

# API Documentation

Our APIs give third-parties the ability to easily integrate with the OPS-COM platform.

**Operations**Commander (OPS-COM) has made APIs (Application Programming Interface) available that allow for authenticated third-parties to access and manipulate data in OPS-COM as needed. This enables the accomplishment of specific functions, such as the ability to add valid permits from pay-and-display machines.

## What is an API?

APIs, or "Application Programming Interfaces," are specially-designed channels for communication between software systems. Essentially, they give programmers the ability to connect your custom software application to an external system, whether it is running within your office space or out in the cloud.

When you have multiple software systems within your business, it can require a lot of duplicate data entry, which costs you time and money. The API integration project aims to save you money by reducing excess data entry by making it easy for your staff to maintain multiple databases from a single location.

Vendors program APIs into their software to allow external developers to access and manipulate their data. The third-party integration project aims to utilize these systems to eliminate excess data entry and provide new features to the software.

Please note this documentation is specific to APIs we have developed for third-parties to connect to OPS-COM.

We also offer API integration for other third-party hardware and software vendors.

- [API Error Codes](#)
- [Permits API](#)
  - [Pull API: Permit Stats](#)
  - [Push API: Permit Delete](#)
  - [Push API: Permit Create](#)
  - [Push API: Permit Update](#)
- [Push API: Vehicle Create](#)
- [Pull API: UserType](#)

- [Push API: User Create/Update](#)
- [Pull API: Overdue Violations](#)

# API Error Codes

API error codes indicate a failure while communicating with the OPS-COM API.

## Example Error

*Content-Type: application/json*

```
{
  "ErrorCode" : 9001,
  "ErrorMessage" : "API Token is missing from the request."
}
```

## Error Codes

Error Code	Error Message
9000	Client Code is missing from the request.
9001	API Token is missing from the request.
9002	The supplied API token does not have permission to perform that request.
9003	Could not parse the request.
9004	End Time is a required field.
9005	The end time value is invalid.
9006	Plate is a required field.
9007	The start time value is invalid.
9008	Reference ID is required. On a new permit push, a reference id is returned. This is required for updates and deletes.
9009	Record not found.
9010	The Plate Type is required.
9011	The province or state is required.
9012	The vehicle is already in the database.
9013	A unique id is required.
9014	The login source is required.

9015	The login source is invalid.
9016	The user e-mail is required.
9017	The user's first name is required.
9018	The user's last name is required.
9019	The user's email address must be unique.
9020	The user's username must be unique.
9021	The vehicle's plate length exceeds 50 characters.
9022	A record already exists with the supplied details.
9030	The field's maximum number of characters was exceeded.
9031	The field is required.
9032	The field has a minimum number of characters.
9033	The field's value is invalid.

# Permits API

# Pull API: Permit Stats

The OPS-COM Controller provides a simple JSON based API to integrate with. Clients use this API to gather stats on permits pushed into OPS-COM.

Make sure you set the HTTP Content-Type header to be application/json.

## Making API Requests

### Raw Request:

```
POST /api/OC-TOMA/v1/permits/stats HTTP/1.1
Host: controller.operationscommander.io
Accept: application/json
Content-Type: application/json
Cache-Control: no-cache
```

```
{
  "apiToken": "YOUR-API-TOKEN",
  "zones": "NOT-REQUIRED--LIST-OF-ZONES"
}
```

## JavaScript Request

```
var request = new XMLHttpRequest();
```

```
request.open( 'POST' , 'https://controller.operationscommander.io/api/OC-TOMA/v1/permits/stats' );
```

```
request.setRequestHeader( 'Content-Type' , 'application/json' );
request.setRequestHeader( 'Accept' , 'application/json' );
```

```
request.onreadystatechange = function () {
  if ( this.readyState === 4 ) {
    console.log( 'Status:' , this.status);
    console.log( 'Headers:' , this.getAllResponseHeaders());
    console.log( 'Body:' , this.responseText);
  }
};
```

```

var body = {
  "apiToken" : "YOUR-API-TOKEN" ,
  "zones" : "Lot 01,Lot 02"
}
request.send(JSON.stringify(body));

```

# Request Object Attributes

Attribute	Type	Limits	Possible Names	Description
apiToken	String	50-character alphanumeric including dashes	apiToken	<b>(Required)</b> Your supplied API Token.
zones	String	Listed zones match zone names in database	zones	<b>(Not Required)</b> Comma delimited list of zones e.g. zone1,zone2,Lot 03,Red Lot,Street parking

# Successful Response

The response will be a json object. The same reference id will be returned.

*Content-Type: application/json*

```

{
  "data" : [
    {
      "type" : "standard" ,
      "zone" : "zone1" ,
      "total" : "4"
    },
    {
      "type" : "temp" ,
      "zone" : "zone1" ,
      "total" : "35"
    },
  ],
  :
  :
}

```

# Push API: Permit Delete

The OPS-COM Controller provides a simple JSON based API to integrate with. Clients use this API to directly feed details about existing paid permits and their changes into OPS-COM from other systems such as Parking apps.

Make sure you set the HTTP Content-Type header to be application/json.

## Making API Requests

### Raw Request:

```
POST /api/OC-TOMA/v1/permits/delete HTTP/1.1
Host: controller.operationscommander.io
Accept: application/json
Content-Type: application/json
Cache-Control: no-cache

{
  "apiToken": "YOUR-API-TOKEN",
  "referenceID": "PREVIOUS-REFERENCE-ID"
}
```

### JavaScript Request:



```

var request = new XMLHttpRequest();

request.open( 'POST' , 'https://controller.operationscommander.io/api/OC-TOMA/v1/permits/delete ');

request.setRequestHeader( 'Content-Type' , 'application/json' );
request.setRequestHeader( 'Accept' , 'application/json' );

request.onreadystatechange = function () {
  if ( this .readyState === 4) {
    console.log( 'Status:' , this .status);
    console.log( 'Headers:' , this .getAllResponseHeaders());
    console.log( 'Body:' , this .responseText);
  }
};

var body = {
  "apiToken" : "YOUR-API-TOKEN" ,
  "referenceID" : "PREVIOUS-REFERENCE-ID"
}
request.send(JSON.stringify(body));

```

# Request Object Attributes

Attribute	Type	Limits	Possible Names	Description
apiToken	String	50-character alphanumeric including dashes	apiToken	<b>(Required)</b> Your supplied API Token.
Reference ID	String	50-character alphanumeric including dashes	referenceid referenceID reference_id	<b>(Required)</b> This value is supplied to when the permit push api is successful. e.g. 1a9b5375-cb75-4c71-9939-eeae550b09ac

# Successful Response

The response will be a json object. The same reference id will be returned.

*Content-Type: application/json*

```

{
  "status" : "success" ,
  "reference_id" : "1a9b5375-cb75-4c71-9939-eeae550b09ac" ,
  "InternalReferenceID" : "1a9b5375-cb75-4c71-9939-eeae550b09ac"
}

```



# Push API: Permit Create

The OPS-COM Controller provides a simple JSON based API to integrate with. Clients use this API to directly feed paid permit details into OPS-COM from other systems such as Parking Meters. Any permit types whether Validator, Temporary or Standard can be feed to OPS-COM using this API

Make sure you set the HTTP Content-Type header to be application/json.

## Making API Requests

### Raw Request:

```
POST /api/OC-TOMA/v1/permits/push HTTP/1.1
Host: controller.operationscommander.io
Accept: application/json
Content-Type: application/json
Cache-Control: no-cache
```

```
{
  "apiToken": "YOUR-API-TOKEN",
  "Amount": "14.50",
  "CurrencyID": "CAD",
  "LicencePlate": "PL8RDR",
  "zone": "Lot 4",
  "permitNo": "L4-1138",
  "startTime": "2018-07-02T09:00:00",
  "endTime": "2018-07-02T09:30:00"
}
```

### JavaScript Request:

```

var request = new XMLHttpRequest();

request.open( 'POST' , 'https://controller.operationscommander.io/api/OC-TOMA/v1/permits/push ');

request.setRequestHeader( 'Content-Type' , 'application/json' );
request.setRequestHeader( 'Accept' , 'application/json' );

request.onreadystatechange = function () {
  if ( this .readyState === 4) {
    console.log( 'Status:' , this .status);
    console.log( 'Headers:' , this .getAllResponseHeaders());
    console.log( 'Body:' , this .responseText);
  }
};

var body = {
  "apiToken" : "YOUR-API-TOKEN" ,
  "Amount" : "14.50" ,
  "CurrencyID" : "CAD" ,
  "LicencePlate" : "PL8RDR" ,
  "zone" : "Lot 4" ,
  "permitNo" : "L4-1138" ,
  "startTime" : "2018-07-02T09:00:00" ,
  "endTime" : "2018-07-02T09:30:00"
}
request.send(JSON.stringify(body));

```

# Request Object Attributes

Attribute	Type	Limits	Possible Names	Description
apiToken	String	50-character alphanumeric including dashes	apiToken	<b>(Required)</b> Your supplied API Token.
Amount	String	9-character decimal	amount Amount	<i>(Optional)</i> Transaction amount This must contain at least 3 digits, two of which are penny values. The minimum allowable value is \$0.01, and the maximum allowable value is \$999999.99.
Currency	String	10-characters	currency CurrencyID	<i>(Optional)</i> CAD, USD

Attribute	Type	Limits	Possible Names	Description
Start Date	String	20-characters Y-m-d\TH:i:s format.	startTime StartDateUtc	<b>(Required)</b> Must be in the format of Y-m-d\TH:i:s e.g. 2000-05-30T14:38:22 For formatting help, see <a href="#">PHP Date Formatting</a>
End Date	String	20-characters Y-m-d\TH:i:s format.	endTime EndDateUtc	<b>(Required)</b> Must be in the format of Y-m-d\TH:i:s e.g. 2000-05-30T14:38:22 For formatting help, see <a href="#">PHP Date Formatting</a>
License Plate	String	25-characters	plate LicencePlate	<b>(Required)</b> The plate of the vehicle.
Ticket Number	String	50-characters	permitNo TicketNumber	<i>(Optional)</i>
Zone Name	String	200-characters	zone ParkingZoneName	<i>(Optional)</i> If the zone does not match a zone in our system, it will be a disconnected record and may not report properly.

# Successful Response

The response will be a json object.

Content-Type: application/json

```
{
  "status" : "success" ,
  "reference_id" : "1a9b5375-cb75-4c71-9939-eeae550b09ac" ,
  "InternalReferenceID" : "1a9b5375-cb75-4c71-9939-eeae550b09ac"
}
```

# Push API: Permit Update

The OPS-COM Controller provides a simple JSON based API to integrate with. Clients use this API to directly feed paid permit details into OPS-COM from other systems such as Parking Apps.

Make sure you set the HTTP Content-Type header to be application/json.

## Making API Requests

### Raw Request:

```
POST /api/OC-TOMA/v1/permits/update HTTP/1.1
Host: controller.operationscommander.io
Accept: application/json
Content-Type: application/json
Cache-Control: no-cache

{
  "apiToken": "YOUR-API-TOKEN",
  "referenceID": "PREVIOUS-REFERENCE-ID",
  "plate": "PL8RDR",
  "Amount": "14.50",
  "currency": "CAD",
  "endTime": "2018-07-02T09:30:00"
}
```

### JavaScript Request:

```

var request = new XMLHttpRequest();

request.open( 'POST' , 'https://controller.operationscommander.io/api/OC-TOMA/v1/permits/update ');

request.setRequestHeader( 'Content-Type' , 'application/json' );
request.setRequestHeader( 'Accept' , 'application/json' );

request.onreadystatechange = function () {
    if ( this .readyState === 4) {
        console.log( 'Status:' , this .status);
        console.log( 'Headers:' , this .getAllResponseHeaders());
        console.log( 'Body:' , this .responseText);
    }
};

var body = {
    "apiToken" : "YOUR-API-TOKEN" ,
    "referenceID" : "PREVIOUS-REFERENCE-ID" ,
    "plate" : "PL8RDR" ,
    "Amount" : "14.50" ,
    "currency" : "CAD" ,
    "endTime" : "2018-07-02T09:30:00"
}
request.send(JSON.stringify(body));

```

# Request Object Attributes

Attribute	Type	Limits	Possible Names	Description
apiToken	String	50-character alphanumeric including dashes	apiToken	<b>(Required)</b> Your supplied API Token.
Reference ID	String	50-character alphanumeric including dashes	referenceid referenceID reference_id	<b>(Required)</b> This value is supplied to when the permit push api is successful. e.g. 1a9b5375-cb75-4c71-9939-eeae550b09ac
End Date	String	20-characters Y-m-d\TH:i:s format.	endTime EndDateUtc	<i>(Optional)</i> Must be in the format of Y-m-d\TH:i:s e.g. 2000-05-30T14:38:22 For formatting help, see <a href="#">PHP Date Formatting</a>
License Plate	String	25-characters	plate LicencePlate	<i>(Optional)</i> The plate of the vehicle.

Attribute	Type	Limits	Possible Names	Description
Amount	String	9-character decimal	amount Amount	<i>(Optional)</i> Transaction amount This must contain at least 3 digits, two of which are penny values. The minimum allowable value is \$0.01, and the maximum allowable value is \$999999.99.
Currency	String	10-characters	currency CurrencyID	<i>(Optional)</i> CAD, USD

# Successful Response

The response will be a json object. The same reference id will be returned.

Content-Type: application/json

```
{
  "status" : "success" ,
  "reference_id" : "1a9b5375-cb75-4c71-9939-eeae550b09ac" ,
  "InternalReferenceID" : "1a9b5375-cb75-4c71-9939-eeae550b09ac"
}
```



# Push API: Vehicle Create

The OPS-COM Controller provides a simple JSON based API to integrate with. Clients use this API to directly feed vehicles into OPS-COM from other systems.

Make sure you set the HTTP Content-Type header to be application/json.

## Making API Requests

### Raw Request:

```
POST /api/OC_TOMA/v1/vehicles/push HTTP/1.1
Host: controller.operationscommander.io
Accept: application/json
Content-Type: application/json
Cache-Control: no-cache
```

```
{
  "apiToken": "YOUR-API-TOKEN",
  "plate": "PL8RDR",
  "plateType": "Motorcycle",
  "prov": "MA",
  "make": "8",
  "type": "3",
  "colour": "red",
  "year": "2011",
  "vin": "8GKS1AKC7FR518845"
}
```

### JavaScript Request:

```

var request = new XMLHttpRequest();

request.open( 'POST' , 'https://controller.operationscommander.io/api/OC_TOMA/v1/vehicles/push ');

request.setRequestHeader( 'Content-Type' , 'application/json' );
request.setRequestHeader( 'Accept' , 'application/json' );

request.onreadystatechange = function () {
  if ( this .readyState === 4) {
    console.log( 'Status:' , this .status);
    console.log( 'Headers:' , this .getAllResponseHeaders());
    console.log( 'Body:' , this .responseText);
  }
};

var body = {
  "apiToken" : "YOUR-API-TOKEN" ,
  "plate" : "PL8RDR" ,
  "plateType" : "Motorcycle" ,
  "prov" : "MA" ,
  "make" : "kia" ,
  "type" : "commercial" ,
  "colour" : "red" ,
  "year" : "2021" ,
  "vin" : "8GKS1AKC7FR518845"
}
request.send(JSON.stringify(body));

```

# Request Object Attributes

Attribute	Type	Limits	Possible Names	Description
apiToken	String	50-character alphanumeric including dashes.	apiToken	<b>(Required)</b> Your supplied API Token.
plate	String	50-character alphanumeric.	plate	<b>(Required)</b> The license plate.
plateTypeID	String	The ID of the VechiclePlateType record.	plateTypeID	<b>(One of plateTypeID or plateType is required)</b> Your supplied VehiclePlateType identifier.
plateType	String	50-character alphanumeric including dashes.	plateType	<b>(One of plateTypeID or plateType is required)</b> The name of the plate type.

Attribute	Type	Limits	Possible Names	Description
provID	String	The ID of the state/province.	provID	<b>(One of provID or prov is required)</b> Your supplied state or province identifier.
prov	String	50-character alphanumeric including dashes	prov	<b>(One of provID or prov is required)</b> The full name of the state/province or the corresponding postal abbreviation.
makeID	String	The ID of the vehicle make.	makeID	<i>(Optional) Your supplied vehicle make identifier.</i>
make	String	50-character alphanumeric including dashes	make	<i>(Optional) The name of the vehicle manufacturer.</i>
typeID	String	The ID of the vehicle type.	typeID	<i>(Optional) Your supplied vehicle type identifier.</i>
type	String	50-character alphanumeric including dashes	type	<i>(Optional) The name of the type of vehicle that you provided.</i>
colourID	String	The ID of the vehicle colour.	colourID	<i>(Optional) Your supplied vehicle colour identifier.</i>
colour	String	50-character alphanumeric including dashes	colour	<i>(Optional) The name of a colour that you have provided..</i>
year	String	4 digit year.	year	<i>(Optional) The model year.</i>
vin	String	25-characters	vin	<i>(Optional) The vehicle identification number.</i>

# Successful Response

The response will be a json object.

Content-Type: application/json

```
{
  "status" : "success" ,
  "vehicle_id_id" : "158" ,
  "warnings" : [ "The vehicle colour name was too long and has been truncated." ]
}
```

# Pull API: UserType

The OPS-COM Controller provides a simple JSON based API to integrate with. Clients use this API to obtain a list of the current profile user types in their system.

Make sure you set the HTTP Content-Type header to be application/json.

## Making API Requests

### Raw Request:

```
POST /api/OC_TOMA/v1/profiles/types/list HTTP/1.1
Host: controller.operationscommander.io
Accept: application/json
Content-Type: application/json
Cache-Control: no-cache
```

```
{
  "apiToken": "YOUR-API-TOKEN",
}
```

### JavaScript Request:

```
var request = new XMLHttpRequest();

request.open( 'POST' , 'https://controller.operationscommander.io/api/OC_TOMA/v1/profiles/types/list ');

request.setRequestHeader( 'Content-Type' , 'application/json' );
request.setRequestHeader( 'Accept' , 'application/json' );

request.onreadystatechange = function () {
  if ( this .readyState === 4) {
    console.log( 'Status:' , this .status);
    console.log( 'Headers:' , this .getAllResponseHeaders());
    console.log( 'Body:' , this .responseText);
  }
};

var body = {
  "apiToken" : "YOUR-API-TOKEN"
}
request.send(JSON.stringify(body));
```

# Request Object Attributes

Attribute	Type	Limits	Possible Names	Description
apiToken	String	50-character alphanumeric including dashes.	apiToken	<b>(Required)</b> Your supplied API Token.

# Successful Response

The response will be a json object.

Content-Type: application/json

```
{
  "status" : "success" ,
  "user_types" : [
    {
      "id" : "1" ,
      "type_name" : "Full Time Student" ,
      "ext_info" : "Student"
    },
    {
      "id" : "6" ,
      "type_name" : "Demo" ,
      "ext_info" : "Public"
    },
    {
      "id" : "7" ,
      "type_name" : "Full Time Staff" ,
      "ext_info" : "Staff"
    },
    {
      "id" : "8" ,
      "type_name" : "Part Time Staff" ,
      "ext_info" : "Staff"
    },
    {
      "id" : "9" ,
      "type_name" : "Part Time Student" ,
      "ext_info" : "Student"
    },
    {
      "id" : "10" ,
      "type_name" : "Exchange Student" ,
      "ext_info" : "Student"
    },
    {
      "id" : "11" ,
      "type_name" : "Athletics Member" ,
      "ext_info" : "Athletics"
    },
    {
      "id" : "12" ,
      "type_name" : "Complimentary" ,
      "ext_info" : "Public"
    },
    {
      "id" : "13" ,
      "type_name" : "Daily Reserved" ,
      "ext_info" : "Public"
    }
  ]
}
```

# Push API: User Create/Update

The OPS-COM Controller provides a simple JSON based API to integrate with. Clients use this API to push to OPS-COM new users and update existing users directly from another system. For example, you may wish to push Student and Staff information from Banner directly to OPS-COM.

Make sure you set the HTTP Content-Type header to be application/json.

## Making API Requests

### Raw Request:



POST /api/OC\_TOMA/v1/profiles/push HTTP/1.1

Host: controller.operationscommander.io

Accept: application/json

Content-Type: application/json

Cache-Control: no-cache

```
{
  "apiToken": "YOUR-API-TOKEN",
  "unique_id": "tester23",
  "login_source": "OPSCOM",
  "first_name": "firstname",
  "last_name": "lastname",
  "user_name": "username",
  "email": "test@test.com",
  "street": "123 Main Street, 123 Main Street,123 Main Street,123 Main Street,123 Main Street",
  "city": "Everywhere",
  "province": "bc",
  "state": "NY",
  "postal_code": "HOHOHO",
  "zip": "12345-1212",
  "street2": "123 General Street",
  "city2": "Somewhere",
  "province2": "AB",
  "state2": "MA",
  "postal_code2": "A9A9A9",
  "zip2": "54321-1212",
  "phone_cell": "613-555-1212",
  "user_type_id": "6",
  "employ_no": "employee number: default",
  "employee_phone": "emp ph. klondike 555",
  "student_no": "SN 543209854",
  "student_phone": "999",
  "driver_licence_num": "QC 99999999",
  "driver_license_num": "NYNY",
  "date_of_birth": "1901-01-31",
  "locker_user_type_id": "3",
  "driver_licence_prov": "BC",
  "driver_license_state": "CA"
}
```

## JavaScript Request:

```

var request = new XMLHttpRequest();

request.open( 'POST' , 'https://controller.operationscommander.io/api/OC_TOMA/v1/profiles/push ');

request.setRequestHeader( 'Content-Type' , 'application/json' );
request.setRequestHeader( 'Accept' , 'application/json' );

request.onreadystatechange = function () {
    if ( this .readyState === 4) {
        console.log( 'Status:' , this .status);
        console.log( 'Headers:' , this .getAllResponseHeaders());
        console.log( 'Body:' , this .responseText);
    }
};

var body = {
    "apiToken" : "YOUR-API-TOKEN" ,
    "unique_id" : "tester23" ,
    "login_source" : "OPSCOM" ,
    "first_name" : "firstname" ,
    "last_name" : "lastname" ,
    "user_name" : "username" ,
    "email" : "test@test.com" ,
    "street" : "123 Main Street, 123 Main Street,123 Main Street,123 Main Street,123 Main Street,123 Main Street" ,
    "city" : "Everywhere" ,
    "province" : "bc" ,
    "state" : "NY" ,
    "postal_code" : "HOHOHO" ,
    "zip" : "12345-1212" ,
    "street2" : "123 General Street" ,
    "city2" : "Somewhere" ,
    "province2" : "AB" ,
    "state2" : "MA" ,
    "postal_code2" : "A9A9A9" ,
    "zip2" : "54321-1212" ,
    "phone_cell" : "613-555-1212" ,
    "user_type_id" : "6" ,
    "employ_no" : "employee number: default" ,
    "employee_phone" : "emp ph. klondike 555" ,
    "student_no" : "SN 543209854" ,
    "student_phone" : "999" ,
    "driver_licence_num" : "QC 99999999" ,
    "driver_license_num" : "NYYNY" ,
    "date_of_birth" : "1901-01-31" ,
    "locker_user_type_id" : "3" ,
    "driver_licence_prov" : "BC" ,
    "driver_license_state" : "CA"
}
request.send(JSON.stringify(body));

```

# Request Object Attributes

Attribute	Type	Limits	Possible Names	Description
apiToken	String	50-character alphanumeric including dashes.	apiToken	<b>(Required)</b> Your supplied API Token.

Attribute	Type	Limits	Possible Names	Description
Unique ID	String	50-character alphanumeric including dashes.	unique_id	<b>(Required)</b> An unique identification number of the user.
Login Source	String	20-character alphanumeric including dashes.	login_source	<b>(Required)</b> Your supplied login source.
First Name	String	50-character alphanumeric including dashes.	first_name	<i>(Optional)</i> User's first name.
Last Name	String	50-character alphanumeric including dashes.	last_name	<i>(Optional)</i> User's last name.
User Name	String	50-character alphanumeric including dashes.	user_name	<i>(Optional)</i> Unique username.
Email	String	100-character alphanumeric including dashes.	email	<i>(Optional)</i> Valid email address .
Street - address 1	String		street	<i>(Optional)</i> User's primary street number and name.
City - address 1	String	50-character alphanumeric including dashes.	city	<i>(Optional)</i> User's primary city name.
Province - address 1	String	2-character postal abbreviation eg. "MA"	province, state	<i>(Optional)</i> User's primary province or state.
Postal Code - address 1	String	20-character alphanumeric including dashes.	postal_code, zip	<i>(Optional)</i> User's primary postal code or zip.
Street - address 2	String	20-character alphanumeric including dashes.	street2	<i>(Optional)</i> User's alternate street number and name.
City - address 2	String	50-character alphanumeric including dashes.	city2	<i>(Optional)</i> User's alternate city name.
Province - address 2	String	2-character postal abbreviation eg. "MA"	province2, state2	<i>(Optional)</i> User's alternate province or state.
Postal_Code - address 2	String	20-character alphanumeric including dashes.	postal_code2, zip2	<i>(Optional)</i> User's alternate postal code or zip.

Attribute	Type	Limits	Possible Names	Description
Cell phone number	String	20-character alphanumeric including dashes.	phone_cell	(Optional) User's cell phone number
User Type ID	String	Id number of UserType	user_type_id	(Optional) A reference number to the type of user.
Employee Number	String	50-character alphanumeric including dashes.	employ_no	(Optional) User's employee number.
Employee phone number	String	50-character alphanumeric including dashes.	employee_phone	(Optional) User's employee phone number.
Student number	String	50-character alphanumeric including dashes.	student_no	(Optional) User's student number.
Student phone number	String	50-character alphanumeric including dashes.	student_phone	(Optional) User's student phone number.
Driver licence number	String	255-character alphanumeric including dashes.	driver_licence_num, driver_license_num	(Optional) User's driver's licence number or driver's license number.
Date of Birth	String	10-character date in format 'yyyy-mm-dd'	date_of_birth	(Optional) User's date of birth in format "YYYY-MM-DD.
Locker User Type ID	String	Id number of LockerUserType	locker_user_type_id	(Optional) A reference the the user's locker type id of the user.
Driver's licence province	String	2-character postal abbreviation eg. "MA"	driver_licence_prov, driver_license_state	(Optional) The province or state of the user's driver's licence.

# Successful Response

The response will be a json object.

Content-Type: application/json

```
{
  "status" : "success",
  "reference_id" : 44
}
```



# Pull API: Overdue Violations

The OPS-COM Controller provides a simple JSON based API to integrate with.

Clients use this API to export a list of the currently overdue violations, which are then marked as having been sent to collections.

Make sure you set the HTTP Content-Type header to be application/json.

## Information

When accessed, this API will send all overdue violations that have not been sent to collections yet in a JSON object. It will also mark them as having been sent to collections, so subsequent calls to the API will not get the same information more than once.

This is a POST request. Data is being posted to the server.

**POST /api/{client}/v1/violations/send\_overdue\_to\_collections**

**Sample Request - All Params**

/api/OC\_TOMA/v1/violations/send\_overdue\_to\_collections

## Making API Requests

### Raw Request:

```
POST /api/OC_TOMA/v1/violations/send_overdue_to_collections HTTP/1.1
Host: controller.operationscommander.io
Accept: application/json
Content-Type: application/json
Cache-Control: no-cache

{
  "apiToken": "YOUR-API-TOKEN",
}
```

# JavaScript Request:

```
var request = new XMLHttpRequest();

request.open( 'POST' , 'https://controller.operationscommander.io/api/OC_TOMA/v1/violations/send_overdue_to_collections ');

request.setRequestHeader( 'Content-Type' , 'application/json' );
request.setRequestHeader( 'Accept' , 'application/json' );

request.onreadystatechange = function () {
  if ( this .readyState === 4) {
    console.log( 'Status:' , this .status);
    console.log( 'Headers:' , this .getAllResponseHeaders());
    console.log( 'Body:' , this .responseText);
  }
};

var body = {
  "apiToken" : "YOUR-API-TOKEN"
}
request.send(JSON.stringify(body));
```

## Request Object Attributes

Attribute	Type	Limits	Possible Names	Description
apiToken	String	50-character alphanumeric including dashes.	apiToken	<b>(Required)</b> Your supplied API Token.

## Successful Response

The response will be a JSON object.

Content-Type: application/json







```
{
  "records" : [
    {
      "ViolationID" : 9,
      "SemPermitID" : 0,
      "Ticket" : "1-100013" ,
      "VehicleID" : 8,
      "Spoiled" : 0,
      "TicketType" : 2,
      "Issued" : "2016-05-03T20:00:00.000000Z" ,
      "convNotice" : null ,
      "Due" : "2016-05-23T20:00:00.000000Z" ,
      "ViolationTypeID" : -1,
      "Fine" : 50,
      "AdjustedFine" : 0,
      "Towing" : 0,
      "taxAmount" : "0.0000" ,
      "Writer" : 8,
      "LocationID" : 6,
      "Comment" : null ,
      "TicketAppeal" : null ,
      "appealType" : 0,
      "appealFormat" : null ,
      "AccessAdminID" : 7,
      "ActionedPer" : null ,
      "Created" : "2020-10-15T20:13:24.000000Z" ,
      "AppealUserID" : 0,
      "AppealProcessDate" : null ,
      "AppealAdminID" : 0,
      "AppealComment" : null ,
      "AppealAdminComment" : null ,
      "AutoNotice" : 2,
      "ProcessedByCollection" : null ,
      "SentToCollections" : null ,
      "PrivateComments" : null ,
      "DriveAway" : 0,
      "UUID" : "d9a1c8bb-4ed1-411e-91b1-0b63ba52e04d" ,
      "VioNotice" : null ,
      "latitude" : null ,
      "longitude" : null ,
      "Warning" : null ,
      "userid" : null ,
      "incidentID" : null ,
      "failTolIdentify" : null ,
      "pin" : null ,
      "duplicate" : null ,
      "AdjustmentReason" : null ,
      "user" : null ,
      "vehicle" : {
        "VehicleID" : 8,
        "Active" : 1,
        "lastUpdate" : "2020-10-15T20:13:03.000000Z" ,
        "Plate" : "AJNR123" ,
        "PlateTypeID" : 4,
        "ProvID" : 9,
        "MakeID" : 13,
        "TypeID" : 5,
        "ColourID" : 14,
        "Year" : 2006,
        "TotalVio" : 0,
        "TotalUnpaid" : 0,
        "TotalWarning" : 0,
        "created" : "2020-10-15T20:13:03.000000Z" ,
        "externallookupdate" : null ,
        "externallookupRequestID" : null ,
        "modified" : "2022-03-16T21:36:09.000000Z" ,
        "vehicleAlert" : null ,
        "vin" : null ,
        "drivers" : [
          {
            "UserID" : 73,
            "enabled" : null ,
            "salutation" : "Dr." ,
            "firstName" : "stephen_14Oct_1114" ,
```

