



Lizzy API

Last Modified By: Glenn Hancock
Last Modified: 08/14/2017
Version: 2.0.0.11

Introduction:

This document covers the use of the Lizzy API, this API is a REST interface. The Lizzy API makes use of JSON objects for passing data between the client and the server. Before you can use the Lizzy API, you must be granted access by the customer you're wishing to do business with. To learn more about this option the Lizzy customer should watch video in Interface Setup Procedures 1.9) Controlling Who Has Access To Your Data.

Available REST Methods

This section will list each available method and its current version. A brief description will accompany each method and you should refer to the respective section below for more detailed information on the method.

- ◆ **orderdetails** – This method can be called against a Lizzy Invoice Number to retrieve order details such as customer information and part details.
- ◆ **orderstatus** – This method allows you to check the status of an order.
- ◆ **submitorder** – This method allows you to submit an order into the respective company.
- ◆ **inventoryfeed** – This method takes a date and returns all current inventory changes for parts modified since that time.
- ◆ **inventoryitem** – This method takes a part number and returns information about it specifically
- ◆ **externalorders** – This method allows you to see orders generated for a webstore that were put directly into Lizzy with no reference currently available in the webstore itself.
- ◆ **serializedunits** – This method returns all currently in stock units or a single unit if it receives a VIN number.

orderdetails

This method is a GET request and requires two parameters **APIKey** and **InvoiceNumber**. The **APIKey** is the key provided to you when your access request was accepted. **InvoiceNumber** is the invoice number from Lizzy you wish to get information on. Only Phone in orders where the order details wouldn't already exist on the webstore a callable through this method.

The response will return a HTTP status code and the body will contain response text. If an error has occurred, the response text will be an error message, otherwise it will contain a JSON Array with the requested data. See appendix A for HTTP status codes.

URL: <https://api.nizex.com/orderdetails>

JSON Sample Response:

```
{["BusinessName":"Some Company", "FirstName":"John", "LastName":"Smith", "Address":"123 45th St.", "Address2":""," "EmailAddress":"jsmith@somecompany.com", "Phone":""," "Zipcode":"30332", "City":"Atlanta", "State":"Georgia", "Country":"United States", "ContactID":"123", "InvoiceDate":"2013-01-22 12:09:22", "Total":"22.01"]}
```

JSON Array Format:

```
Array
(
    [0] => stdClass Object
    (
        [ContactID]           => 3 (Lizzy's Internal ContactID)
        [BusinessName]       => Some Company
        [FirstName]          => John
        [LastName]           => Smith
        [EmailAddress]       => jsmith@somecompany.com
        [Phone]              => 7705154546
        [Address]            => 123 45th St.
        [Address2]           =>
        [City]               => Atlanta
        [State]              => Georgia
        [Zipcode]            => 30332
        [Country]            => United States
        [InvoiceDate]        => 2013-01-22 12:09:22
        [ShippingAmount]     => 10.95
        [ShipVia]            => UPS
        [Tax]                => 0
        [Total]              => 22.01
        [Parts]
        [0] => stdClass Object
        (
            [VendorName]     => Honda
            [PartNumber]     => 12345
            [Description]    => Generator
            [Qty]            => 1
            [ExtPrice]       => 12.95
        )
    )
    ...
)
```

serializedunits

This method is a GET request and has two parameters: **APIKey**, **WebstoreId**, **DateFrom** and **LocationID**. The **APIKey** is the key provided to you when your access request was created and LocationID is used on the dealership side and will only be relevant if the dealership has more than one physical location. Default value is 1 and this field is not required. **WebstoreId** is sent through if you wish to have Lizzy limit the results based on a certain Webstore you've setup in Lizzy. **DateFrom** if passed in will tell it to only provide units that have been modified since the date provided. This enables you to retrieve a delta release of serialized items. It is sent through as "11/01/2017 12:00:00" to retrieve all units received or sold since that date and time.

Note: InStock field below is only useful if doing a delta pull to show units that you may have received before but are no longer in stock and need to be pulled from the site

The response will return a HTTP status code and the body will contain response text. If an error has occurred, the response text will be an error message, otherwise it will contain a JSON Array with the requested data. See appendix A for HTTP status codes.

URL: <https://api.nizex.com/serializedunits>

JSON Array Format:

```
Array
(
    [0] => stdClass Object
    (
        [VendorName] => Name of Vendor in Lizzy
        [VIN] => Unique VIN or Serial number of units
        [UnitID] => Lizzy Internal ID. Only used if units are linked together.
        [InStock] => Whether unit is instock or not 1= instock, 0 = sold
        [ParentUnitID] => Passed back if this unit is attached to something else. Ie: Boat and Trailer
        [ModelNumber] => Units Model Number
        [ModelName] => Secondary Model Number that also represents the unit
        [VehicleType] => Type of vehicle (Street, Dirt...)
        [UnitType] => N,U,D,R ( New, Used, Demo, Rental)
        [Year] => Manufactured Year
        [MSRP] => List Price for Unit
        [SalePrice] => Asking Price for the unit
        [Accessories] => Total for accessories added to unit
        [AdditionalFees] => Setup, Doc, Freight...
        [CurrentOdometer] => Current Odometer Reading
        [CurrentHours] => Current Hours on the Unit
        [ModelDescription] => Detailed description of unit model (generic description)
        [UnitDescription] => Detailed description of unit (specific to this unit)
        [UnitCondition] => Description of current condition of unit
        [Color] => Primary color of unit
        [FuelType] => Type of Fuel used in Engine(s)
        [EngineMake] => make of Engine(s)
        [EngineCCs] => size of Engine(s)
        [EngineHP] => Horsepower of Engine(s)
        [EngineCyls] => Number of cylinders for engines(s)
        [Images]
```

```
[0]=> stdClass Object
(
  [URL]          => Location where image can be retrieved
  [ImageType]   => (T,S,M,L) Thumbnail, Small, Medium, Large
)

[DriveTrain]    => Drive train in unit
[Axes]          => Number of axles on unit

)
...
)
```

inventoryfeed

This method is a GET request and requires three parameters **APIKey**, **LocationID** and **DateFrom**. The **APIKey** is the key provided to you when your access request was added. **DateFrom** is the starting date that you want to use for producing the list of inventory and will return all items sold since that time. So if you pass in 5 minutes ago it will send back all parts inventory that have been modified or sold in the last 5 minutes. **WebstoreID** is not required but if sent will limit the response to only vendors that have been linked to that specific website.

NOTE: To retrieve the webstore id go to settings → webstore → setup webstore interface and look at the number in parens. LocationID can be found in settings → general → multi locations.

The **DateFrom** field supports both date and datetime values, the format is YYYY-MM-DD or YYYY-MM-DD HH:MI:SS. The LocationID is an integer field specified for the location (warehouse) you're interested in. Omit this variable if you want all inventory returned. You may also send a comma separated list like 1,3 and it will return inventory for locations 1 and 3. For information on what number to send for which location, speak with the Lizzy customer and have them provide the numbers from Settings → General → Multi-Locations.

The response will return a HTTP status code and the body will contain response text. If an error has occurred, the response text will be an error message, otherwise it will contain a JSON Array with the requested data. See appendix A for HTTP status codes.

URL: <https://api.nizex.com/inventoryfeed>

JSON Sample Response:

```
{["VendorID": "33", "VendorName": "Honda", "PartNumber": "12345", "Qty": "3",  
"List": "12.94", "Description": "belt", "SupersessionNumber": "", "OrigManuf": "cc  
bearing", "OrigManufPartNo": "434", "NLA": "NLA", ]}
```

JSON Array Format:

```
Array  
(  
  [0] => stdClass Object  
  (  
    [VendorID] => Internal ID for this Vendor in Lizzy  
    [VendorName] => Name of Vendor in Lizzy  
    [ItemNumber] => Actual Part number  
    [Description] => Full description of item  
    [Qty] => Total stock for item in Lizzy (Not location specific)  
    [List] => List Price for part  
    [OrigManuf] => Original Manufacturer of part in case ordering from supplier  
    [OrigManufPartNo] => Original Manufacturer part number  
    [NLA] => No Longer Available  
    [Supersession] => Number that supersedes this number  
    [MAP] => Minimum Advertised Price  
    [BackorderETA] => If backordered when is it expected to arrive  
    [BackorderNote] => Any available note for the item on backorder  
    [OnOrderQty] => How many are currently on open orders  
    [PromisedQty] => How many are currently promised to customers from orders
```

[ToBePulledQty] => How many items instock are spoken for already
)
...
)

inventoryitem

This method is a GET request and requires parameters **APIKey** and **ItemNumber** with two limiting parameters for **VendorID** and **VendorName**. The **APIKey** is the key provided to you when your access request was accepted. **ItemNumber** is required as is the **APIKey** but if you fail to give it a **VendorID** or a **VendorName** the response may include multiple entries where you will need to choose the one you wanted.

The response will return a HTTP status code and the body will contain response text. If an error has occurred, the response text will be an error message, otherwise it will contain a JSON Array with the requested data. See appendix A for HTTP status codes.

URL: <https://api.nizex.com/inventoryitem>

JSON Sample Response:

```
[{"VendorID": "33", "VendorName": "Honda", "PartNumber": "12345", "Qty": "3",  
"List": "12.94", "Description": "belt", "SupersessionNumber": "", "OrigManuf": "cc  
bearing", "OrigManufPartNo": "434", "Status": "NLA"}]
```

JSON Array Format:

```
Array  
(  
  [0] => stdClass Object  
  (  
    [VendorID] => Internal ID for this Vendor in Lizzy  
    [VendorName] => Name of Vendor in Lizzy  
    [ItemNumber] => Actual Part number  
    [Description] => Full description of item  
    [Qty] => Total stock for item in Lizzy (Not location specific)  
    [List] => List Price for part  
    [OrigManuf] => Original Manufacturer of part in case ordering from supplier  
    [OrigManufPartNo] => Original Manufacturer part number  
    [NLA] => No Longer Available  
    [Supersession] => Number that supersedes this number  
    [MAP] => Minimum Advertised Price  
    [BackorderETA] => If backordered when is it expected to arrive  
    [BackorderNote] => Any available note for the item on backorder  
    [OnOrderQty] => How many are currently on open orders  
    [PromisedQty] => How many are currently promised to customers from orders  
    [ToBePulledQty] => How many items instock are spoken for already  
  )  
  ...  
)
```

orderstatus

This method is a GET request and requires two parameters **APIKey** and **Payload**. The **APIKey** is the key provided to you when your access request was accepted. Payload will contain a JSON object named **OrderNumbers**, this object is a simple array containing the list of purchase order numbers that you want to retrieve. You can also pass in a list of **InvoiceNumbers** instead of **OrderNumbers** if you wish to check status on orders that were not processed through the website but entered directly into Lizzy.

The response will return a HTTP status code and the body will contain response text. If an error has occurred, the response text will be an error message, otherwise it will contain a JSON Array with the requested data. See appendix A for HTTP status codes.

URL: <https://api.nizex.com/orderstatus>

OrderNumbers or InvoiceNumbers JSON Object:

`{"OrderNumbers":["1","2","3"]}` (reference number passed from webstore) or
`{"InvoiceNumbers":["1001","2004","3001"]}` (Internal Lizzy Invoice Number)

JSON Sample Response:

`[{"OrderNumber":"1","Status":"Approved","TrackingNumber":"9400111969000940000011"},]`
or
`[{"InvoiceNumber":"1","Status":"Approved","TrackingNumber":"9400111969000940000011"},]`

JSON Array Format:

```
Array
(
    [0] => stdClass Object
    (
        [OrderNumber] => 1
        [InvoiceNumber] => 1001
        [Status] => Approved
        [BoxNumber] => 1
        [TrackingNumber] => 9400111969000940000011
        [ShipMethod] => UPS
        [DateShipped] => 2016-01-30
        [Parts]
        [0] => stdClass Object
        (
            [PartNumber] => 12345
            [Qty] => 2
            [Cost] => 2.45
            [Discount] => 0
            [ExtCost] => 4.90
        )
    )
)
```


submitorder

This method is a POST request and requires three parameters **APIKey**, **WebstoreID** and **Payload**. The **APIKey** is the key provided to you when your access request was accepted. Payload will contain a JSON object named **Orders**, this object is an array containing the purchase order objects that you want to submit to Lizzy. WebstoreID is a field supplied by the business you're sending orders into and can be found in Settings → Webstore → Setup Webstore Interfaces. It is supplied as an Integer and is not required. It is only necessary if you want the incoming order to be tied to a particular interface in Lizzy.

The response will return a HTTP status code and the body will contain response text. If an error has occurred, the response text will be an error message, if successful there is no response text. See appendix A for HTTP status codes.

URL: <https://api.nizex.com/submitorder>

Orders JSON Object:

```
{ "Orders": [{"OrderNumber": "825713", "Date": "2013-02-12", "Time": "15:28", "InvoiceNumber": "", "Tax": 1.99, "ShippingCost": 5.99, "Total": 28.83, "ShippingAddress": {"FirstName": "John", "LastName": "Smith", "EmailAddress": "jsmith@somecompany.com", "CompanyName": "Some Company", "Phone": "6783594184", "Address": "1735 Pennsylvania Ave.", "Address2": "", "City": "McDonough", "State": "GA", "ZipCode": "30353", "Country": "US", "Method": "Priority Mail"}, "BillingAddress": {"FirstName": "John", "LastName": "Smith", "EmailAddress": "jsmith@somecompany.com", "CompanyName": "Some Company", "Phone": "6783594184", "Address": "1735 Pennsylvania Ave.", "Address2": "", "City": "McDonough", "State": "GA", "ZipCode": "30353", "Country": "US"}, "OrderItems": [{"ProductID": "01-0013", "ProductCode": "PUN", "ProductDesc": "Oil Filter, Kawasaki", "Price": "6.95", "Quantity": "2"}, {"ProductID": "01-0021", "ProductCode": "PUN", "ProductDesc": "Oil Filter, Honda", "Price": "6.95", "Quantity": "1"}]}]}
```

Orders Array Format:

```
stdClass Object
(
  [Orders] => Array
  (
    [0] => stdClass Object
    (
      [OrderNumber]      => 825713
      [Date]             => 2013-02-12
      [Time]             => 15:28
      [InvoiceNumber]    =>
      [Tax]              => 1.99
      [SalesmanID]       => SalesPersonid from Lizzy that gets credit for sale if available.
      [ShippingCost]     => 5.99
      [CheckName]        => 0 false, 1=true Verify finding record in Lizzy to include first and last name.
                        (only applies to Amazon import interface)
```

[Total] => 28.83

[Customer]=> stdClass Object

```
(
  [FirstName]      => John
  [LastName]       => Smith
  [EmailAddress]   => jsmith@somecompany.com
  [CompanyName]   => Some Company
  [Phone]          => 6783594184
  [Address]        => 1735 Pennsylvania Ave.
  [Address2]       =>
  [City]           => McDonough
  [State]          => GA
  [ZipCode]        => 30353
  [Country]        => US
)
```

[ShippingAddress]=> stdClass Object

```
(
  [FirstName]      => John
  [LastName]       => Smith
  [CompanyName]   => Some Company
  [Phone]          => 6783594184
  [Address]        => 1735 Pennsylvania Ave.
  [Address2]       =>
  [City]           => McDonough
  [State]          => GA
  [ZipCode]        => 30353
  [Country]        => US
  [Method]         => Priority Mail
)
```

[BillingAddress]=> stdClass Object

```
(
  [FirstName]      => John
  [LastName]       => Smith
  [EmailAddress]   => jsmith@somecompany.com (used only if missing from customer)
  [CompanyName]   => Some Company
  [Phone]          => 6783594184
  [Address]        => 1735 Pennsylvania Ave.
  [Address2]       =>
  [City]           => McDonough
  [State]          => GA
  [ZipCode]        => 30353
  [Country]        => US
)
```

[OrderItems]=> Array

```
(
  [0] => stdClass Object
  (
    [ProductID]    => 01-0013
    [ProductCode]  => PUN
```


externalorders

This method is a GET request and requires parameters **APIKey** and **Date** with one limiting parameter for **WebstoreId** that you wish to check against. The **APIKey** is the key provided to you when your access request was accepted. It will respond with all **InvoiceNumbers** generated with that were created in Lizzy manually and tagged for a specific site. This method is used to collect data from Lizzy on orders a customer might have placed via phone instead of closing them out directly through the respective store.

Note: The WebstoreId can be obtained by going to Settings → Webstore → Setup Webstore Interfaces and looking in parens beside the name field.

The response will return a HTTP status code and the body will contain response text. If an error has occurred, the response text will be an error message, otherwise it will contain a JSON Array with the requested data. See appendix A for HTTP status codes.

URL: <https://api.nizex.com/externalorders>

JSON Sample Response:

```
 [{"InvoiceNumber": "33"} ]
```

JSON Array Format:

```
Array  
(  
  [0] => stdClass Object  
  (  
    [InvoiceNumber] => Internal Invoice Number in Lizzy  
  )  
  ...  
)
```


Appendix A:

HTTP Status Codes:

500	Internal Server Error, something went wrong on the server contact support
400	Bad Request, something is wrong with the submitted request check the response text for details
204	No Content, there was no data for the submitted request
201	Created, a record was created for your submitted request
200	OK, request was successful and data was found check the response text for the JSON Array