext": 2500
      }
    }
  },
  {
    ...
  }
]
```

HTTP status responses:

- ∴ 200: Se obtiene la lista de posts del lugar.
- ∴ 204: Se resuelve el lugar, pero ningún cliente ha hecho posts sobre ese lugar.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: El lugar con dicho ID no fue encontrado.

■ Get reviews from place

Endpoint URL	localhost:3000/api/places/{id}/reviews
HTTP Method	GET
Authorization	Required BASIC token

Description:

Retorna la información completa de todos los reviews asociados a un lugar (que todos los usuarios han publicado de ese lugar).

Parameters:

∴ {id}: ID del lugar.

Response:

Lista con la información completa de los reviews del lugar, y la información básica del usuario que hizo el review.

```
[
  {
    "id": "0rev0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a",
    "text": "Such an amazing place! I would come back for sure",
    "stars": 5,
    "client": {
      "id": "0a0a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a",
      "name": "Default Client",
      "description": "This is a Vamo' account for a default client",
      "picture": "...",
      "rank": {
        "name": "wood",
        "level": 0,
        "xpNext": 2500
      }
    }
  },
  {
    ...
  }
]
```

HTTP status responses:

- ∴ 200: Se obtiene la lista de reviews del lugar.
- ∴ 204: Se resuelve el lugar, pero ningún cliente ha hecho reviews sobre ese lugar.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: El lugar con dicho ID no fue encontrado.

■ Publish review to place

Endpoint URL	localhost:3000/api/places/{id}/reviews
HTTP Method	POST
Authorization	Required ANY CLIENT token

Description:

Permite a un cliente crear una nueva reseña sobre un lugar.

Parameters:

- ∴ {id}: ID del lugar.

Body:

La reseña que se va a crear sobre el lugar, con el ID del cliente que está publicando la reseña.

```
{
  "text": "Such an amazing place! I would come back for sure",
  "stars": 5,
  "client": {
    "id": "0a0a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a"
  }
}
```

Response:

No se obtiene cuerpo de respuesta, únicamente un mensaje OK o de error con el mensaje correspondiente.

HTTP status responses:

- ∴ 201: Se creó la review correctamente.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: El lugar con dicho ID no fue encontrado.
- ∴ 412: La review no validó las reglas de negocio.

▪ Update review from place

Endpoint URL	localhost:3000/api/places/{id}/reviews/{reviewId}
HTTP Method	PUT
Authorization	Required CLIENT X token

Description:

Permite a un cliente actualizar una reseña sobre un lugar.

Parameters:

- ∴ {id}: ID del lugar.
- ∴ {reviewId}: ID de la review a actualizar.

Body:

Los atributos que se vayan a modificar del lugar, no se deben incluir todos los atributos.

```
{  
  "stars":4  
}
```

Response:

No se obtiene cuerpo de respuesta, únicamente un mensaje OK o de error con el mensaje correspondiente.

HTTP status responses:

- ∴ 200: Se actualizó la review correctamente.
- ∴ 401: El token dado no posee los permisos para este endpoint.

- ∴ 403: Se está intentando modificar información que no se puede modificar, como el ID.
- ∴ 404: El lugar con dicho ID no fue encontrado o el review con dicho ID no fue encontrado.
- ∴ 412: La review no validó las reglas de negocio.

■ Remove review from place

Endpoint URL	localhost:3000/api/places/{id}/reviews/{reviewId}
HTTP Method	DELETE
Authorization	Required CLIENT X token

Description:

Permite a un cliente eliminar una reseña sobre un lugar.

Parameters:

- ∴ {id}: ID del lugar.
- ∴ {reviewId}: ID de la review a eliminar.

Response:

No se obtiene cuerpo de respuesta, únicamente un mensaje OK o de error con el mensaje correspondiente.

HTTP status responses:

- ∴ 204: Se eliminó la review correctamente.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: El lugar con dicho ID no fue encontrado o el review con dicho ID no fue encontrado.

■ Get events of place

Endpoint URL	localhost:3000/api/places/{id}/events
HTTP Method	GET

Authorization	Required BASIC token
---------------	----------------------

Description:

Retorna la información completa de todos los eventos asociados a un lugar.

Parameters:

∴ {id}: ID del lugar.

Response:

Lista con la información completa de los eventos del lugar.

```
[
  {
    "id": "0e0a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a",
    "title": "My Event",
    "description": "Como here, free food for everyone from 8am to 2pm",
    "type": "Food",
    "startDate": "2022-10-10T00:08:00-05:00",
    "endDate": "2022-10-10T00:14:00-05:00",
    "url": "",
    "poster": "...",
  },
  {
    ...
  }
]
```

HTTP status responses:

- ∴ 200: Se obtiene la lista de eventos del lugar.
- ∴ 204: Se resuelve el lugar, pero no tiene eventos.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: El lugar con dicho ID no fue encontrado.

■ Create event

Endpoint URL	localhost:3000/api/places/{id}/events
--------------	---------------------------------------

HTTP Method	POST
Authorization	Required BUSINESS X token

Description:

Permite a un negocio crear un evento en uno de sus lugares.

Parameters:

∴ {id}: ID del lugar.

Body:

El evento que se va a crear sobre el lugar.

```
{
  "id": "0e0a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a",
  "title": "My Event",
  "description": "Como here, free food for everyone from 8am to 2pm",
  "type": "Food",
  "startDate": "2022-10-10T00:08:00-05:00",
  "endDate": "2022-10-10T00:14:00-05:00",
  "url": "",
  "poster": "..."
```

Response:

No se obtiene cuerpo de respuesta, únicamente un mensaje OK o de error con el mensaje correspondiente.

HTTP status responses:

- ∴ 201: Se creó el evento correctamente.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: El lugar con dicho ID no fue encontrado.
- ∴ 412: El evento no validó las reglas de negocio.

▪ Update event

Endpoint URL	localhost:3000/api/places/{id}/events/{eventId}
--------------	---

HTTP Method	PUT
Authorization	Required BUSINESS X token

Description:

Permite a un negocio actualizar un evento sobre uno de sus lugares.

Parameters:

- ∴ {id}: ID del lugar.
- ∴ {eventId}: ID del evento a actualizar.

Body:

Los atributos que se vayan a modificar del evento, no se deben incluir todos los atributos.

```
{
  "title": "My event modified"
}
```

Response:

No se obtiene cuerpo de respuesta, únicamente un mensaje OK o de error con el mensaje correspondiente.

HTTP status responses:

- ∴ 200: Se actualizó el evento correctamente.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 403: Se está intentando modificar información que no se puede modificar, como el ID.
- ∴ 404: El lugar con dicho ID no fue encontrado o el evento con dicho ID no fue encontrado.
- ∴ 412: El evento no validó las reglas de negocio.

▪ Delete event

Endpoint URL	localhost:3000/api/places/{id}/events/{eventId}
HTTP Method	DELETE
Authorization	Required BUSINESS X token

Description:

Permite a un negocio eliminar un evento sobre su lugar.

Parameters:

- ∴ {id}: ID del lugar.
- ∴ {eventId}: ID del evento a eliminar.

Response:

No se obtiene cuerpo de respuesta, únicamente un mensaje OK o de error con el mensaje correspondiente.

HTTP status responses:

- ∴ 204: Se eliminó el evento correctamente.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: El lugar con dicho ID no fue encontrado o el evento con dicho ID no fue encontrado.

▪ Get photos/videos from place

Endpoint URL	localhost:3000/api/places/{id}/media
HTTP Method	GET
Authorization	Required BASIC token

Description:

Retorna la información completa de todas las fotos/videos de un lugar

Parameters:

∴ {id}: ID del lugar.

Response:

Lista con la información completa de Las fotos/lugares del lugar.

```
[
  {
    "id": "0m0a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a",
    "url": "...",
    "type": "IMAGE"
  },
  {
    "id": "0m1a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a",
    "url": "...",
    "type": "VIDEO"
  },
  {
    ...
  }
]
```

HTTP status responses:

- ∴ 200: Se obtiene la lista de la media del lugar.
- ∴ 204: Se resuelve el lugar, pero no tiene media.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: El lugar con dicho ID no fue encontrado.

■ Add photo/video to place

Endpoint URL	localhost:3000/api/places/{id}/media
HTTP Method	POST
Authorization	Required BUSINESS X token

Description:

Permite a un negocio agregar una foto/video para uno de sus lugares.

Parameters:

∴ {id}: ID del lugar.

Body:

La foto/video que se va a crear sobre el lugar.

```
{  
  "url": "...",  
  "type": "IMAGE"  
}
```

Response:

No se obtiene cuerpo de respuesta, únicamente un mensaje OK o de error con el mensaje correspondiente.

HTTP status responses:

- ∴ 201: Se agregó la media correctamente.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: El lugar con dicho ID no fue encontrado.
- ∴ 412: La media no validó las reglas de negocio.

▪ Delete photo/video from place

Endpoint URL	localhost:3000/api/places/{id}/media/{mediaId}
HTTP Method	DELETE
Authorization	Required BUSINESS X token

Description:

Permite a un negocio eliminar una foto/video sobre su lugar.

Parameters:

- ∴ {id}: ID del lugar.
- ∴ {mediaId}: ID de la foto/video a eliminar.

Response:

No se obtiene cuerpo de respuesta, únicamente un mensaje OK o de error con el mensaje correspondiente.

HTTP status responses:

- ∴ 204: Se eliminó la media correctamente.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: El lugar con dicho ID no fue encontrado o la media con dicho ID no fue encontrado.

Event

▪ Get photos/videos from event

Endpoint URL	localhost:3000/api/events/{id}/media
HTTP Method	GET
Authorization	Required BASIC token

Description:

Retorna la información completa de todas las fotos/videos de un evento.

Parameters:

- ∴ {id}: ID del evento.

Response:

Lista con la información completa de Las fotos/lugares del evento.

```
[
  {
    "id": "0m0a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a",
    "url": "...",
    "type": "IMAGE"
  },
  {
    "id": "0m1a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a",
```

```
        "url": "...",
        "type": "VIDEO"
    },
    {
        ...
    }
]
```

HTTP status responses:

- ∴ 200: Se obtiene la lista de la media del evento.
- ∴ 204: Se resuelve el evento, pero no tiene media.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: El evento con dicho ID no fue encontrado.

■ Add photo/video to event

Endpoint URL	localhost:3000/api/events/{id}/media
HTTP Method	POST
Authorization	Required BUSINESS X token

Description:

Permite a un negocio agregar una foto/video para uno de los eventos de sus lugares.

Parameters:

- ∴ {id}: ID del evento.

Body:

La foto/video que se va a crear sobre el evento.

```
{
  "url": "...",
  "type": "IMAGE"
}
```

Response:

No se obtiene cuerpo de respuesta, únicamente un mensaje OK o de error con el mensaje correspondiente.

HTTP status responses:

- ∴ 201: Se agregó la media correctamente.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: El evento con dicho ID no fue encontrado.
- ∴ 412: La media no validó las reglas de negocio.

▪ Delete photo/video from event

Endpoint URL	localhost:3000/api/events/{id}/media/{mediaId}
HTTP Method	DELETE
Authorization	Required BUSINESS X token

Description:

Permite a un negocio eliminar una foto/video sobre un evento de un lugar.

Parameters:

- ∴ {id}: ID del evento.
- ∴ {mediaId}: ID de la foto/video a eliminar.

Response:

No se obtiene cuerpo de respuesta, únicamente un mensaje OK o de error con el mensaje correspondiente.

HTTP status responses:

- ∴ 204: Se eliminó la media correctamente.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: El evento con dicho ID no fue encontrado o la media con dicho ID no fue encontrado.

MissionClient

- [Admin] Disassociate a mission from all clients that haven't completed it

Endpoint URL	localhost:3000/api/missions/{id}/uncomplete
HTTP Method	DELETE
Authorization	Required ADMIN token

Description:

Elimina todas las instancias de misiones **incompletas** de una misión específica.

Parameters:

∴ {id}: ID de la misión.

Response:

Listado de instancias de misiones eliminadas, con el ID del cliente de la instancia.

```
[
  {
    "id": "0mis0a0a-0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a",
    "percentage": "0.8",
    "client": {
      "id": "0a0a0a0a-0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a"
    }
  },
  ...
]
```

HTTP status responses:

- ∴ 200: Se eliminaron las instancias de misiones correctamente.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: La misión con dicho ID no fue encontrada.

▪ [Admin] Disassociate a mission from a client

Endpoint URL	localhost:3000/api/missions/{id}/clients/{clientId}
HTTP Method	DELETE
Authorization	Required ADMIN token

Description:

Elimina una instancia de misión de un cliente específico, si el cliente tiene dicha instancia de misión.

Parameters:

- ∴ {id}: ID de la misión.
- ∴ {clientId}: ID del cliente.

Response:

No se obtiene cuerpo de respuesta, únicamente un mensaje OK o de error con el mensaje correspondiente.

HTTP status responses:

- ∴ 204: Se eliminó la instancia de misión correctamente.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: La misión con dicho ID no fue encontrada o el cliente con dicho ID no fue encontrado.
- ∴ 412: El cliente no tiene una instancia de dicha misión.

Mission

▪ [Admin] Create a mission

Endpoint URL	localhost:3000/api/missions
HTTP Method	POST
Authorization	Required ADMIN token

Description:

Crea una nueva misión y una instancia de esta misión para todos los clientes registrados.

Body:

La información de la misión por crear.

```
{
  "name": "Upload 5 post in Bahai place",
  "description": "",
  "prizeXp": 3500,
  "base": false,
  "type": "PUBLISH",
  "required": 5,
  "tag": null,
  "places": [
    { "id": "0p0a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a" }
  ]
}
```

Response:

Información de la misión creada, con su ID.

```
{
  "id": "0mis0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a",
  "name": "Upload 5 post in Bahai place",
  "description": "",
  "prizeXp": 3500,
  "base": false,
  "type": "PUBLISH",
  "required": 5,
  "tag": null,
  "places": [
    { "id": "0p0a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a" }
  ]
}
```

HTTP status responses:

- ∴ 201: La misión es creada correctamente junto con todas las instancias de misiones.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 412: La misión no validó todas las reglas de negocio.

▪ [Admin] Update a mission

Endpoint URL	localhost:3000/api/missions/{id}
HTTP Method	PUT
Authorization	Required ADMIN token

Description:

Actualiza la misión y todas las instancias de la misión **que no estén completas ya**.

Parameters:

- ∴ {id}: ID de la misión.

Body:

La información que se va a actualizar de la misión, no se debe incluir toda la información.

```
{  
  "prizeXp":5000  
}
```

Response:

Información de la misión actualizada, junto con la lista de los IDs de los clientes cuya instancia de misión se actualizó.

```
{  
  "id":"0mis0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a",  
  "name":"Upload 5 post in Bahai place",  
  "description":"",  
  "prizeXp":5000,  
  "base":false,
```

```

    "type": "PUBLISH",
    "required": 5,
    "tag": null,
    "places": [
      { "id": "0p0a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a" }
    ],
    "instances": [
      {
        "client": {
          "id": "0a0a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a"
        }
      },
      ...
    ]
  }
}

```

HTTP status responses:

- ∴ 201: La misión es actualizada correctamente junto con todas las instancias de misiones que no estén completas.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 403: Se está intentando modificar información que no se puede modificar, como el ID.
- ∴ 404: La misión con dicho ID no fue encontrada.
- ∴ 412: La misión no validó todas las reglas de negocio.

Rank

▪ Get all ranks

Endpoint URL	localhost:3000/api/ranks
HTTP Method	GET
Authorization	Required BASIC token

Description:

Retorna la información básica de todos los rangos.

Response:

Información de todos los rangos, para mostrarlos en una lista sin detalles.

```
[
  {
    "id": "0r0a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a",
    "name": "Silver",
    "level": 1,
    "xpNext": 15000
  },
  ...
]
```

HTTP status responses:

- ∴ 200: Se obtiene la lista de rangos.
- ∴ 204: No hay ningún rango registrado.
- ∴ 401: El token dado no posee los permisos para este endpoint.

▪ [Admin] Create a rank

Endpoint URL	localhost:3000/api/ranks
HTTP Method	POST
Authorization	Required ADMIN token

Description:

Crea un nuevo rango.

Body:

La información del rango a crear.

```
{
  "name": "Silver",
  "level": 1,
  "xpNext": 15000
}
```

Response:

Información del rango creado.

```
{  
  "name": "Silver",  
  "level": 1,  
  "xpNext": 15000  
}
```

HTTP status responses:

- ∴ 201: El rango es creado correctamente.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 412: El rango no validó todas las reglas de negocio o ya existe un rango con dicho nombre.

▪ [Admin] Update a rank

Endpoint URL	localhost:3000/api/ranks/{name}
HTTP Method	PUT
Authorization	Required ADMIN token

Description:

Permite actualizar un rango.

Parameters:

- ∴ {name}: Nombre del rango.

Body:

Información por modificar del rango, no se debe incluir toda la información.

```
{  
  "name": "Au"  
}
```


Response:

Información completa del rango, actualizada.

```
{  
  "name": "Au",  
  "level": 1,  
  "xpNext": 15000  
}
```

HTTP status responses:

- ∴ 200: El rango es actualizado correctamente.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: El rango con dicho nombre (antiguo) no fue encontrado.
- ∴ 412: El rango no validó todas las reglas de negocio o ya existe un rango con dicho nombre.

▪ [Admin] Delete a rank

Endpoint URL	localhost:3000/api/ranks/{name}
HTTP Method	DELETE
Authorization	Required ADMIN token

Description:

Elimina por completo a un rango de la aplicación.

Parameters:

- ∴ {name}: Nombre del rango.

Response:

No se obtiene cuerpo de respuesta, únicamente un mensaje OK o de error con el mensaje correspondiente.

HTTP status responses:

- ∴ 204: El rango fue eliminado correctamente.

- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: El rango con dicho nombre no fue encontrado.

Tag

▪ Get all tags

Endpoint URL	localhost:3000/api/tags
HTTP Method	GET
Authorization	Required BASIC token

Description:

Retorna todos los tags.

Response:

Lista con los nombres de todos los tags.

<pre>["restaurant", "colorful", ...]</pre>
--

HTTP status responses:

- ∴ 200: Se obtiene el nombre de todos los rangos.
- ∴ 204: No hay ningún tag registrado.
- ∴ 401: El token dado no posee los permisos para este endpoint.

▪ [Admin] Create a tag

Endpoint URL	localhost:3000/api/tags?tag={tag}
HTTP Method	POST
Authorization	Required ADMIN token

Description:

Crea un nuevo tag.

Parameters:

∴ {tag}: Nombre del tag a crear.

Response:

Tag creado.

```
{  
  "tag": "restaurant"  
}
```

HTTP status responses:

- ∴ 201: El tag es creado correctamente.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 412: Ya existe un tag con dicho nombre.

▪ [Admin] Update a tag

Endpoint URL	localhost:3000/api/tags/{prevTag}?tag={tag}
HTTP Method	PUT
Authorization	Required ADMIN token

Description:

Actualiza el nombre de un tag (únicamente se debería utilizar cuando se quiere corregir el nombre de un tag).

Parameters:

- ∴ {prevTag}: Nombre del tag anterior.
- ∴ {tag}: Nuevo nombre del tag.

Response:

Tag actualizado.

```
{  
  "tag": "restaurant"  
}
```

HTTP status responses:

- ∴ 201: El tag es creado correctamente.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: El tag con dicho nombre (antiguo) no fue encontrado.
- ∴ 412: Ya existe un tag con el nuevo nombre.

▪ [Admin] Delete a tag

Endpoint URL	localhost:3000/api/tags/{tag}
HTTP Method	DELETE
Authorization	Required ADMIN token

Description:

Elimina un tag.

Parameters:

- ∴ {tag}: Nombre del tag a eliminar.

Response:

Lista de todos los IDs de los lugares que tenían dicho tag y que ya no lo tienen.

```
[  
  { "id": "0p0a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a" },  
  { "id": "0p1a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a" },  
  ...  
]
```

HTTP status responses:

- ∴ 204: El tag es eliminado correctamente.
- ∴ 401: El token dado no posee los permisos para este endpoint.
- ∴ 404: El tag con el dicho nombre no fue encontrado.

NOTAS de desarrollador

Tipos de Tokens

Existen diferentes tipos de autenticaciones para identificar qué tipo de usuario está accediendo a cierto endpoint, estos son:

- UNREGISTERED: Es el token que indica que el endpoint es accedido desde nuestro frontend, esto es para evitar que otras aplicaciones puedan acceder a nuestra información, es el token de los usuarios no registrados y apenas se van a registrar o iniciar sesión.
- BASIC: Token de un negocio o cliente cualquiera.
- ANY CLIENT: Hace referencia al token de un cliente cualquiera.
- ANY BUSINESS: Hace referencia al token de un negocio cualquiera.
- CLIENT X: Hace referencia al token del cliente específico sobre el cuál se está accediendo/creando/modificando/eliminando el recurso.
por ejemplo: solo el usuario puede eliminar sus propios reviews.
- BUSINESS X: Hace referencia al token del negocio específico sobre el cuál se está accediendo/creando/modificando/eliminando el recurso.
por ejemplo: solo el negocio puede modificar la info de sus lugares.
- ADMIN: Solo lo tenemos nosotros los desarrolladores, admins de la app.

Consideraciones importantes al momento de desarrollar

1. Cuando un usuario ejecute alguna de las siguientes acciones:
 - a. Publicar un post.

- b. Publicar un review.
- c. Reportar que se cumplió una misión (y efectivamente se cumplió).

Se debe actualizar el porcentaje de todas las misiones relacionadas; **adicionalmente**, al actualizar el porcentaje de la misión, se debe revisar si este es el 100%, en este caso, se actualiza el XP del usuario; **adicionalmente**, si el XP y el rango del usuario es suficiente para subir de rango, se debe actualizar el rango.

2. Cuando un usuario ejecute alguna de las siguientes acciones:

- a. Agregar un lugar a "gustados".
- b. Eliminar un lugar de "gustados".
- c. Agregar un lugar a "pendientes".
- d. Eliminar un lugar de "pendientes".
- e. Publicar un review.
- f. Actualizar un review.

Se deben actualizar los pesos de los tags del usuario.

3. Cuando un usuario ejecute alguna de las siguientes acciones:

- a. Publicar un review.
- b. Actualizar un review.
- c. Eliminar un review.

Se debe actualizar el atributo "average" del lugar, con el nuevo promedio de estrellas.

4. Cuando un admin ejecute alguna de las siguientes acciones:

- a. Crear una misión (de cualquier tipo)

Se debe crear una instancia de esa misión para todos los usuarios, con porcentaje 0.0.

5. Cuando un cliente ejecute alguna de las siguientes acciones:

- a. Registrar un cliente

Se deben crear instancias asociadas al cliente, de todas las misiones que tengan el atributo "base"=true.

- 6. Cuando un usuario ejecute alguna de las siguientes acciones:

- a. Agregar un tag a un lugar
- b. Remover un tag de un lugar

Se debe crear también el tag, en caso de no existir ya.

Todo esto se hace a nivel de backend; por esto no existen endpoints para actualizar las misiones ni los pesos.