

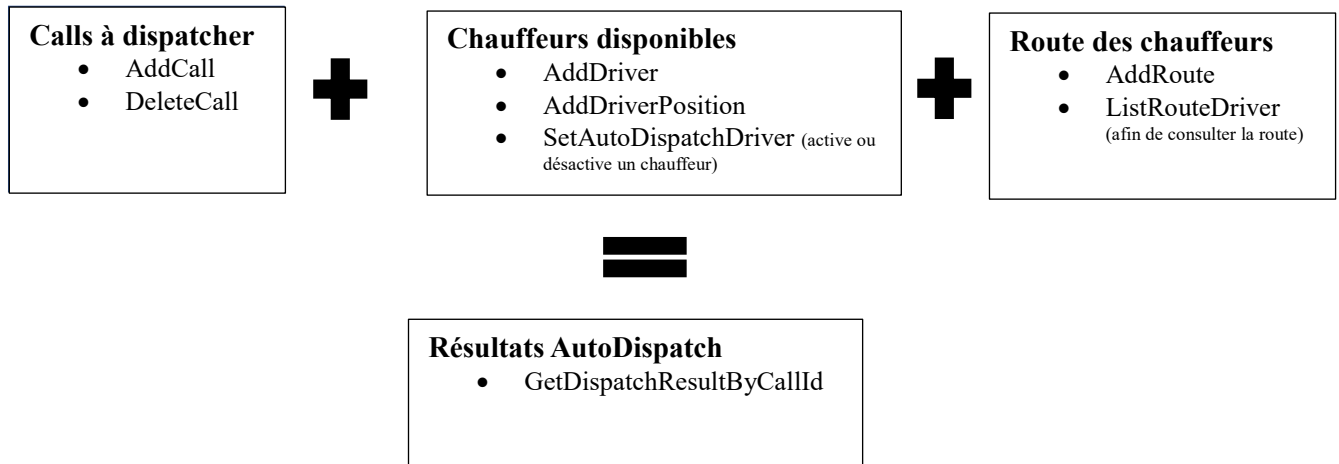
Version du document:	V 1.5
Projet:	Dispatch Courier
Date de la dernière révision:	Jundi 2 mars 2023

Sommaire

1 – Présentation	3
2 - Informations techniques	3
1.1 - Méthodes:	3
1.2 - URLs.....	3
3 - EndPoints.....	4
3.1 – EndPoint (post): v1/DeleteCall	4
3.1.1 - Exemple Requête:	4
3.1.2 – Réponse:.....	4
3.2 – EndPoint (post): v1/AddCall	5
3.2.1 - Exemple Requête:	5
3.2.2 - Réponse:.....	6
3.3 – EndPoint (post): v1/GetDispatchResultByCallId	7
3.2.1 - Exemple Requête:	7
3.3.2 – Réponse:.....	7
3.4 – EndPoint (post): v1/AddDriver	8
3.4.1 - Exemple Requête:	8
3.4.2 - Réponse:.....	9
3.5 – EndPoint (post): v1/SetAutoDispatchDriver	10
3.5.1 - Exemple Requête:	10
3.5.2 – Réponse:.....	10
3.6 – EndPoint (post): v1/AddDriverPosition	11
3.6.1 - Exemple Requête:	11
3.6.2 - Réponse:.....	11
3.7 – EndPoint (post): v1/AddRouteWithCall	12
3.7.1 - Schéma d'objet ListRouteWithCallRequest:	12
3.7.2 - Exemple Requête:	12
3.7.3 - Réponse:.....	15
3.8 – EndPoint (post): v1/ListRouteDriver	16
3.8.1 - Exemple Requête:	16
3.8.2 - Réponse:.....	16
4 - Schémas - Liste des champs	17
4.1 – AddCallRequest	17
4.2 – AddDriverPositionRequest.....	17
4.3 – AddDriverRequest	17
4.4 – AddressRequest.....	18
4.5 – CallIdsResponse	18
4.6 – CallRequest	18
4.7 – ClntBadDriverRequest	19
4.8 – DeleteCallRequest	20
4.9 – DispatchByClientCallIdRequest.....	20
4.10 – DispatchResponse	20
4.11 – DriverAddressExcRequest.....	21
4.12 – DriverCorrRequest	21
4.13 – DriverExcServiceRequest	22
4.14 – DriverIncZoneRequest	22
4.15 – DriverPositionRequest	22
4.16 – DriverRequest.....	24
4.17 – DriverRouteRequest	25
4.18 – DriverZoneServiceExclRequest	25
4.19 – ItensDriverRouteRequest	26
4.20 – ListRouteDriverRequest.....	26
4.21 – ListRouteWithCallRequest	27
4.22 – LoginRequest	27
4.23 – RouteWithCallRequest.....	27
4.24 – RouteDriverResponse	27
4.25 – SetAutoDispatchDriverRequest	28
5 – Postman.....	29
5.1 – Authorization Configuration	29
5.2 – Headers Configuration.....	30

1 – Présentation

Ce document vise à présenter les *Endpoints* de base de l'API DISP Commerce. Cette API permet aux entreprises partenaires de d'obtenir des résultats de Dispatch automatique à travers sa flotte de livreur. Voici un résumé des *Endpoints* selon leurs utilités dans le programme.



Calls à dispatcher : Sert à ajouter tous les appels à répartir et ceux qui se trouvent dans la route des chauffeurs.

Chauffeurs disponibles : Sert à paramétrer chacun des chauffeurs afin d'identifier le meilleur résultat.

Route des chauffeurs : Sert à fournir la route actuelle d'un chauffeur, et ce, en ordre.

Résultats AutoDispatch : Finalement, envoi les résultats des meilleurs chauffeurs pour effectuer cette livraison.

2 - Informations techniques

1.1 - Méthodes:

Get / Post - JSON – CURL

Les données sont encodées en JSON puis envoyées avec la bibliothèque CURL.

La même méthode est utilisée pour les informations de retour.

1.2 - URLs

- URL de développement :

<https://www.soscourrier.ca/apis/apiAutoDispatchDev/v1/>

- URL Swagger : N/A

- URL de production

<https://www.soscourrier.ca/apis/apiAutoDispatch/v1/>

3 - EndPoints

3.1 – EndPoint (post): v1/DeleteCall

Http Method: Post;

Ce terminal(“*endpoint*”) permet au client de **supprimer** un appel de la liste des ramassages et de livraison, supprimant également cet appel de la liste de la route. Fondamentalement, les informations nécessaires seront l'identifiant et le mot de passe du client, ainsi que l'identifiant de l'appelant.

3.1.1 - Exemple Requête:

<https://www.soscourrier.ca/apis/apiAutoDispatchDev/v1/DeleteCall>

JSON - Corps de la Requête:

```
{
  "loginRequest": {
    "username": "DISPDev",
    "password": "DISPDEV@123"
  },
  "clientCallId": "ABC-01"
}
```

3.1.2 – Réponse:

Status 200 Ok: "Ok, Call Number ABC-01 has been removed."

3.2 – EndPoint (post): v1/AddCall

Http Method: Post;

Ce terminal(“endpoint”) vise à permettre au client d'**ajouter** un nouvel appel à la liste des ramassages et livraisons. Fondamentalement, les informations nécessaires seront l'identification et le mot de passe du client, le service choisi, les données sur la collecte (quantité et poids), l'adresse de collecte et l'adresse de livraison.

3.2.1 - Exemple Requête:

<https://www.soscourrier.ca/apis/apiAutoDispatchDev/v1/AddCall>

JSON - Corps de la Requête:

```
{
  "loginRequest": {
    "username": "DISPDev",
    "password": "DISPDEV@123"
  },
  "listCallRequest": [
    {
      "clientCallId": "ABC-01",
      "callCallTime": null,
      "clientId": 6000002,
      "serviceId": 1,
      "readyTime": "2022-12-05T08:00:00",
      "delvMinTime": null,
      "maxTime": null,
      "pickMaxTime": null,
      "delveValTime": null,
      "callPickupTime": null,
      "callDeliveredTime": null,
      "returnTime": null,
      "callEnveloppes": 1,
      "callParcels": 0,
      "callParcelsWeight": 0,
      "callTotalCharge": 0,
      "callRoundTrip": false,
      "vehicule": "A",
      "noBiker": false,
      "addressOrig": {
        "clientAddressId": "1",
        "streetNumber": "1817",
        "streetName": "Des Laurentides",
        "postalCode": "H7M 2P7",
        "city": "Laval",
        "province": "QC",
        "country": "CA"
      },
      "addressDest": {
        "clientAddressId": "2",
        "streetNumber": "2105",
        "streetName": "INDUSTRIEL BOUL.",
        "postalCode": "H7S-1P7",
        "city": "LAVAL",

```

```
        "province": "QC",  
        "country": "CA"  
    }  
  ]  
}
```

3.2.2 - Réponse:

Status 200 Ok: ""

3.3 – EndPoint (post): v1/GetDispatchResultByCallId

Http Method: Post;

Ce terminal(“endpoint”) permettra au client de récupérer les informations de chaque chauffeur en relation avec l'appel. Notez que l'API envoi déjà automatiquement le DispatchResult une fois disponible. Ce terminal est pour une requête additionnelle.

3.2.1 - Exemple Requête:

<https://www.soscourrier.ca/apis/apiAutoDispatchDev/v1/GetDispatchResultByCallId>

JSON - Corps de la Requête:

```
{
  "loginRequest": {
    "username": "DISPDev",
    "password": "DISPDEV@123"
  },
  "listClientCallId": [
    "ABC-01"
  ]
}
```

3.3.2 – Réponse:

Status 200 Success - JSON - Corps de la Réponse:

```
[{
  "id": 153225271,
  "ClientCallId": "ABC-01",
  "callid": 6915488,
  "drvid": 46,
  "dptype": "N4",
  "pattern": "1.2.3.5.6.4.",
  "points2": 11.4099,
  "time1": "2022-12-08T16:51:20.424Z",
  "dpserv": 10,
  "vehicule": "A"
},
{
  "id": 153225272,
  "ClientCallId": "ABC-01",
  "callid": 6915488,
  "drvid": 54,
  "dptype": "N4",
  "pattern": "1.2.3.4.6.7.5.",
  "points2": 16.9413,
  "time1": "2022-12-08T16:51:20.424Z",
  "dpserv": 10,
  "vehicule": "A"
},
]
```

3.4 – EndPoint (post): v1/AddDriver

Http Method: Post;

Ce point de terminaison(endpoint) permet l'inclusion d'un nouveau chauffeur.

3.4.1 - Exemple Requête:

<https://www.soscourrier.ca/apis/apiAutoDispatchDev/v1/AddDriver>

JSON - Corps de la Requête:

```
{
  "loginRequest": {
    "username": "DISPDev",
    "password": "DISPDEV@123"
  },
  "listDriverRequest": [
    {
      "id": 0,
      "nom": "",
      "autoDispatch": true,
      "type": "A",
      "speedPerc": 100,
      "monBeg": "08:00",
      "monEnd": "17:00",
      "tueBeg": "08:00",
      "tueEnd": "17:00",
      "wenBeg": "08:00",
      "wenEnd": "17:00",
      "thuBeg": "08:00",
      "thuEnd": "17:00",
      "friBeg": "08:00",
      "friEnd": "17:00",
      "satBeg": "00:00",
      "satEnd": "00:00",
      "sunBeg": "00:00",
      "sunEnd": "00:00",
      "maxWeight": 0,
      "electrique100": false,
      "addressDriver": {
        "clientAddressId": "",
        "streetNumber": "",
        "streetName": "",
        "postalCode": "",
        "city": "Montreal",
        "province": "QC",
        "country": "CA"
      },
      "listDriverIncZoneRequest": [
        {"Zone": "A01"}, {"Zone": "L01"}, {"Zone": "L02"}, {"Zone": "L03"}, {"Zone": "LOE"},
        {"Zone": "LOW"}, {"Zone": "N01"}, {"Zone": "W01"}, {"Zone": "W02"}, {"Zone": "W03"},
        {"Zone": "W09"}, {"Zone": "E01"}, {"Zone": "E02"}, {"Zone": "E03"}, {"Zone": "C01"},
        {"Zone": "C02"}, {"Zone": "S01"}, {"Zone": "S02"}, {"Zone": "S03"}
      ],
      "listDriverAddressExcRequest": [
```



```
{
  "nociv": "",
  "address": "",
  "room": ""
}
],
"listDriverCorrRequest": [
  {
    "reg1": "",
    "reg2": "",
    "points": 0,
    "pointsNddelv": 0
  }
],
"listClientBadDriverRequest": [
  {
    "cltid": 0,
    "code": "",
    "phone": ""
  }
],
"listDriverExcServiceRequest": [
  {
    "svcid": 0
  }
],
"listDriverZoneServiceExclRequest": [
  {
    "zone": "",
    "svcid": 0
  }
]
]
}
]
```

3.4.2 - Réponse:

Status 200 Ok: "Ok";

3.5 – EndPoint (post): v1/SetAutoDispatchDriver

Http Method: Post;

Ce point de terminaison(“*endpoint*”) permet à l'utilisateur d'activer ou de désactiver le chauffeur, en faisant en sorte que le système "dispatch" utilise ou non ce chauffeur dans le calcul de l'itinéraire.

3.5.1 - Exemple Requête:

<https://www.soscourrier.ca/apis/apiAutoDispatchDev/v1/SetAutoDispatchDriver>

JSON - Corps de la Requête:

```
{
  "loginRequest": {
    "username": "DISPDev",
    "password": "DISPDEV@123"
  },
  "driverId": 123,
  "autoDispatch": true
}
```

3.5.2 – Réponse:

Status 200 Ok: "Ok";

3.6 – EndPoint (post): v1/AddDriverPosition

Http Method: Post;

Ce point de terminaison (“*endpoint*”) doit être utilisé au moins toutes les 14 minutes pour informer le système de la position du chauffeur, permettant ainsi une plus grande précision.

3.6.1 - Exemple Requête:

<https://www.soscourrier.ca/apis/apiAutoDispatchDev/v1/AddDriverPosition>

JSON - Corps de la Requête:

```
{
  "loginRequest": {
    "username": "DISPDev",
    "password": "DISPDEV@123"
  },
  "listDriverPositionRequest": [
    {
      "driverId": "1",
      "recvtime": "2022-12-06T08:03:00",
      "latitude": 45.60035,
      "longitude": -73.725365
    },
    {
      "driverId": "72",
      "recvtime": null,
      "latitude": 45.60035,
      "longitude": -73.725365
    },
    {
      "driverId": "74",
      "recvtime": null,
      "latitude": 45.60035,
      "longitude": -73.725365
    }
  ]
}
```

3.6.2 - Réponse:

Status 200 Ok: "Ok";

3.7 – EndPoint (post): v1/AddRouteWithCall

Http Method: Post;

Ce point de terminaison(“endpoint”) est utilisé pour définir et contrôler l'itinéraire d'un chauffeur. Lorsqu'il y a un changement dans la route du chauffeur, ex : Ramassage, livraison, changement dans l'ordre de sa route, etc, *AddRouteWithCall* doit être utiliser pour mettre à jour les appels et la route du chauffeur.

3.7.1 - Schéma d'objet ListRouteWithCallRequest:

```
ListRouteWithCallRequest{
  loginRequest LoginRequest{...}
  listRouteWithCallRequest [
    RouteWithCallRequest{
      listCallRequestInRoute [ CallRequest{...} ]
      driverRouteRequest DriverRouteRequest{...}
    }
  ]
}
```

3.7.2 - Exemple Requête:

<https://www.soscourrier.ca/apis/apidispcommercedev/v1/ListRouteWithCallRequest>

JSON - Corps de la Requête:

```
{
  "loginRequest": {
    "username": "SOSDEV",
    "password": "SOSDEV@123"
  },
  "listRouteWithCallRequest": [
    {
      "ListCallRequestInRoute": [
        {
          "ClientCallId": "6938817",
          "CallCallTime": "2023-02-28T14:05:13",
          "Billid": "6938817",
          "ClientId": "6001123",
          "ServiceId": "2",
          "ReadyTime": "2023-02-28T14:04:49",
          "DelvMinTime": "2023-02-28T14:04:49",
          "MaxTime": "2023-02-28T16:04:49",
          "PickMaxTime": "2023-02-28T16:04:49",
          "DelveValTime": "2023-02-28T16:04:49",
          "CallPickupTime": "2023-02-28T14:22:14.557",
          "CallDeliveredTime": "2023-03-01T08:56:46.12",
          "ReturnTime": null,
          "CallEnvelopes": "1",

```

```
"CallParcels":0,
"CallParcelsWeight":0,
"CallTotalCharge":44.62,
"CallRoundTrip":false,
"Vehicule":"A",
"NoBiker":false,
"AddressOrig":{
  "ClientAddressId":"6390744",
  "StreetNumber":"620",
  "StreetName":"CATHCART",
  "PostalCode":"H3B 1M1",
  "City":"MONTREAL",
  "Province":"QC",
  "Country":"CA"
},
"AddressDest":{
  "ClientAddressId":"6684551",
  "StreetNumber":"750",
  "StreetName":"CURE-LABELLE BD",
  "PostalCode":"H7V-2T9",
  "City":"CHOMEDEY",
  "Province":"QC",
  "Country":""
}
},
{
  "ClientCallId":"6938790",
  "CallCallTime":"2023-02-28T12:06:47.973",
  "Billid":6938790,
  "ClientId":6011376,
  "ServiceId":3,
  "ReadyTime":"2023-02-28T12:15:00",
  "DelvMinTime":"2023-02-28T12:15:00",
  "MaxTime":"2023-03-01T17:00:00",
  "PickMaxTime":"2023-03-01T17:00:00",
  "DelveValTime":"2023-03-01T17:00:00",
  "CallPickupTime":"2023-02-28T12:41:44.18",
  "CallDeliveredTime":null,
  "ReturnTime":null,
  "CallEnveloppes":1,
  "CallParcels":0,
  "CallParcelsWeight":0,
  "CallTotalCharge":37.35,
  "CallRoundTrip":true,
  "Vehicule":"A",
  "NoBiker":false,
  "NoAutoDp":true,
  "AddressOrig":{
    "ClientAddressId":"6665292",
    "StreetNumber":"2520",
    "StreetName":"DES ENTREPRISES",
    "PostalCode":"J6X-4J8",
    "City":"TERREBONNE",
    "Province":"QC",
    "Country":"CA"
  }
},
```

```
"AddressDest":{
  "ClientAddressId":"6716348",
  "StreetNumber":"1320",
  "StreetName":"GRAHAM BD",
  "PostalCode":"H3P3C8",
  "City":"MONT-ROYAL",
  "Province":"QC",
  "Country":"CA"
}
},
"DriverRouteRequest":
{
  "DriverId":34,
  "ItemsDriverRouteRequest":[
    {
      "RouteId":0,
      "ClientCallId":"0",
      "RouteOperation":4,
      "Instructions":null,
      "Message":"34. [A] MARC JACQUES LABRECH",
      "RouteSchedule":"1899-12-30T00:00:00",
      "ETA":null,
      "addressRequest":null
    },
    {
      "RouteId":1,
      "ClientCallId":"6938817",
      "RouteOperation":0,
      "Instructions":"Dernier Arret",
      "Message":null,
      "RouteSchedule":"2023-02-28T16:04:49",
      "ETA":"2023-02-28T16:01:00",
      "addressRequest":{
        "ClientAddressId":"6684551",
        "StreetNumber":"750",
        "StreetName":"CURE-LABELLE BD",
        "PostalCode":"H7V-2T9",
        "City":"CHOMEDEY",
        "Province":"QC",
        "Country":""
      }
    },
    {
      "RouteId":36445,
      "ClientCallId":"6938790",
      "RouteOperation":2,
      "Instructions":"RETOUR",
      "Message":null,
      "RouteSchedule":"2023-03-01T12:00:00",
      "ETA":null,
      "addressRequest":null
    },
    {
      "RouteId":36455,
      "ClientCallId":"6938795",
      "RouteOperation":2,
```

```
        "Instructions": "",
        "Message": null,
        "RouteSchedule": "2023-02-28T15:31:53",
        "addressRequest": null
    },
    {
        "RouteId": 36580,
        "ClientCallId": "6938865",
        "RouteOperation": 1,
        "Instructions": "",
        "Message": null,
        "RouteSchedule": "2023-03-01T10:30:00",
        "ETA": null,
        "addressRequest": null
    }
]
}
```

3.7.3 - Réponse:

Status 200 Ok: "Ok";

3.8 – EndPoint (post): v1/ListRouteDriver

Http Method: Post;

Ce point de terminaison(“EndPoint”) est utilisé pour lister la route d'un chauffeur spécifique ou de tous les conducteurs.

3.8.1 - Exemple Requête:

<https://www.soscourrier.ca/apis/apiAutoDispatchDev/v/ListRouteDriver>

JSON - Corps de la Requête:

```
{
  "loginRequest": {
    "username": "DISPDev",
    "password": "DISPDEV@123"
  },
  "driverId": 123
}
```

3.8.2 - Réponse:

Status 200 Success - JSON - Corps de la Réponse :

```
{
  "driverID": 0,
  "routeID": 0,
  "addressID": 0,
  "clientCallId": 0,
  "callId": 0,
  "routeOperation": 0,
  "dpserv": 0,
  "address": "string",
  "region": "string",
  "pc": "string",
  "km": 0,
  "delay": 0,
  "instructions": "string",
  "message": "string",
  "done": true,
  "messid": 0,
  "eta": "2022-12-07T10:40:46.257Z",
  "routeSchedule": "2022-12-07T10:40:46.257Z"
}
```


4 - Schémas - Liste des champs

4.1 – AddCallRequest

```
AddCallRequest {
  loginRequest   LoginRequest{...}
  listCallRequest [...]
}
```

Champ/Objet	Type/Classe	Nullable	Noter
loginRequest	LoginRequest{...}		Reportez-vous au schéma <<LoginRequest>>
listCallRequest	[CallRequest{...}]		Reportez-vous au schéma <<CallRequest>>

4.2 – AddDriverPositionRequest

```
AddDriverPositionRequest{
  loginRequest   LoginRequest{...}
  listDriverPositionRequest [...]
}
```

Champ/Objet	Type/Classe	Noter
loginRequest	LoginRequest{..}	Reportez-vous au schéma <<LoginRequest>>
listDriverPositionRequest	DriverPositionRequest[...]	Reportez-vous au schéma <<DriverPositionRequest>>

4.3 – AddDriverRequest

```
AddDriverRequest{
  loginRequest   LoginRequest{...}
  listDriverRequest [...]
}
```

Champ/Objet	Type/Classe	Noter
loginRequest	LoginRequest{...}	Reportez-vous au schéma <<LoginRequest>>
listDriverRequest	DriverRequest[...]	Reportez-vous au schéma <<DriverRequest>>

4.4 – AddressRequest

```

AddressRequest{
  clientAddressId* string
  streetNumber*   string
  streetName*     string
  postalCode*     string
  city*           string
  province*       string
  country*        string
}

```

Champ/Objet	Type/Classe	Nullable	Noter
clientAddressId*	string(32)		numéro d'identification de l'adresse du client.
streetNumber*	string(6)		Numéro civique
streetName*	string(50)		Nom de la rue pour le ramassage ou la livraison. MinLength: 2
postalCode*	string(7)		Postal Code (xxx-xxx). MinLength: 3
city*	string(20)		Nom de la ville. minLength: 2
province*	string(2)		Abréviation de la province (XX). MinLength: 2
country*	string(2)		Abréviation du pays (XX). MinLength: 2

4.5 – CallIdsResponse

```

CallIdsResponse{
  callId      integer($int32)
  clientCallId string - nullable: true
  assigned    string - nullable: true
}

```

4.6 – CallRequest

```

CallRequest{
  clientCallId*   string
  callCallTime   string($date-time) - nullable: true
  billId         integer($int32) - default: 0
  clientId       integer($int32) - default: 0 - nullable: true
  serviceId*    integer($int32) - default: 1
  readyTime      string($date-time) - nullable: true
  delvMinTime    string($date-time) - nullable: true
  maxTime        string($date-time) - nullable: true
  pickMaxTime    string($date-time) nullable: true
  delveValTime*  string($date-time)
  callPickupTime string($date-time) default: null nullable: true
  callDeliveredTime string($date-time) default: null nullable: true
  returnTime     string($date-time) default: null nullable: true
  callEnveloppes* integer($int32) default: 0
  callParcels*   integer($int32) default: 0
  callParcelsWeight* integer($int32) default: 0
  callTotalCharge* number($double) default: 0
  callRoundTrip* boolean default: false
}

```

```

    vehicule*           string default: A
    noBiker             boolean default: false
    addressOrig         AddressRequest{...}
    addressDest         AddressRequest{...}
  }

```

Champ/Objet	Type/Classe	Nullable	Noter
clientCallId*	string(32)		Numéro d'identification du client - minLength: 1
callCallTime	string(\$date-time)	True	Date et heure de création de l'appel
billid	integer(\$int32)		Numéro d'identification du bill
clientId	integer(\$int32)	true	Numéro d'identification client interne(auto-créé par API), pour l'utilisation de l'API et le service de dispatch
serviceId*	integer(\$int32)		Numéro d'identification du service, selon paramètre client.
readyTime	string(\$date-time)	true	Date et heure auxquelles les marchandises seront prêtes pour le ramassage
delvMinTime	string(\$date-time)	true	Date et heure minimale de livraison (plage de livraison)
maxTime	string(\$date-time)	true	Date et heure maximale de livraison (plage de livraison)
pickMaxTime	string(\$date-time)	true	Date et heure maximale de Pick-up
delveValTime*	string(\$date-time)	true	la date et l'heure maximale de livraison
callPickupTime	string(\$date-time)	true	la date et l'heure de ramassage
callDeliveredTime	string(\$date-time)	true	la date et l'heure de livraison
callDeliveredTime	string(\$date-time)	true	la date et l'heure de retour
callEnveloppes*	integer(\$int32)		Nombre d'enveloppes
callParcels*	integer(\$int32)		Nombre de paquets
callParcelsWeight*	integer(\$int32)		Poids du paquets
callTotalCharge*	number(\$double)		Prix total
callRoundTrip*	boolean		default:false
vehicule*	string(1)		Véhicule à utiliser : A (auto), B (vélo), C (camionnette), T (Transport); default: A
noBiker	boolean		default: false
noAutoDp	boolean		default: false
addressOrig	AddressRequest{...}		Reportez-vous au schéma <<AddressRequest>>
addressDest	AddressRequest{...}		Reportez-vous au schéma <<AddressRequest>>

4.7 – ClntBadDriverRequest

Exclus un chauffeur pour un client en particulier. Idéal pour les messagers à problème.

```

ClntBadDriverRequest{
    cltid    integer($int32) - default: 0
    code    string - default: B - nullable: true
}

```

Champ/Objet	Type/Classe	Nullable	Noter
Cltid	integer(\$int32)		numéro d'identification du client. Default: 0
Code	string()	true	"B" : Chauffeur est barré de ce clientId

4.8 – DeleteCallRequest

```

DeleteCallRequest{
    loginRequest    LoginRequest{...}
    clientCallId   string - nullable: true
}
  
```

Champ/Objet	Type/Classe	Nullable	Noter
loginRequest	LoginRequest{...}		Reportez-vous au schéma <<LoginRequest>>
clientCallId	string		numéro d'identification de l'appelant du client utilisateur de l'API. Nullable: true

4.9 – DispatchByClientCallIdRequest

```

DispatchByClientCallIdRequest{
    loginRequest    LoginRequest{...}
    listClientCallId [...]
}
  
```

Champ/Objet	Type/Classe	Nullable	Noter
loginRequest	LoginRequest{...}		Reportez-vous au schéma <<LoginRequest>>
listClientCallId	string[...]		liste ou tableau de type string avec le numéro d'identification de l'appelant du client utilisateur de l'API.

4.10 – DispatchResponse

Résultats finaux de l'AutoDispatch. Il est automatiquement envoyé par l'API lorsque disponible. Il est également possible de faire une requête à l'API via le Endpoint *GetDispatchResultByCallId*.

```

DispatchResponse{
    id                integer($int32)
    clientCallId     string - nullable: true
    callId           integer($int32)
    drvid            integer($int32)
    dptype           string - nullable: true
    pattern          string - nullable: true
    points2          number($double) - nullable: true
    time1            string($date-time) - nullable: true
    dpserv           integer($int32) - nullable: true
    vehicule         string - nullable: true
}
  
```

Champ/Objet	Type/Classe	Nullable	Noter
Id	integer(\$int32)		numéro séquentiel d'identification d'enregistrement de table
clientCallId	string(32)	true	numéro d'identification de l'appelant du client utilisateur de l'API.
CallId	integer(\$int32)		numéro d'identification de l'appelant du client utilisateur de l'API.
Drvid	integer(\$int32)		numéro d'identification du chauffeur
dptype	String(4)	true	numéro interne(auto-créé par API), pour contrôler les étapes
pattern	String(120)	true	Ordre de la nouvelle route créée par l'AutoDispatch. Le 2 chiffres les plus hauts sont l'emplacements de l'origine et de la destination du CallId analysé. Ex : 1,2,10,3,4,5,11,6,7,8,9
points2	number(\$double) true		valeur de score générée par le système qui indique le meilleur choix. Plus l'option est élevée, mieux c'est.
time1	string(\$date-time)	true	la date et l'heure de génération de l'enregistrement
dperv	integer(\$int32)	true	numéro interne(auto-créé par API), pour contrôler le service responsable du traitement
vehicule	String(1)	true	Véhicule à utiliser : A (auto), B (vélo), C (camionnette), T (Transport);

4.11 – DriverAddressExcRequest

Exclus un chauffeur pour une adresse en particulier

```
DriverAddressExcRequest{
    nociv string - nullable: true
    address string - nullable: true
    room string - nullable: true
}
```

Champ/Objet	Type/Classe	Nullable	Noter
Nociv	string()	true	Numéro civique . Default:""
address	string	true	Nom de la rue. Default:""
Room	string	true	numéro d'appartement ou de suite. Default:""

4.12 – DriverCorrRequest

Augmente le pointage d'un chauffeur pour 2 zones (origine et destination). Idéal pour des chauffeurs qui sont souvent choisis dans un secteur en particulier. Suggestion de points à utiliser : entre 3 et 5.

```
DriverCorrRequest{
    reg1 string - nullable: true
    reg2 string - nullable: true
    points number($double) - default: 0 - nullable: true
    pointsNddelv number($double) - default: 0 - nullable: true
}
```

Champ/Objet	Type/Classe	Nullable	Noter
reg1	String(3)	true	Identification de la région initiale qui forme le corridor. Cette identification est formée par trois chiffres/lettres. Default: "" Zone à utiliser : A01,L01,L02,L03, LOE, LOW,N01,W01,W02,W03,W09,E01,E02,E03,C01,C02,S01,S02,S03
reg2	String(3)	true	Identification de la région finale qui forme le corridor. Cette identification est formée par trois chiffres/lettres. Default: "" Zone à utiliser : A01,L01,L02,L03, LOE, LOW,N01,W01,W02,W03,W09,E01,E02,E03,C01,C02,S01,S02,S03
points	number(\$double)	true	Nombre de points bonus pour la région. Default: 0
pointsNddelv	number(\$double)	true	Nom de points bonus pour la région. Default: 0

4.13 – DriverExcServiceRequest

Représente l'exclusion d'un service pour ce chauffeur en particulier.

```
DriverExcServiceRequest{
    svcid    integer($int32) - default: 0
}
```

Champ/Objet	Type/Classe	Nullable	Noter
Svcid	integer(\$int32)	true	numéro d'identification du service à exclure pour ce chauffeur. Default: 0

4.14 – DriverIncZoneRequest

Élément important à l'AutoDispatch. Représente les zones incluses pour ce chauffeur. Si aucune zone n'est inscrite, chauffeur sera automatiquement exclus de tous les choix.

```
DriverIncZoneRequest{
    zone    string - nullable: true
}
```

Champ/Objet	Type/Classe	Nullable	Noter
Zone	String(3)	true	Identification de la zone. Zone à utiliser : A01,L01,L02,L03, LOE, LOW,N01,W01,W02,W03,W09,E01,E02,E03,C01,C02,S01,S02,S03

4.15 – DriverPositionRequest

Élément important afin d'obtenir le meilleur résultat possible par l'AutoDispatch.

```
DriverPositionRequest{
    driverId    string - nullable: true
    recvtim    string($date-time) - nullable: true
    latitude    number($double) - default: 0 - nullable: true
    longitude   number($double) - default: null - nullable: true
}
```

Champ/Objet	Type/Classe	Nullable	Noter
driverId	String(32)	true	numéro d'identification du chauffeur. Default: ""
recvtime	string(\$date-time)	true	la date et l'heure auxquelles les coordonnées de la position du chauffeur ont été générées.
latitude	number(\$double)	true	latitude des coordonnées du poste du chauffeur. Default: null
longitude	number(\$double)	true	longitude des coordonnées du poste du chauffeur. Default: null

4.16 – DriverRequest

Établi les informations et détails du chauffeurs.

```

DriverRequest{
    id                integer($int32) -default: 0
    nom               string - nullable: true
    autoDispatch     boolean - default: true - nullable: true
    actif            boolean - default: true - nullable: true
    standby          boolean - default: true - nullable: true
    type             string - default: A - nullable: true
    speedPerc        integer($int32) - default: 100 - nullable: true
    monBeg           string - default: 08:00 - nullable: true
    monEnd           string - default: 17:00 - nullable: true
    tueBeg           string - default: 08:00 - nullable: true
    tueEnd           string - default: 17:00 - nullable: true
    wenBeg           string - default: 08:00 - nullable: true
    wenEnd           string - default: 17:00 - nullable: true
    thuBeg           string - default: 08:00 - nullable: true
    thuEnd           string - default: 17:00 - nullable: true
    friBeg           string - default: 08:00 - nullable: true
    friEnd           string - default: 17:00 - nullable: true
    satBeg           string - default: 00:00 - nullable: true
    satEnd           string - default: 00:00 - nullable: true
    sunBeg           string - default: 00:00 - nullable: true
    sunEnd           string - default: 00:00 - nullable: true
    maxWeight        integer($int32) - default: 0 - nullable: true
    electrique100    boolean - default: false - nullable: true
    addressDriver     AddressRequest{...}
    listDriverIncZoneRequest    [...]
    listDriverAddressExcRequest [...]
    listDriverCorrRequest       [...]
    listClientBadDriverRequest  [...]
    listDriverExcServiceRequest [...]
    listDriverZoneServiceExclRequest [...]
}
    
```

Champ/Objet	Type/Classe	Nullable	Noter
id	integer(\$int32)		Numéro d'identification du chauffeur, défaut: 0
nom	string()	true	défaut:""
autoDispatch	boolean	true	défaut: true
type	string	true	défaut: A
speedPerc	integer(\$int32)	true	défaut: 100
monBeg	string	true	l'heure de début des activités du chauffeur le lundi. Default: 08:00
monEnd	string	true	l'heure de fin des activités du chauffeur le lundi. Default: 17:00
tueBeg	string	true	l'heure de début des activités du chauffeur le mardi. Default: 08:00
tueEnd	string	true	l'heure de fin des activités du chauffeur le mardi. Default: 17:00
wenBeg	string	true	l'heure de début des activités du chauffeur le mercredi. Default: 08:00

wenEnd	string	true	l'heure de fin des activités du chauffeur le mercredi. Default: 17:00
thuBeg	string	true	l'heure de début des activités du chauffeur le jeudi. Default: 08:00
thuEnd	string	true	l'heure de fin des activités du chauffeur le jeudi. Default: 17:00
friBeg	string	true	l'heure de début des activités du chauffeur le vendredi. Default: 08:00
friEnd	string	true	l'heure de fin des activités du chauffeur le vendredi. Default: 17:00
satBeg	string	true	l'heure de début des activités du chauffeur le samedi. Default: 00:00
satEnd	string	true	l'heure de fin des activités du chauffeur le samedi. Default: 00:00
sunBeg	string	true	l'heure de début des activités du chauffeur le dimanche. Default: 00:00
sunEnd	string	true	l'heure de fin des activités du chauffeur le dimanche. Default: 00:00
maxWeight	integer(\$int32)	true	default: 0
electrique100	bit		default: false
addressDriver	AddressRequest{...}		Reportez-vous au schéma <<AddressRequest>>
listDriverExcZoneRequest	DriverExcZoneRequest[...]		Reportez-vous au schéma <<DriverExcZoneRequest>>
listDriverAddressExcRequest	DriverAddressExcRequest[...]		Reportez-vous au schéma <<DriverAddressExcRequest>>
listDriverCorrRequest	DriverCorrRequest[...]		Reportez-vous au schéma <<DriverCorrRequest>>
listClientBadDriverRequest	ClientBadDriverRequest[...]		Reportez-vous au schéma <<ClientBadDriverRequest>>
listDriverExcServiceRequest	DriverExcServiceRequest[...]		Reportez-vous au schéma <<DriverExcServiceRequest>>
listDriverZoneServiceExclRequest	DriverZoneServiceExclRequest[...]		Reportez-vous au schéma <<DriverZoneServiceExclRequest>>
listDriverStatRequest	DriverStatRequest[...]		Reportez-vous au schéma <<DriverStatRequest>>

4.17 – DriverRouteRequest

```
DriverRouteRequest{
    driverId          integer($int32)
    itensDriverRouteRequest [...]
}
```

Champ/Objet	Type/Classe	Noter
driverId	integer(\$int32)	numéro d'identification du chauffeur
itensDriverRouteRequest	itensDriverRouteRequest [...]	Reportez-vous au schéma << itensDriverRouteRequest >>

4.18 – DriverZoneServiceExclRequest

Service à exclure pour une zone en particulier, pour ce chauffeur.

```
DriverZoneServiceExclRequest{
    zone    string - nullable: true
    svcid   integer($int32) - default: 0
}
```

Champ/Objet	Type/Classe	Nullable	Noter
Zone	string	true	Identification de la zone. Zone à utiliser : A01,L01,L02,L03, LOE, LOW,N01,W01,W02,W03,W09,E01,E02,E03,C01,C02,S01,S02,S03
Svcid	integer(\$int32)		numéro d'identification du service à exclure pour cette zone. Default: 0

4.19 – ItensDriverRouteRequest

```

ItensDriverRouteRequest{
    routeld          integer($int32)
    clientCallId    string - nullable: true
    routeOperation  integer($int32)
    instructions     string - nullable: true
    message         string - nullable: true
    routeSchedule   string($date-time) - nullable: true
    eta             string($date-time) - nullable: true
    addressRequest  AddressRequest{...}
}
  
```

Champ/Objet	Type/Classe	Nullable	Noter
routeld	integer(\$int32)		numéro séquentiel de l'ordre d'enlèvement et de livraison du parcours du chauffeur. Default: 0
clientCallId	String(32)	true	numéro d'identification de l'appelant du client
routeOperation	integer(\$int32)		Champ qui identifie le type d'opération à effectuer. Les valeurs suivantes doivent être utilisées : 1 pour le ramassage, 2 pour la livraison et 3 pour le retour. Default: 0
instructions	string	true	Informations sur les instructions du chauffeur. default: null
message	string	true	message au chauffeur. Default: null
routeSchedule	string(\$date-time) -		Date et heure maximale selon le service
eta	string(\$date-time)	true	la date et l'heure estimées. Le champ ETA doit être alimenté uniquement pour Routeld = 1(pour le cas de "dernier arrete")
addressRequest	AddressRequest	true	Adresse de l'élément de routage (nécessaire uniquement pour routeoperation autre que 1,2,3)

4.20 – ListRouteDriverRequest

```

ListRouteDriverRequest{
    loginRequest    LoginRequest{...}
    driverId       integer($int32) - nullable: true
}
  
```

Champ/Objet	Type/Classe	Nullable	Noter
loginRequest	LoginRequest{...}		Reportez-vous au schéma <<LoginRequest>>
driverId	integer(\$int32)	true	numéro d'identification du chauffeur

4.21 – ListRouteWithCallRequest

```

ListRouteWithCallRequest{
    loginRequest          LoginRequest{...}
    listRouteWithCallRequest [...]
}
  
```

Champ/Objet	Type/Classe	Noter
loginRequest	LoginRequest{...}	Reportez-vous au schéma <<LoginRequest>>
listRouteWithCallRequest	RouteWithCallRequest{}	Reportez-vous au schéma <<RouteWithCallReques>>

4.22 – LoginRequest

```

LoginRequest{
    username      string nullable: true
    password      string nullable: true
}
  
```

Champs	Type	Nullable	Noter
username	string(20)		nom d'utilisateur du système
password	string(32)		mot de passe d'accès client

4.23 – RouteWithCallRequest

```

RouteWithCallRequest
{
    listCallRequestInRoute[ nullable: true CallRequest{...}]
    driverRouteRequest      DriverRouteRequest{...}
}
  
```

Champ/Objet	Type/Classe	Noter
listCallRequestInRoute	CallRequest {...}	Reportez-vous au schéma << CallRequest >>
driverRouteRequest	DriverRouteRequest {}	Reportez-vous au schéma << DriverRouteRequest >>

4.24 – RouteDriverResponse

```

RouteDriverResponse{
    driverID      integer($int32) - nullable: true
    routeID      integer($int32) - nullable: true
    addressID     integer($int32) - nullable: true
    clientCallId integer($int32) - nullable: true
    callId        integer($int32) - nullable: true
    routeOperation integer($int32) - nullable: true
    dpserv        integer($int32) - nullable: true
    address       string - nullable: true
    region        string - nullable: true
    pc            string - nullable: true
    km            number($double) - nullable: true
    delay         number($double) - nullable: true
    instructions  string - nullable: true
    message       string - nullable: true
}
  
```

```

    eta          string($date-time) - nullable: true
    routeSchedule string($date-time) - nullable: true
  }

```

Champ/Objet	Type/Classe	Nullable	Noter
driverID	integer(\$int32)	true	numéro d'identification du chauffeur
routeID	integer(\$int32)	true	numéro séquentiel de l'ordre d'enlèvement et de livraison du parcours du chauffeur.
addressID	integer(\$int32)		numéro d'identification de l'adresse d'appel
clientCallId	integer(\$int32)	true	numéro d'identification de l'appelant du client
callId	integer(\$int32)	true	numéro d'identification de l'appelant
routeOperation	integer(\$int32)	true	Champ qui identifie le type d'opération à effectuer. Les valeurs suivantes doivent être utilisées : 1 pour le retrait, 2 pour la livraison et 3 pour le retour.
dpserv	integer(\$int32)	true	nombre d'utilisations internes(auto-créé par API) par le système pour la répartition de la charge de traitement
address	string	true	Nom de la rue pour le ramassage ou la livraison
region	string	true	région de l'adresse
pc	string	true	Postal Code (xxx-xxx)
km	number(\$double)	true	distance en kilomètres.
delay	number(\$double)	true	temps estimé en minutes pour le ramassage ou la livraison
instructions	string	true	Informations sur les instructions du chauffeur
message	string	true	message au chauffeur
eta	string(\$date-time)	true	la date et l'heure estimées
routeSchedule	string(\$date-time)	true	Date et heure maximale selon le service

4.25 – SetAutoDispatchDriverRequest

Active ou désactive un chauffeur pour l'AutoDispatch.

```

SetAutoDispatchDriverRequest{
    login          LoginRequest{...}
    driverId       integer($int32)
    autoDispatch   boolean
}

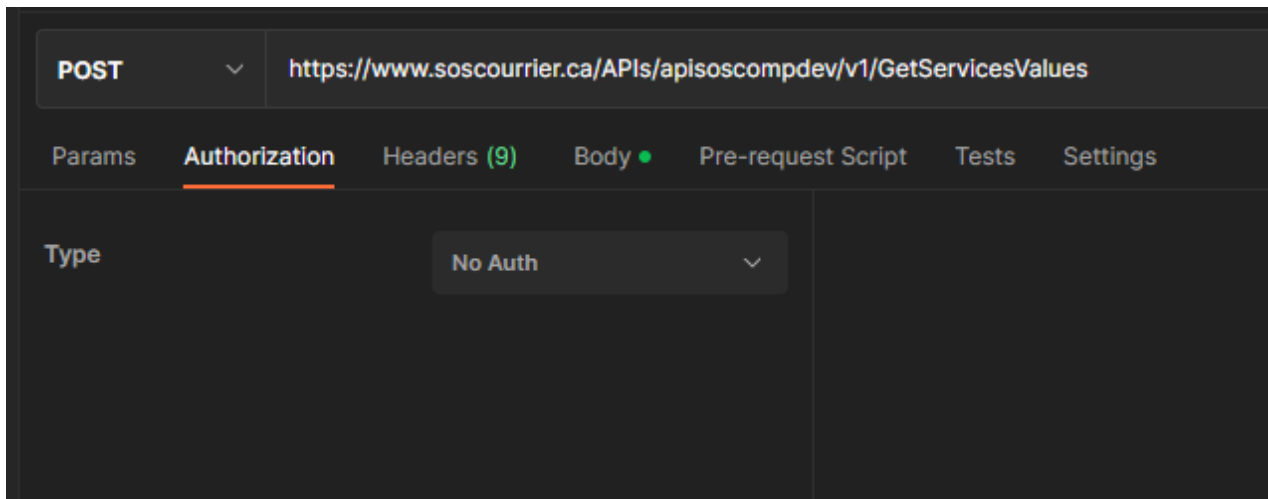
```

Champ/Objet	Type/Classe	Nullable	Noter
Login	LoginRequest{...}		Reportez-vous au schéma <<LoginRequest>>
driverId	integer(\$int32)		numéro d'identification du chauffeur
autoDispatch	boolean		champ permettant d'activer ("true") ou de désactiver ("false") le chauffeur, assurant que le système "dispatch" utilise ou non ce chauffeur dans le calcul de l'itinéraire.

5 – Postman

5.1 – Authorization Configuration

Type = No Auth



5.2 – Headers Configuration

POST ▼ <https://www.soscourrier.ca/APIs/apisoscompdev/v1/GetServicesValues>

Params Authorization **Headers (9)** Body ● Pre-request Script Tests Settings

Headers 🔍 Hide auto-generated headers

	KEY	VALUE
<input checked="" type="checkbox"/>	Postman-Token	<small>ⓘ</small> <calculated when request is sent>
<input checked="" type="checkbox"/>	Content-Type	<small>ⓘ</small> application/json
<input checked="" type="checkbox"/>	Content-Length	<small>ⓘ</small> <calculated when request is sent>
<input checked="" type="checkbox"/>	Host	<small>ⓘ</small> <calculated when request is sent>
<input checked="" type="checkbox"/>	User-Agent	<small>ⓘ</small> PostmanRuntime/7.29.2
<input checked="" type="checkbox"/>	Accept	<small>ⓘ</small> */*
<input checked="" type="checkbox"/>	Accept-Encoding	<small>ⓘ</small> gzip, deflate, br
<input type="checkbox"/>	Connection	<small>ⓘ</small> keep-alive
<input checked="" type="checkbox"/>	Connection	close
	Key	Value