

PassportParking API

The Passport API provides you with a powerful, convenient and simple web services API for interacting with Passport. The API uses JSON for the response format. To get started you need to request an API key for access to all APIs.

Getting Started

To get started you should get a Passport API Key, Review the Basic Scenarios and review the API references.

Requesting an APIKEY

An APIKEY is required to access all Passport API's. It is used for all API calls to any Passport service. Having an API key lets Passport "tag" each API call from all integrations, and helps provide both an additional layer of security and helps Passport support our partners. An API key is REQUIRED for all integrations, and if you want to move to production (make calls to ppprk.com) you must request a production key. An API key is generated for each customer that accesses the system. The API gives us the information about who is requesting the data and which operator the data is being requested for.

You can use the following key for testing in our staging environment:
3822158dad271ec27274d4973005d5d5

Without an API key you will receive the following error on any request:

```
{"status":403,"reason":"You are not authorized to access this API resource based on API Key","apiversion":"2.0"}
```

Whitelist IP Addresses

Our APIs are hosted behind the following IPs. If necessary please make sure to whitelist the following IPs for our US datacenter.

54.174.189.69
54.152.217.142
54.174.73.45 - Staging

Disaster Recovery Environment
52.8.36.174
52.8.98.147
52.8.107.217

API URLs

The base URL for all operators is formatted as follows:

```
https://{server}/apps/{apiVersion}/server/opmgmt/api/partner_index.php
```

Test URL: https://passportmobile.net/apps/v7/server/opmgmt/api/partner_index.php

US Production URL: https://ppprk.com/apps/v7/server/opmgmt/api/partner_index.php

Canada Production URL: https://passportca.com/apps/v7/server/opmgmt/api/partner_index.php

Monitoring

Get Zone List API

The getzonelist API is used for identifying all of the zones that are available for monitoring. A zone is a grouping of either spaces or vehicles usually tied to a particular lot or street. In some instances a single zone is created for the entire parking operations. The zone list will identify to you the type of lot you are monitoring (space based, license pl). With this information you know you'll be receiving a list of spaces that are parked in for a particular lot or a list of license plates.

Passing an additional parameter "zonetype=monitorzone" will return monitoring zones if configured. These zones are used when grouping subzones.

An example of the API call is listed below along with the sample output.

https://passportmobile.net/apps/v7/server/opmgmt/api/partner_index.php/getzonelist?apikey=3822158dad271ec27274d4973005d5d5

```
{
  status: 200,
  data: [
    {
      id: "456623",
      zoneid: "456623",
      zonename: "6401 N Clark St",
      zonenum: "000001",
      enablecashcheckin: "0",
      forcelogofftime: "0",
      zonetype_id: "3",
      monitorspacebasedactiveonly: "0",
      zonetype_name: "license pl",
      operator_id: "64",
      children: []
    },
    {
      id: "457982",
```

```

zoneid: "457982",
zonename: "Test Lot 918 Rate",
zonenumber: "000002",
enablecashcheckin: "0",
forcelogofftime: "0",
zonetype_id: "3",
monitorspacebasedactiveonly: "0",
zonetype_name: "license pl",
operator_id: "64",
children: [ ]
}
]
}

```

Detailed field descriptions of Get Zone List API

Field	Type	Description
Zoneid	Integer	Unique id of zone. primary key will not change.
zonename	String	Friendly Zone Name defined by the operator
zonenumber	String	Friendly Zone number defined by the operator
Enablecashcheckin	Integer	Can cashbox checkin take place in park monitor? used when operators allow manual checkins
Forcelogofftime	Integer	Force logoff at set time. Hour of day that all parking resets.
monitorspacebasedactiveonly	Integer	Only return active spaces. When a zone has thousands of spaces, we have the option to only show the available spaces instead of all spaces.
Zonetype_id	Integer	Unique id to reference zone type
Zonetype_name	String	Description of zone type (space base, license pl)
Operator_id	Integer	Unique id of operator
zonetype	string	monitorzone = Returns monitoring zones for space based grouping of subzones

Get Zone Info API

Based on the result from the previous API call (getzonelist) you can call getzoneinfo to retrieve the active spaces or license plates for the requested parking lot. This lookup is done based on the zoneid retrieved from getzonelist API.

The API also gives you the ability to look at an exact space number or license plate number for exact match checking.

An example of looking up a zone (456195) via getzoneinfo

https://passportmobile.net/apps/v7/server/opmgmt/api/partner_index.php/getzoneinfo?apikey=3822158dad271ec27274d4973005d5d5&zoneid=456195

Inbound Parameters

Field	Type	Description
Apikey	String	Provided for access to api

Zoneid	Integer	Unique id to access zone (not required for some operators)
Spacenumber	String (not required)	Friendly number for space that is being check
Lpn	String (Not required)	Lpn being checked (state not required)
monitorzoneid	Integer	gets groups of spaces based on subzones.
startspace	Integer	startspace and end space can be used to limit the number of records returned when querying large zones
endsapce	Integer	

Based on the type of zone that was requested you will receive the space or license plate results

1. Sample Response for Space based zone

```
{
  status: 200,
  data: [
    {
      operator_id: "1",
      zoneid: "413",
      zonenumber: "123",
      zonename: "Downtown",
      zonetype: {
        id: "2",
        name: "space base"
      },
      enablecashcheckin: "0",
      forcelogofftime: "0",
      locations_spaces: [
        {
          locationid: "265",
          locationname: "Downtown",
          zoneid: "413",
          spaces: [
            {
              id: "132363",
              number: "128",
              name: "7-101A",
              zone_id: "366291",
              zonenumber: "2222223",
              entrytime: "2015-05-28 17:05:00",
              exittime: "2015-05-28 19:05:00",
              formattedentrytime: "2015-05-28 13:05:00",
              formattedexittime: "2015-05-28 15:05:00",
              activelpn: false,
              expiration: "118",
              expirationtimeinsecs: "118",
              isactive: "1",
              occupied: "-1"
            },
            {
              id: "132546",
              number: "311",
              name: "MA-706B",
              zone_id: "366291",
              zonenumber: "2222223",
              entrytime: "2015-05-28 17:20:00",
              exittime: "2015-05-28 19:20:00",
              formattedentrytime: "2015-05-28 13:20:00",
              formattedexittime: "2015-05-28 15:20:00",
            }
          ]
        }
      ]
    }
  ]
}
```

```

    active_lpn: false,
    expiration: "0",
    expiration_time_in_secs: "0",
    is_active: false,
    occupied: "-1"
  }
}
],
space_count: 50,
location_count: 1,
check_in_enabled: false,
location_lpn: [ ]
}
]
}

```

Result Object Field Descriptions

Field	Type	Description
Id	Integer	Unique id of the zone
Name	String	Friendly Zone Name
Number	String	Friendly Zone Number
Zonetype_name	String	Zone type
Zonetype_id	Integer	Unique id to reference the type of zone
Operator_id	Integer	Unique id to reference operator
Spaces	Object	See Space Object

Space Object (Array of Spaces in the selected zone)

Field	Type	Description
Id	Integer	Unique id of parking space
Number	String	Friendly parking space number (designated on meter or space number)
name	String	Internal Reference for Space (Internal Meter Number)
Zone_id	Integer	Unique id of the zone
Zonenum	String	Friendly Number of zone
entrytime	DateTime	UTC time of most recent entry
exittime	DateTime	UTC of most recent exit
formattedentrytime	DateTime	Local time of most recent entry
formattedexittime	DateTime	Local time of most recent exit
Active_lpn	String	This field has user entered data from the parking monitor user, usually a LPN number but can be used for other things. returns false when null
Expiration	String	Remaining seconds as a string
Expiration_time_in_secs	Integer	Remaining seconds as an integer
is_active	String	1 or false (If parker is currently paid)
occupied	String	Used for sensor based systems

2. If parameter [spacenumber] is passed in the result will be an occupancy true result or false. In the event the space does not exist you will receive a 405 for failure.

```
{
  status: 200,
  data:
  {
    occupied: true
  }
}
```

3. Sample Response for LPN Based Lot

```
{
  "status": 200,
  "data": [
    {
      "operator_id": "9",
      "zoneid": "66",
      "zonenumber": "9995",
      "zonename": "LPN Test Zone",
      "zonetype":
      {
        "id": "3",
        "name": "license pl"
      },
      "enablecashcheckin": "1",
      "forcelogofftime": "0",
      "locations_spaces": [ ],
      "space_count": 0,
      "location_count": 0,
      "checkinenabled": false,
      "location_lpn": [
        {
          "parkerentryid": "9",
          "entrytime": "2013-04-02 10:56:49",
          "exittime": "2013-04-02 10:56:49",
          "vehicleid": "1630",
          "licenseplatenumber": "NSHAN",
          "expirationtimeinsecs": "3597",
          "lpn": "IL TEST",
          "stateabbreviation": "IL",
          "zonenumber": "9995",
        }
      ]
    }
  ]
}
```

Result Object Field Descriptions

Field	Type	Description
Id	Integer	Unique id of the zone
Name	String	Friendly Zone Name
Number	String	Friendly Zone Number
Zonetyppename	String	Zone type
Zonetype_id	Integer	Unique id to reference the type of zone
Operatorid	Integer	Unique id to reference operator

Location_LPN	Object	See Location_LPN Option
--------------	--------	-------------------------

Location_LPN Object (Array of Vehicles in the selected zone)

Field	Type	Description
Entrytime	Date time	Entry time of vehicle (returned in localtime)
Vehicleid	Integer	Unique id for vehicle that is parked
Zoneid	Integer	Unique id for zone
Zonenumber	String	Friendly number for zone
Zonename	String	Friendly name of zone
Licenseplatenumber	String	License Plate of active vehicle
Expirationtimeinsecs	String	Remaining seconds as a string

Lookup LPN

The getlpnstatus API give you the ability to query a particular license plate number across any zone for that particular operator. This can be used for a final check before issuing a citation or as a custom lookup for license plates.

An example below of looking up license plate "TEST"

https://passportmobile.net/apps/v7/server/opmgmt/api/partner_index.php/getlpnstatus?apikey=3822158dad271ec27274d4973005d5d5&lpn=TEST

Inbound Parameters

Field	Type	Description
Apikey	String	Provided for access to api
Lpn	String	Lpn being checked
includetransactionexitafter	DateTime	Optional, Used to include expired transactions in GMT. Usually used 2016-06-22 00:00:00

Sample Response

```
{
  "status":200,
  "data":[
    {
      "parkerentryid":"12345",
      "entrytime":"2013-09-23 13:11:37",
      "exittime":"2013-09-23 16:11:37",
      "vehicleid":"10273",
      "licenseplatenumber":"BRAD",
      "expirationtimeinsecs":"10239",
      "lpn":"AK BRAD",
      "stateabbreviation":"AK",
      "ratenamename":"All Day Rate",
      "zonenumber":"101"
    }
  ]
}
```

```
}
}
```

Result Object Field Descriptions

Field	Type	Description
Entrytime	Date time	Entry time of vehicle (returned in localtime)
exittime	Date time	exit time of vehicle (returned in localtime)
Vehicleid	Integer	Unique id for vehicle that is parked
Zoneid	Integer	Unique id for zone
Zonenumber	String	Friendly number for zone
Zonename	String	Friendly name of zone
Licenseplatenumber	String	License Plate of active vehicle
lpn	string	state + license plate
stateabbreviation	String	Plate state
Expirationtimeinsecs	String	Remaining seconds as a string
Authorizationamountincent	Integer	amount authorized on parkers card

Rate Tester API

The ratetester API gives you the ability to test rates in an automated fashion by providing a start date / time and a duration. The API will also provide the shortcuts that would be present to a user at that point in time.

An example below of looking up license plate "TEST"

https://passportmobile.net/apps/v7/server/opmgmt/api/partner_index.php/testrate?apikey=3822158dad271ec27274d4973005d5d5&zonenumber=139&entrydatetime=2015-01-05%2011:42&duration=60

Inbound Parameters

Field	Type	Description
Apikey	String	Provided for access to api
zonenumber	String	Lpn being checked
entrydatetime	date	date time of entry
duration	int	duration in minutes

Sample Response

```
[
{
  lockouts: [ ],
  shortcuts: [
    {
```



```

        parkingfeeincents: 600,
        conveniencefeeincents: "0",
        taxdetail: false,
        shortcut: "Day Rate (until 6:00 PM on Monday)"
    },
    {
        parkingfeeincents: 1100,
        conveniencefeeincents: "0",
        taxdetail: false,
        shortcut: "Night Rate (until 7:00 AM on Tuesday)"
    }
],
durations: [
    {
        durationinminutes: "60",
        invalidreason: "The parker doesn't have paymentcard information."
    }
]
}
]

```

Result Object Field Descriptions

Field	Type	Description
lockouts	array	Restrictions that would prevent parking at this time
shortcuts	array	shortcut for parker based on breakpoints configured in the system
durations	object	response for input duration

Completed Transactions API

Completed Transaction API

The completed transaction API gives you the ability to pull all completed transaction by date / time

[http://ppprk.com/apps/v7/server/opmngmt/api/partner_index.php/getsettledtransaction?apikey=\[KEYHERE\]&startdate=05/01/2014&enddate=05/01/2014&reportdate_filter=capture_date](http://ppprk.com/apps/v7/server/opmngmt/api/partner_index.php/getsettledtransaction?apikey=[KEYHERE]&startdate=05/01/2014&enddate=05/01/2014&reportdate_filter=capture_date)

Inbound Parameters

Field	Type	Description
Apikey	String	Provided for access to api
startdate	date	05/01/2014 00:00:00

enddate	date	05/01/2014 23:59:59
reportdate_filter	string	capture_date

Response

```
{
  parkers:
  [
    {
      recordid: "384429",
      operator_id: "1",
      operator_name: "Operator",
      operator_contact: "7042711189",
      parkerentryid: "439517",
      parkerid: "101580",
      entrytime: "05/01/2014 09:19 PM",
      exittime: "05/01/2014 10:00 PM",
      entrytime_utc: "2014-05-02 02:19:00",
      exittime_utc: "2014-05-02 03:00:00",
      entrytimestamp: "1398979140",
      exittimestamp: "1398981600",
      zone_id: "455891",
      parkingdurationinmins: "41",
      phonenumber: "4803634976",
      zonenumber: "123",
      zonename: "109 N Halsted St",
      locationname: null,
      entrytype: "app",
      mobiletransactionfeeincents: "10",
      totalconveniencefeeincents: "35",
      spacelpn: "ABC123",
      internalspaceid: null,
      licenseplatestate: "IL",
      validatedamountincents: "0",
      transaction_fee: "10",
      net_fee: "25",
      show_par: "1",
      fullparkingfeeincents: "273",
      totalparkingfeeincents: "308",
      total_net_revenue: "298",
      totaltaxincents: "0",
      billingtype: "Zone Cash",
      billingtypeid: "2",
      cardtail: "1017",
      cardtype: "AmericanExpress",
      transactionid: "796331",
      parentzonename: null,
      parentzonenumber: null,
      rateenforcementname: "$4.00",
      timezonename: "America/Chicago",
      tickets_count: "1",
    }
  ]
}
```

```
    zones: "455891",
    zones_count: "1",
    additional_info: null,
    captured: "1",
    vehicleid: null,
    vehiclename: null,
    rate_name: "428011"
  },
```

Push Service

The Passport Push Service integration allows you to receive a JSON http request of the information. On each parking transaction the following information will be posted to a URL that you specify.

Authentication methods supported:

1. X-API-Key
2. Https Userpassword
3. Generic API key

Sample Post JSON - Vehicle-based Parking Lot

```
{
  "method": "publishParkedVehicle",
  "parkerentryid": 392939,
  "entrytime": "2013-09-23 13:11:37",
  "exittime": "2013-09-23 16:11:37",
  "currenttime": "2013-09-23 16:11:37",
  "vehicleid": "10273",
  "licenseplatenumber": "BRAD",
  "expirationtimeinsecs": "10239",
  "lpn": "AK BRAD",
  "stateabbreviation": "AK",
  "authorizationamountincents": "835",
  "ratename": "39 Lot Rate",
  "zonenum": "101",
  "zonename": "Library Lane"
}
```

Sample Post JSON - Space-based Parking Lot

```
{
  "method": "publishParkedSpace",
  "parkerentryid": 392949,
  "entrytime": "2013-09-23 13:11:37",
  "exittime": "2013-09-23 16:11:37",
  "currenttime": "2013-09-23 16:11:37",
  "spacenum": 1,
  "spaceinternalid": 1,
  "spacemappingvalue": 1,
  "expirationtimeinsecs": "10239",
  "authorizationamountincents": "835",
}
```

```

"ratename":"39 Lot Rate"
"zonenumber":"101",
"zonename":"Library Lane"
}

```

Object Field Descriptions

Field	Type	Description
method	String	publishParkedVehicle, publishParkedSpace
entrytime	Date time	Entry time of vehicle (returned in localtime)
exittime	Date time	exit time of vehicle (returned in localtime)
vehicleid	Integer	Unique id for vehicle that is parked
zoneid	Integer	Unique id for zone
zonenumber	String	Friendly number for zone
zonename	String	Friendly name of zone
licenseplatenumber	String	License Plate of active vehicle
lpn	string	state + license plate
stateabbreviation	String	Plate state
Expirationtimeinsecs	String	Remaining seconds as a string - Check on this
Authorizationamountincent	Integer	amount authorized on parkers card
spacenum	String	Space number the parker has parked in
extensionentrytime*	Date time	Time of an extension, when transaction is an extension

Additional Integration Helpers

System Availability Check

Our system availability check gives you access to check if our systems are active. You can ping this every minute or at your desire to determine if our systems are available. At any point in time you can check the availability of the system by hitting the following link. A valid response would be a status code "200"

http://ppprk.com/apps/v7/server/opmgmt/api/partner_index.php

```

{
    status: 200,
    data:
    {
        Name: "Operator",
        Version: "2.0",
        DocumentationPath: "doc"
    }
}

```

Creating Test Data

Need to create some test transactions to see them in the monitoring API? Here is the link to the test application that correlates with the monitoring API and zones you have access to.

Use the zonelistAPI to find the zone numbers you can park in.

<http://easypparking.com/park>

Park in the zone related to the zone you are testing.

Use Credit Card: 4111-1111-1111-1111 Exp 12/20. Zip: 12345