API Documentation for Florida Turnpikes App

Table of Contents

Schema Routes 2

Last Update Timestamp 2

Mobile Schema 2

Routing Routes 4

Get Valid Endpoints 4

Get Route 4

# HTTP Response Codes

|  |  |  |
| --- | --- | --- |
| statusCode | status (Text) | Description |
| 200 | successful | Everything was A-ok |
| 400 | successful | Returned when no results found in database |
| 404 | fail | Returned when route not found or the request never made it to the backend |
| 500 | fail | ‘An internal Error Happens’. Problem in back somewhere. |
|  |  |  |
|  |  |  |
|  |  |  |

Schema Routes

## Last Update Date (Real)

Type: GET

Route: <url>/api/schema/lastUpdate

Example: <http://nodejsapi-maryland.rhcloud.com/api/schema/lastUpdate>

Parameters: none

Notes: This returns the date for the last time any data in the backend was updated.

Example Response:

{

results: [

{

lastUpdated: "06/19/2014"

}

],

resultCount: 1,

statusCode: "200",

message: null,

status: "successful",

responseTime: "0ms"

}

## Mobile Schema(mock)

Type: GET

Route: <url>/api/schema/mobileSchema

Example: <http://nodejsapi-maryland.rhcloud.com/api/schema/mobileSchema>

Parameters: none

Notes: This returns the schema and data for all data that will be used in the mobile applications. This returns the point data for facility/toll plaza locations, regions, cartypes (axles), payment methods, website, phone number, and an object for other info.

Example Response:

# TO BE ADDEDRouting Routes

## Get Valid Endpoints (Real)

Type: GET

Route: <url>/api/routing/validEndPoints/startFacilityID/{startfacilityid}/startExitNum/{startexitnum}

Example: [http://nodejsapi-maryland.rhcloud.com/api/routing/validEndPoints/startFacilityID/V/startExitNum/2B](http://nodejsapi-maryland.rhcloud.com/api/routing/validEndPoints/startPointID/6230)

Parameters: REQUIRED – startfacilityid: string, startexitnum: string

Notes: This returns all valid end points along with toll plazas given the start point. The attributes include exitName, exitFacilityType (‘Exit’ for valid end point, ‘Toll Plaza’ for toll plaza), exitNum, sunPassOnly(Boolean, can be used for filter given payment type), lat, lon.

Example Response:

{

"results":[

{

"exitName":"Clearwater (S.R. 60 West)",

"exitFacilityType":"Exit",

"exitFacilityID":"V",

"exitNum":"2A",

"sunPassOnly":false,

coordinates: {

lat: 27.967032,

lon: -82.546801

}

}

],

"resultCount":1,

"statusCode":"200",

"message":null,

"status":"successful",

"responseTime":"0ms"

}

## Get Route (Real)

Type: GET

Route and Toll for specific car type: <url>/api/routing/route/startFacilityID/{startfacilityid}/startExitNum/{startexitnum}/endFacilityID/{endfacilityid}/endExitNum/{endexitnum}/axleCount/{axlecount}

Example: http://nodejsapi-maryland.rhcloud.com/api/routing/route/startFacilityID/V/startExitNum/16/endFacilityID/V/endExitNum/9/axleCount/2

Parameters:

* startfacilityid (REQUIRED) – string, starting facility id for trip
* startexitnum (REQUIRED) – string, starting exit number for trip
* endfacilityid (REQUIRED) – string, ending facility id for trip
* endexitnum (REQUIRED) – string, ending exit number for trip
* axlecount (OPTIONAL) – integer, number of axles for the vehicle. When this is absent, the API returns the route information for all axles in an array

Notes: This returns route, as well as costs of the trip.

Example Response:

{

results: [

{

routeAmount: {

start: {

sp: 0,

cash: 0

},

end: {

sp: 0.26,

cash: 0.5

},

tollPlazas: [

{

name: "Sugarwood Toll Plaza",

sp: 0.77,

cash: 1

}

]

},

axle: 2,

sp: 1.03,

cash: 1.5,

miles: 8.26

}

],

resultCount: 1,

statusCode: "200",

message: null,

status: "successful",

responseTime: "1ms"

}

## Get TollInfo (Real)

Type: GET

Toll Info for specific car type: <url>/api/routing/route/facilityID/{facilityid}/type/{type}/name/{name}/axleCount/{axlecount}

Example:

Toll plaza

http://nodejsapi-maryland.rhcloud.com/api/schema/tollinfo/facilityID/M/type/tollplaza/name/Leesburg Toll Plaza/axleCount/2

exit:

http://nodejsapi-maryland.rhcloud.com/api/schema/tollinfo/facilityID/M/type/exit/name/C.R.%20470/axleCount/2

Parameters:

* facilityID (REQUIRED) – string, facility id
* type (REQUIRED) – string, facility type, can be either ‘exit’ or ‘tollplaza’
* name (REQUIRED) – string, name of exit or toll plaza
* axlecount (REQUIRED) – integer, number of axles for the vehicle.

Notes: This returns route, as well as costs of the trip.

Example Response:

Toll plaza

{

* **results**:

[

* + {
    - **name**: "Leesburg Toll Plaza",
    - **isBridge**: "N",
    - **description**: "blah, blah",
    - **cashToll**: 2.5,
    - **sunPassToll**: 2,
    - **coordinates**:

{

* + - * **lat**: 28.669225,
      * **lon**: -81.84133

},

* + - **milePost**: 241.202

}

],

* **resultCount**: 1,
* **statusCode**: "200",
* **message**: null,
* **status**: "successful",
* **responseTime**: "0ms"

}

exit:

{

* **results**:

[

* + {
    - **name**: "C.R. 470",
    - **exitNum**: "296",
    - **exitNumber**: "296",
    - **milePost**: 249.584,
    - **validStart**: true,
    - **NB\_ON**:

{

* + - * **sunpass**: 0.75,
      * **cash**: 1

},

* + - **NB\_OFF**:

{

* + - * **sunpass**: 0,
      * **cash**: 0

},

* + - **SB\_ON**:

{

* + - * **sunpass**: 0,
      * **cash**: 0

},

* + - **SB\_OFF**:

{

* + - * **sunpass**: 0.75,
      * **cash**: 1

},

* + - **EB\_ON**:

{

* + - * **sunpass**: 999,
      * **cash**: 999

},

* + - **EB\_OFF**:

{

* + - * **sunpass**: 999,
      * **cash**: 999

},

* + - **WB\_ON**:

{

* + - * **sunpass**: 999,
      * **cash**: 999

},

* + - **WB\_OFF**:

{

* + - * **sunpass**: 999,
      * **cash**: 999

},

* + - **coordinates**:

{

* + - * **lat**: 28.755398,
      * **lon**: -81.938771

},

* + - **sunpassOnly**: false,
    - **ticketSystem**: false,
    - **tollByPlate**: false

}

],

* **resultCount**: 1,
* **statusCode**: "200",
* **message**: null,
* **status**: "successful",
* **responseTime**: "1ms"

}