
Spécifications de l'API

Projet Calc-Road

Rédigé le 30 novembre par :
- Sylvan LE DEUNFF

| | |
|--|-----------|
| Avant-propos | 3 |
| Spécifications du backend | 4 |
| Authentification | 4 |
| Utilisateurs | 6 |
| Lister / chercher des utilisateurs | 6 |
| Créer un utilisateur | 7 |
| Supprimer un utilisateur | 8 |
| Récupérer l'utilisateur courant | 9 |
| Cartes | 10 |
| Lister / chercher des cartes | 10 |
| Récupérer les données d'une carte | 11 |
| Créer une nouvelle carte | 12 |
| Editer les informations d'une carte | 13 |
| Supprimer une carte | 14 |
| Utilisateurs d'une carte | 15 |
| Lister les utilisateurs et leurs autorisations pour une carte donnée | 15 |
| Ajouter un utilisateur à une carte | 16 |
| Modifier les autorisations d'un utilisateur pour une carte | 17 |
| Supprimer un utilisateur d'une carte | 18 |
| Routes | 19 |
| Lister les routes d'une carte | 19 |
| Afficher les informations d'une route | 20 |
| Créer une route sur une carte | 21 |
| Éditer une route | 22 |
| Supprimer une route | 23 |
| Trajets | 23 |
| Lister les trajets d'une carte | 24 |
| Afficher les informations d'un trajet | 25 |
| Créer un trajet sur une carte | 26 |
| Editer un trajet | 26 |
| Supprimer un trajet | 28 |
| Événements | 28 |
| Lister les événements d'une carte | 29 |
| Récupérer les informations concernant un événement | 30 |
| Créer un événement sur une carte donnée | 31 |
| Editer un événement | 31 |
| Supprimer un événement | 33 |
| Simulation | 34 |
| Récupérer l'état d'une simulation | 34 |
| Lancer la simulation pour une carte | 35 |
| Supprimer les données d'une simulation | 36 |
| Récupérer les positions des véhicules d'une simulation | 37 |
| Tests automatique du backend | 38 |
| Commandes pour les tests | 38 |
| Utilisation du Makefile | 38 |
| Format des résultats | 39 |

Avant-propos

Ce document décrit le fonctionnement détaillé du coeur de l'application (API).

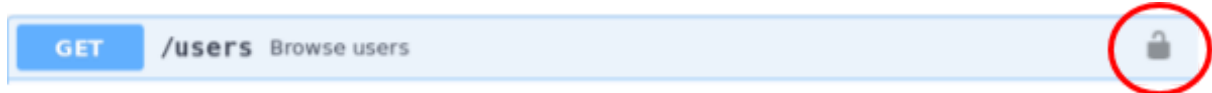
- Dans un premier lieu, il explicite les différents endpoints disponibles à savoir
 - les paramètres GET d'une requête.
 - les informations à envoyer dans le corps d'une requête.
 - La valeur de retour d'une requête.
- La dernière partie décrit le fonctionnement des tests de ces endpoints.

Spécifications du backend

Authentification

La plupart des ressources de l'API sont protégées une clé API (token JWT). Les ressources nécessitant une autorisation sont indiquées par un cadenas dans la documentation.

Exemple de ressource nécessitant une authentification.



Pour pouvoir accéder à une ressource protégée, une requête doit contenir un header "Authorization" composé comme suit :

```
Authorization: Bearer <CLE_API>
```

où

Bearer est le préfixe indiquant le type de token.

<CLE_API> doit être remplacée par la clé API obtenue effectuant une requête sur le endpoint décrit ci-après.

Au préalable, il est nécessaire d'avoir créé un compte utilisateur.

Voir la partie "Utilisateurs" de cette même documentation.

Requête permettant d'obtenir un token d'accès

POST **/auth/token** Obtain access token by sending credentials

Parameters

| Name | Description |
|-------------------------------------|---|
| payload * required (body) | <div>Example Value Model</div> <div>Credentials ∨ { email* string example: jean.dupont@example.com password* string example: dupond }</div> |

Responses Response content type **application/json** ∨

| Code | Description |
|------|--|
| 200 | <div>Success</div> <div>Example Value Model</div> <div>Grant ∨ { access_token* string example: eyJ0eXAi ... 0l_kTAooiW0I refresh_token* string example: eyJ0eXAi ... -SbhhpUMeEf4 expires* integer example: 86400 }</div> |


Si le couple (email, password) correspond bien à un utilisateur enregistré, la réponse contient les paramètres de l'autorisation.

La propriété la plus importante ici est "access_token" (ou API_KEY) qui peut être utilisée pour effectuer des requêtes vers des endpoints protégés.

D'autres informations sont disponibles comme un "refresh_token" qui peut être utilisé pour demander un nouvel "access_token". Ainsi que la durée de validité du token (en secondes).


Utilisateurs


Lister / chercher des utilisateurs

GET **/users** Browse users 

Parameters

| Name | Description |
|------------|-------------|
| query | |
| string | |
| (query) | |
| first_name | |
| string | |
| (query) | |
| last_name | |
| string | |
| (query) | |
| email | |
| string | |
| (query) | |

Responses Response content type **application/json** 

| Code | Description | | | | | | |
|-----------------------|---|--------|-------------|------|-----------------------|-----------|--------|
| 200 | <div><i>Success</i></div> <div>Example Value Model</div> <div> [User > {...}]</div> <div>Headers:</div> <table><thead><tr><th>Name</th><th>Description</th><th>Type</th></tr></thead><tbody><tr><td>Authorization: Bearer</td><td>JWT TOKEN</td><td>string</td></tr></tbody></table> | Name | Description | Type | Authorization: Bearer | JWT TOKEN | string |
| Name | Description | Type | | | | | |
| Authorization: Bearer | JWT TOKEN | string | | | | | |

Créer un utilisateur

POST **/users/account** Create user account

Parameters

| Name | Description |
|-------------------------------------|--|
| payload * required (body) | <div>Example Value Model</div> <div>UserWithPassword ∨ { first_name* string example: Jean last_name* string example: Dupont email* string example: jean.dupont@example.com password* string example: dupond }</div> |

Responses Response content type **application/json** ∨

| Code | Description |
|------|--|
| 200 | <div>Success</div> <div>Example Value Model</div> <div>User ∨ { id integer first_name* string example: Jean last_name* string example: Dupont email* string example: jean.dupont@example.com }</div> |

Supprimer un utilisateur

DELETE

/users/account/{user_id}

Delete user account

Parameters

| Name | Description |
|---------------------------|-------------|
| user_id * required | |
| integer | |
| (path) | |

Responses

Response content type

application/json


| Code | Description |
|------|--|
| 200 | <div>Success</div> <div>Example Value Model</div> <div><div>User</div><div><div><div>id</div><div>integer</div></div><div><div>first_name*</div><div>string</div><div>example: Jean</div></div><div><div>last_name*</div><div>string</div><div>example: Dupont</div></div><div><div>email*</div><div>string</div><div>example: jean.dupont@example.com</div></div></div></div> |

Headers:

| Name | Description | Type |
|-----------------------|-------------|--------|
| Authorization: Bearer | JWT TOKEN | string |

Récupérer l'utilisateur courant

Récupère les informations de l'utilisateur associé au token d'accès.

GET **/users/me** Get authenticated user's details 

Parameters

| Name | Description |
|--|-------------------------|
| X-Fields <code>string(\$mask)</code> <i>(header)</i> | An optional fields mask |


Responses **Response content type** application/json ▼

| Code | Description | | | | | | |
|-----------------------|--|--------|-------------|------|-----------------------|-----------|--------|
| 200 | <div>Success</div> <div>Example Value Model</div> <div><pre>User { id integer first_name* string example: Jean last_name* string example: Dupont email* string example: jean.dupont@example.com }</pre></div> <div>Headers:</div> <table><thead><tr><th>Name</th><th>Description</th><th>Type</th></tr></thead><tbody><tr><td>Authorization: Bearer</td><td>JWT TOKEN</td><td>string</td></tr></tbody></table> | Name | Description | Type | Authorization: Bearer | JWT TOKEN | string |
| Name | Description | Type | | | | | |
| Authorization: Bearer | JWT TOKEN | string | | | | | |

Cartes


Lister / chercher des cartes


Le paramètre “scope” permet de ne lister que les maps pour lesquels l'utilisateur courant possède des droits suffisamment élevé. Il peut prendre les valeurs “read”, “write” et “owner”.

GET **/maps** Browse maps 


Parameters

| Name | Description |
|----------------------------|--|
| title string (query) | |
| scope string (query) | display maps on which current user has required permissions Available values : read, write, owner Default value : read |

Responses Response content type **application/json** 


| Code | Description | | | | | | |
|-----------------------|--|--------|-------------|------|-----------------------|-----------|--------|
| 200 | <div>Success</div> <div>Example Value Model</div> <div>MapSimple  { id integer title* string center* point(\$lat, \$lng) example: List [48.73333, -3.46667] public boolean }</div> <div>Headers:</div> <table><thead><tr><th>Name</th><th>Description</th><th>Type</th></tr></thead><tbody><tr><td>Authorization: Bearer</td><td>JWT TOKEN</td><td>string</td></tr></tbody></table> | Name | Description | Type | Authorization: Bearer | JWT TOKEN | string |
| Name | Description | Type | | | | | |
| Authorization: Bearer | JWT TOKEN | string | | | | | |

Récupérer les données d'une carte

GET `/maps/{map_id}` Retrieve a map's details 


Parameters

| Name | Description |
|--------------------------|-------------|
| map_id * required | |
| integer | |
| (path) | |

Responses Response content type **application/json** 

| Code | Description |
|------|---|
| 200 | <div><i>Success</i></div> <div>Example Value Model</div> <div><pre>Map { id integer title* string center* point(\$lat, \$lng) example: List [48.73333, -3.46667] public boolean journeys > [...] roads > [...] }</pre></div> |
| 401 | <div><i>Unauthorized</i></div> |
| 403 | <div><i>Access forbidden [authorization(s) required: read]</i></div> |
| 404 | <div><i>Map or object not found</i></div> |

Créer une nouvelle carte

POST **/maps** Create a map 


Parameters

| Name | Description |
|-------------------------------------|---|
| payload * required (body) | Example Value Model <div>MapSimple ▾ { id integer title* string center* point(\$lat, \$lng) public example: List [48.73333, -3.46667] boolean }</div> |

Responses **Response content type** **application/json** ▾


| Code | Description | | | | | | |
|-----------------------|--|--------|-------------|------|-----------------------|-----------|--------|
| 200 | <div>Success</div> <div>Example Value Model<div>Map ▾ { id integer title* string center* point(\$lat, \$lng) public example: List [48.73333, -3.46667] boolean journeys > [...] roads > [...] }</div></div> <div>Headers:</div> <table><thead><tr><th>Name</th><th>Description</th><th>Type</th></tr></thead><tbody><tr><td>Authorization: Bearer</td><td>JWT TOKEN</td><td>string</td></tr></tbody></table> | Name | Description | Type | Authorization: Bearer | JWT TOKEN | string |
| Name | Description | Type | | | | | |
| Authorization: Bearer | JWT TOKEN | string | | | | | |

Editer les informations d'une carte

PUT `/maps/{map_id}` Edit a map's details 

Parameters

| Name | Description |
|---|-------------|
| map_id * required integer (path) | |

Responses Response content type `application/json` 

| Code | Description |
|------|---|
| 200 | <div><i>Success</i></div> <div>Example Value Model</div> <div><pre>Map { id integer title* string center* point(\$lat, \$lng) example: List [48.73333, -3.46667] public boolean journeys > [...] roads > [...] }</pre></div> |
| 401 | <div><i>Unauthorized</i></div> |
| 403 | <div><i>Access forbidden [authorization(s) required: owner]</i></div> |
| 404 | <div><i>Map or object not found</i></div> |

Supprimer une carte

DELETE `/maps/{map_id}` Delete a map 

Parameters


| Name | Description |
|---|-------------|
| map_id * required integer (path) | |

Responses Response content type **application/json** 

| Code | Description |
|------|--|
| 401 | <code>Unauthorized</code> |
| 403 | <code>Access forbidden [authorization(s) required: owner]</code> |
| 404 | <code>Map or object not found</code> |

Utilisateurs d'une carte

Lister les utilisateurs et leurs autorisations pour une carte donnée

GET `/maps/{map_id}/users` List user's of a map and their permissions 

Parameters


| Name | Description |
|--------------------------|-------------|
| map_id * required | |
| integer | |
| (path) | |

Responses Response content type **application/json** ▼

| Code | Description |
|------|---|
| 401 | <code>Unauthorized</code> |
| 403 | <code>Access forbidden [authorization(s) required: read]</code> |
| 404 | <code>Map or object not found</code> |

Ajouter un utilisateur à une carte

Cette route est accessible uniquement par les propriétaires de la carte.

POST `/maps/{map_id}/users` Add a user to a map with given permissions 

Parameters


| Name | Description |
|--------------------------|-------------|
| map_id * required | |
| integer | |
| (path) | |

Responses Response content type application/json ▼

| Code | Description |
|------|--|
| 401 | <code>Unauthorized</code> |
| 403 | <code>Access forbidden [authorization(s) required: owner]</code> |
| 404 | <code>Map or object not found</code> |

Modifier les autorisations d'un utilisateur pour une carte

Accessible uniquement pour les propriétaires de la carte.

PUT `/maps/{map_id}/users/{user_id}` Edit user permission for a map 

Parameters


| Name | Description |
|---------------------------|-------------|
| map_id * required | |
| integer | |
| (path) | |
| user_id * required | |
| integer | |
| (path) | |

Responses Response content type application/json ▼

| Code | Description |
|------|--|
| 401 | <code>Unauthorized</code> |
| 403 | <code>Access forbidden [authorization(s) required: owner]</code> |
| 404 | <code>Map or object not found</code> |

Supprimer un utilisateur d'une carte

Accessible uniquement au propriétaire de la carte.

DELETE `/maps/{map_id}/users/{user_id}` Delete user permission for a map 

Parameters


| Name | Description |
|--|-------------|
| map_id * required integer (path) | |
| user_id * required integer (path) | |

Responses Response content type application/json ▼

| Code | Description |
|------|--|
| 401 | <i>Unauthorized</i> |
| 403 | <i>Access forbidden [authorization(s) required: owner]</i> |
| 404 | <i>Map or object not found</i> |


Routes


Lister les routes d'une carte

GET `/maps/{map_id}/roads` Get roads for a given map 


Parameters

| Name | Description |
|---|-------------|
| map_id * required integer (path) | |

Responses Response content type **application/json** 

| Code | Description |
|------|--|
| 200 | <div>Success</div> <div>Example Value Model</div> <div> [Road > {...}]</div> |
| 401 | <div>Unauthorized</div> |
| 403 | <div>Access forbidden [authorization(s) required: read]</div> |
| 404 | <div>Map or object not found</div> |

Afficher les informations d'une route

GET `/maps/{map_id}/roads/{road_id}` Get a road 


Parameters

| Name | Description |
|--|-------------|
| map_id * required integer (path) | |
| road_id * required integer (path) | |

Responses Response content type application/json ▼


| Code | Description |
|------|--|
| 200 | <div>Success</div> <div>Example Value Model</div> <div>▼ [Road > {...}]</div> |
| 401 | <div>Unauthorized</div> |
| 403 | <div>Access forbidden [authorization(s) required: read]</div> |
| 404 | <div>Map or object not found</div> |

Créer une route sur une carte

POST `/maps/{map_id}/roads` Create road on a given map 


Parameters

| Name | Description |
|--------------------------|-------------|
| map_id * required | |
| integer | |
| (path) | |

Responses Response content type `application/json` 

| Code | Description |
|------|--|
| 401 | <code>Unauthorized</code> |
| 403 | <code>Access forbidden [authorization(s) required: write]</code> |
| 404 | <code>Map or object not found</code> |

Éditer une route

PUT /maps/{map_id}/roads/{road_id} Edit a road 


Parameters

| Name | Description |
|--|---|
| payload * required (body) | Example Value Model <pre>Road { id integer name* string coordinates* > [...] max_speed* integer lanes* integer }</pre> |
| map_id * required integer (path) | |
| road_id * required integer (path) | |

Responses Response content type **application/json** ▼


| Code | Description |
|------|---|
| 200 | Success Example Value Model <pre>Road { id integer name* string coordinates* > [...] max_speed* integer lanes* integer }</pre> |
| 401 | Unauthorized |
| 403 | Access forbidden [authorization(s) required: write] |
| 404 | Map or object not found |

Supprimer une route

DELETE /maps/{map_id}/roads/{road_id} Delete a road 

Parameters


| Name | Description |
|--|-------------|
| map_id * required integer (path) | |
| road_id * required integer (path) | |

Responses Response content type **application/json** 

| Code | Description |
|------|--|
| 200 | <div>Success</div> <div>Example Value Model</div> <div><pre>Road { id integer name* string coordinates* > [...] max_speed* integer lanes* integer }</pre></div> |
| 401 | <div>Unauthorized</div> |
| 403 | <div>Access forbidden [authorization(s) required: write]</div> |
| 404 | <div>Map or object not found</div> |

Trajets

Lister les trajets d'une carte

GET `/maps/{map_id}/journeys` Get journeys of a map 


Parameters

| Name | Description |
|---|-------------|
| map_id * required integer (path) | |

Responses Response content type `application/json` ▼

| Code | Description |
|------|--|
| 200 | <div>Success</div> <div>Example Value Model</div> <div>▼ [Journey > {...}]</div> |
| 401 | <div>Unauthorized</div> |
| 403 | <div>Access forbidden [authorization(s) required: read]</div> |
| 404 | <div>Map or object not found</div> |

Afficher les informations d'un trajet

GET `/maps/{map_id}/journeys/{journey_id}` Browse journeys of a map 

Parameters

| Name | Description |
|---|-------------|
| map_id * required integer (path) | |
| journey_id * required integer (path) | |

Responses Response content type **application/json** ▾

| Code | Description |
|------|---|
| 200 | <div><i>Success</i></div> <div>Example Value Model</div> <div><pre>Journey ▾ { id integer name integer start_point point(\$lat, \$lng) example: List [48.73333, -3.46667] end_point point(\$lat, \$lng) example: List [48.73333, -3.46667] start_time time(\$hh:mm:ss.millisec) example: 13:31:33.280905 max_vehicules integer }</pre></div> |
| 401 | <div><i>Unauthorized</i></div> |
| 403 | <div><i>Access forbidden [authorization(s) required: read]</i></div> |
| 404 | <div><i>Map or object not found</i></div> |

Créer un trajet sur une carte

POST `/maps/{map_id}/journeys` Create a journey on a given map 


Parameters

| Name | Description |
|---|--|
| payload * required (body) | Example Value Model <pre>Journey { id integer name integer start_point point(\$lat, \$lng) example: List [48.73333, -3.46667] end_point point(\$lat, \$lng) example: List [48.73333, -3.46667] start_time time(\$hh:mm:ss.millisec) example: 13:31:33.280905 max_vehicules integer }</pre> |
| map_id * required integer (path) | |

Responses Response content type **application/json**


| Code | Description |
|------|--|
| 200 | Success Example Value Model <pre>Journey { id integer name integer start_point point(\$lat, \$lng) example: List [48.73333, -3.46667] end_point point(\$lat, \$lng) example: List [48.73333, -3.46667] start_time time(\$hh:mm:ss.millisec) example: 13:31:33.280905 max_vehicules integer }</pre> |
| 401 | Unauthorized |
| 403 | Access forbidden [authorization(s) required: write] |
| 404 | Map or object not found |

Editer un trajet

PUT /maps/{map_id}/journeys/{journey_id} Edit a journey 


Parameters

| Name | Description |
|---|--|
| payload * required (body) | Example Value Model <pre>Journey { id integer name integer start_point point(\$lat, \$lng) example: List [48.73333, -3.46667] end_point point(\$lat, \$lng) example: List [48.73333, -3.46667] start_time time(\$hh:mm:ss.millisec) example: 13:31:33.280905 max_vehicles integer }</pre> |
| map_id * required integer (path) | |
| journey_id * required integer (path) | |

Responses Response content type **application/json** 


| Code | Description |
|------|--|
| 200 | Success Example Value Model <pre>Journey { id integer name integer start_point point(\$lat, \$lng) example: List [48.73333, -3.46667] end_point point(\$lat, \$lng) example: List [48.73333, -3.46667] start_time time(\$hh:mm:ss.millisec) example: 13:31:33.280905 max_vehicles integer }</pre> |
| 401 | Unauthorized |
| 403 | Access forbidden [authorization(s) required: write] |

Supprimer un trajet

DELETE `/maps/{map_id}/journeys/{journey_id}` Delete a journey 

Parameters

| Name | Description |
|---|-------------|
| map_id * required integer (path) | |
| journey_id * required integer (path) | |

Responses Response content type **application/json** 

| Code | Description |
|------|---|
| 200 | <div><i>Success</i></div> <div>Example Value Model</div> <div><pre>Journey { id integer name integer start_point point(\$lat, \$lng) example: List [48.73333, -3.46667] end_point point(\$lat, \$lng) example: List [48.73333, -3.46667] start_time time(\$hh:mm:ss.millisec) example: 13:31:33.280905 max_vehicules integer }</pre></div> |
| 401 | <div><i>Unauthorized</i></div> |
| 403 | <div><i>Access forbidden [authorization(s) required: write]</i></div> |
| 404 | <div><i>Map or object not found</i></div> |

Événements

Lister les évènements d'une carte

GET `/maps/{map_id}/events` List events of a map

Parameters

| Name | Description |
|---|-------------|
| map_id * required integer (path) | |

Responses

Response content type **application/json** ▼

| Code | Description |
|------|--|
| 200 | <div><i>Success</i></div> <div>Example Value Model</div> <div>▼ [Event > {...}]</div> |

Récupérer les informations concernant un évènement

GET `/maps/{map_id}/events/{event_id}` Get event details

Parameters

| Name | Description |
|---|-------------|
| map_id * required integer (path) | |
| event_id * required integer (path) | |

Responses Response content type: **application/json**

| Code | Description |
|------|--|
| 200 | <div><i>Success</i></div> <div>Example Value Model</div> <div><pre>Event { id integer coordinates* point(\$lat, \$lng) example: List [48.73333, -3.46667] start* integer stop* integer speed* integer }</pre></div> |

Créer un évènement sur une carte donnée

POST `/maps/{map_id}/events` Create event on a given map

Parameters

| Name | Description |
|---|--|
| payload * required (body) | <div>Example Value Model</div> <div>Event ∨ { id integer coordinates* point(\$lat, \$lng) example: List [48.73333, -3.46667] start* integer stop* integer speed* integer }</div> |
| map_id * required integer (path) | |

Responses Response content type **application/json** ∨

| Code | Description |
|------|--|
| 200 | <div>Success</div> <div>Example Value Model</div> <div>Event ∨ { id integer coordinates* point(\$lat, \$lng) example: List [48.73333, -3.46667] start* integer stop* integer speed* integer }</div> |

Editer un évènement

PUT `/maps/{map_id}/events/{event_id}` Edit event

Parameters

| Name | Description |
|---|--|
| payload * required (body) | <div>Example Value Model</div> <div>Event ∨ { id integer coordinates* point(\$lat, \$lng) example: List [48.73333, -3.46667] start* integer stop* integer speed* integer }</div> |
| map_id * required integer (path) | |
| event_id * required integer (path) | |

Responses Response content type **application/json** ∨

| Code | Description |
|------|--|
| 200 | <div>Success</div> <div>Example Value Model</div> <div>Event ∨ { id integer coordinates* point(\$lat, \$lng) example: List [48.73333, -3.46667] start* integer stop* integer speed* integer }</div> |

Supprimer un évènement

DELETE `/maps/{map_id}/events/{event_id}` Delete event

Parameters

| Name | Description |
|---|-------------|
| map_id * required integer (path) | |
| event_id * required integer (path) | |

Responses Response content type **application/json** ▾

| Code | Description |
|------|---|
| 200 | <div>Success</div> <div>Example Value Model</div> <div>Event ▾ { id integer coordinates* point(\$lat, \$lng) start* integer stop* integer speed* integer }</div> |

Simulation

Récupérer l'état d'une simulation

Une seule simulation peut être stockée pour une carte donnée. Ce endpoint permet de savoir si la simulation nécessite d'être (re)lancée ou non.

| | | | |
|-------------------|--|---------------------------|---|
| GET | | /maps/{map_id}/simulation | Get information concerning a simulation |
| Parameters | | | |
| Name | | Description | |
| map_id * required | | | |
| integer | | | |
| (path) | | | |
| Responses | | Response content type | application/json |
| Code | | Description | |
| 200 | | Success | |

Lancer la simulation pour une carte

Lance la simulation correspondant à une carte, seulement si nécessaire. C'est à dire si la carte a changé depuis la dernière simulation.

| POST <code>/maps/{map_id}/simulation</code> Run simulation if needed | | | | | | | | | |
|---|----------------|------|-------------|--------------------------|----------------|---------|--|--------|--|
| Parameters | | | | | | | | | |
| <table><thead><tr><th>Name</th><th>Description</th></tr></thead><tbody><tr><td>map_id * required</td><td></td></tr><tr><td>integer</td><td></td></tr><tr><td>(path)</td><td></td></tr></tbody></table> | | Name | Description | map_id * required | | integer | | (path) | |
| Name | Description | | | | | | | | |
| map_id * required | | | | | | | | | |
| integer | | | | | | | | | |
| (path) | | | | | | | | | |
| Responses Response content type <code>application/json</code> ▼ | | | | | | | | | |
| <table><thead><tr><th>Code</th><th>Description</th></tr></thead><tbody><tr><td>200</td><td>Success</td></tr></tbody></table> | | Code | Description | 200 | Success | | | | |
| Code | Description | | | | | | | | |
| 200 | Success | | | | | | | | |

Supprimer les données d'une simulation

DELETE `/maps/{map_id}/simulation` Drop existing simulation data

Parameters

| Name | Description |
|---|-------------|
| map_id * required integer (path) | |

Responses

Response content type **application/json** ▼

| Code | Description |
|------|----------------|
| 200 | Success |

Récupérer les positions des véhicules d'une simulation

Ce endpoint permet de récupérer les positions des véhicules d'une simulation à un temps donné, au format GeoJSON. Afin de réduire le nombre de requêtes, il renvoie une liste de GeoJSON décrivant les positions des véhicules pour l'intervalle de temps [time, time+limit].

GET `/maps/{map_id}/simulation/stream` Get information concerning a simulation

Parameters

| Name | Description |
|---|--|
| time * required integer (query) | begin time of the chunks to retrieve |
| limit integer (query) | number of chunks to receive < 50 Default value : 30 |
| map_id * required integer (path) | |

Responses

Response content type **application/json** ▾

| Code | Description |
|------|----------------|
| 200 | Success |

Tests automatique du backend

Commandes pour les tests

Dans l'API, les tests sont effectués en utilisant le **module python pytest**. Il peut être installé via la commande

```
pip install pytest
```

Les jeux de test sont définis dans le répertoire **tests/** de l'application. Avant de lancer le jeu de test, il faut au préalable avoir installé les **dépendances du projet**.

```
pip install -r requirements/test
```

Puis il faut **configurer l'environnement** de l'application comme étant celui de test.

Commande :

```
export FLASK_ENV=test
```

On peut ensuite **lancer le jeu de test** et analyser les résultats

```
python -m pytest -rA -s -W ignore::DeprecationWarning
```

référence: <https://docs.pytest.org/en/latest/>

Utilisation du Makefile

Vous pouvez également lancer les tests via les commandes définies dans le fichier Makefile.

La commande suivante permet de recréer un environnement virtuel (permet de tester également que les dépendances du projet sont à jour) et de lancer les tests.

```
make test
```

La création d'un nouvel environnement virtuel prenant quelques secondes, nous avons également défini une commande permettant de lancer les tests en utilisant l'environnement déjà existant.

```
make lazy-test
```

Format des résultats

Lors du lancement du jeu de test, le résultat obtenu doit être le suivant.

```
===== PASSES =====
===== short test summary info =====
PASSED tests/test_1_users.py::test_get_token_invalid_credentials
PASSED tests/test_1_users.py::test_create_user_missing_parameters
PASSED tests/test_1_users.py::test_create_user
PASSED tests/test_1_users.py::test_get_token_valid_credentials
PASSED tests/test_1_users.py::test_get_current_user
PASSED tests/test_2_maps.py::test_create_map
PASSED tests/test_2_maps.py::test_list_maps
PASSED tests/test_2_maps.py::test_delete_unexisting_map
PASSED tests/test_2_maps.py::test_get_map
PASSED tests/test_2_maps.py::test_edit_existing_map
PASSED tests/test_2_maps.py::test_delete_existing_map
PASSED tests/test_2_maps.py::test_get_deleted_map
PASSED tests/test_3_roads.py::test_new_map_roads_empty
PASSED tests/test_3_roads.py::test_create_roads
PASSED tests/test_3_roads.py::test_roads_created
PASSED tests/test_3_roads.py::test_edit_road
PASSED tests/test_3_roads.py::test_delete_road
PASSED tests/test_4_journeys.py::test_new_map_journeys_empty
PASSED tests/test_4_journeys.py::test_create_journeys
PASSED tests/test_4_journeys.py::test_journeys_created
PASSED tests/test_4_journeys.py::test_edit_journey
PASSED tests/test_4_journeys.py::test_delete_journey
PASSED tests/test_5_events.py::test_new_map_events_empty
PASSED tests/test_5_events.py::test_create_events
PASSED tests/test_5_events.py::test_events_created
PASSED tests/test_5_events.py::test_edit_event
PASSED tests/test_5_events.py::test_delete_event
===== 27 passed in 6.10s =====
```

En cas d'erreur, des messages détaillés sont écrits dans la sortie d'erreur. Cela permet notamment de faire échouer une pipeline de déploiement continu lorsque les tests ne sont pas validés.