

# 8

## Dimensioner API

Resolution employs web-service communication to transfer data to the user's Software Infrastructure. Users may freely select the programming language to communicate with the Dimensioner. Please, ask for support by dropping an email to [sales@dimensioning.co.uk](mailto:sales@dimensioning.co.uk) if you require assistance.

### Web Service

Resolution Dimensioners support server and client modes in web service communication.

### Client Mode

When a Resolution dimensioner behaves as a client, it calls (consumes) the web-service of a remote web-server when a successful measurement is performed. This way, measurement results are automatically sent to the customer database in an asynchronous fashion. The address of the service of the remote web server is set in the "Integration Settings" page of GUI and WebUI. The data consists of following fields and values in JSON format:

Data Structure of Resolution Web Service in Client Mode		
Field Name	Data Type	Brief Description
StatusCode	int	0: success. Negative value: failure
StatusMessage	string	Empty when successful otherwise error explanation
Width	string	Width of the measured object
Length	string	Length of the measured object
Height	string	Height of the measured object

DimWt	string	Dimensional weight of the measured object. Calculated using the set dimensional weight coefficient/divisor
RealVolume	string	Real volume of the measured object
Weight	string	Weight value coming from the scale that is connected to the Resolution dimensioner
UnitID	string	Unit ID that is assigned to the device
BranchID	string	Branch (Location) ID that is assigned to the device
Barcode	string	Barcode of the object if barcode data is available otherwise empty
BarcodeType	string	Type of the barcode if available.
BarcodeSource	string	Source of the barcode data. Manual: from a handheld scanner. Auto: from a fixed mount barcode reader
Date	string	Measurement date
Time	string	Measurement time
MeasurementID	string	Unique ID of the measurement
ImageBase64	string	Image of the object encoded as Base64 string
SerialNumber	string	Serial number of the device
DimUnit	string	Unit of the dimensions provided within the result
WeightUnit	string	Unit of the weight provided within the result
IsRegular	int	1: when object is a regular shaped object
		0: when object has an irregular shape.

ObjectRGBCoordinates	string	Object's corner coordinates in RGB image  Format: $x_1, y_1   x_2, y_2   x_3, y_3   x_4, y_4$
Operator	string	User ID of the measurer
Reserved1	string	Reserved for future use
CRC	string	CRC16 checksum of the whole message excluding this value

### An example of successful measurement result message:

```
{
  "StatusCode":0,
  "StatusMessage":"","
  "Width":"40.1",
  "Length":"29.4",
  "Height":"21.0",
  "DimWt":"4.9",
  "RealVolume":"24018",
  "Weight":"0.84",
  "UnitID":"663",
  "BranchID":"1",
  "Barcode":"31959968681",
  "BarcodeType":"","
  "BarcodeSource":"Auto",
  "Date":"19.02.2019",
  "Time":"16:40:28",
  "MeasurementID":"5505",
  "ImageBase64":"","
  "SerialNumber":"160233",
  "DimUnit":"cm",
  "WeightUnit":"kg",
  "IsRegular":1,
  "ObjectRGBCoordinates":"43,44|22,47|48,42|5,5",
  "Operator":"admin",
  "Reserved1":"","
  "CRC":"123123"}

```

### An example successful image message:

```
{
  "StatusCode":0,
  "StatusMessage":"","
  "ID":"254",
  "Barcode":"31959968681",
  "ImageBase64":"<long_string_representation_of_image>"}

```

## Response from Remote-Server in Client Mode

If the “Wait response from remote server” option is enabled, Resolution dimensioner waits for a response from remote web server after sending the measurement data. Results are saved only if a success indicating response from remote server is received. If this option is not enabled, Volumizer does not expect a response from the remote web-server and the results are saved in local storage area regardless of the returned response message.

After a measurement is completed, “Please Wait” dialog box will appear and measurement results are sent to the remote web server. Resolution will wait for a JSON response from the remote server for 10 seconds. A dialog box will appear if a failure indicating response is returned from the server or if timeout occurs. The returned error message is shown on the screen. If the returned message indicates success, “Please Wait” dialog will disappear and results are shown.

Response data from the remote server should be in the following type in JSON format.

Successful response example:

```
{"ResponseCode":0,"ResponseMessage":""}
```

Unsuccessful response example:

```
{"ResponseCode":-3,"ResponseMessage":" Barcode not found"}
```

## Server Mode

Resolution has an internal HTTP server listening port 27018 for incoming requests. Users should make HTTP requests with the GET method if no data is passed from the user to the Resolution device, otherwise, the POST method should be used. HTTP Version 1.1 is supported.

Connection Protocol: HTTP based web service

Port: 27018



Service Address: <device\_ip>:27018/beevisionservice/<command\_name>

Return Data: Returns corresponding data in **JSON** format. If an error at the HTTP request occurs, the HTTP error code is returned. If an error occurs during internal operation, **StatusCode** and **StatusText** fields in returned JSON data provides more information about the error.

If the operation is successful, the returned StatusCode is 0. If the operation fails, returned StatusCode has a negative value indicating the type of the error.

## Accepted Commands by Volumizer API



Following commands may behave differently or may not exist at the Volumizer version prior to 3.4.3. Please, update your Volumizer if you are using an older version.

User can make HTTP requests with addresses provided in the following table. For example, to perform a measurement with a Resolution device with local IP address 192.168.1.10: **(can be simply tested with a browser)**

<http://192.168.1.10:27018/beevisionservice/measure>

returns a similar JSON data if operation was **successful**: (data structure is same as in client mode)

```
{
  "StatusCode": 0,
  "StatusMessage": "OK",
  "Width": "5.90",
  "Length": "5.90",
  "Height": "17.40",
  "DimWt": "0.085550",
  "RealVolume": "551",
  "Weight": "0.000000",
  "UnitID": "0",
  "BranchID": "0000",
  "Barcode": "",
  "BarcodeType": "",
  "BarcodeSource": "",
  "Date": "19.02.2019",
  "Time": "16:40:28",
  "MeasurementID": "5506",
  "ImageBase64": "",
  "SerialNumber": "160122",
  "DimUnit": "cm",
  "WeightUnit": "kg",
  "IsRegular": 1,
  "ObjectRGBCoordinates": "12,44|22,45|6,33|20,6",
  "Operator": "admin",
}
```

```
"Reserved1": "",  
"CRC": "123123"}
```

And returns a similar JSON data if operation **failed**:

```
{"StatusCode":-2,  
"StatusMessage":"Measurement canceled.",  
"Width":"0.0",  
"Length":"0.0",  
"Height":"0.0",  
"DimWt":"0.0",  
"RealVolume":"","  
"Weight":"","  
"UnitID":"","  
"BranchID":"","  
"Barcode":"","  
"BarcodeType":"","  
"BarcodeSource": "",  
"Date": "",  
"Time": "",  
"MeasurementID": "",  
"ImageBase64": "",  
"SerialNumber": "160122",  
"DimUnit": "cm",  
"WeightUnit": "kg",  
"IsRegular": 1,  
"ObjectRGBCoordinates": ""}
```

```
"Operator": "",
"Reserved1": "",
"CRC": "123123"}
```

## Accepted Commands at Resolution Web Service

Available at [http://<device\\_ip>:27018/beevisionservice/<command>](http://<device_ip>:27018/beevisionservice/<command>)

Command Name	HTTP Method Type	Brief Description
<a href="#">/measure</a>	GET	In non-automatic trigger modes, the measure command triggers a new measurement and returns results. Image field in the results is left empty.  If automatic trigger is selected, the latest measurement result is returned.
<a href="#">/captureMeasure</a>	GET	In non-automatic trigger modes, captureMeasure command triggers a new measurement and returns results with the image data encoded as base64 string.  If automatic trigger is selected, the latest measurement result is returned.
<a href="#">/getLastResult</a>	GET	Gets the latest result. Does not trigger a new measurement.  Image field is empty.
<a href="#">/getLastImageAndResult</a>	Get	Gets the latest result and image. Does not trigger a new measurement.
<a href="#">/startCamera</a>	GET	Starts 3D camera and returns OK message
<a href="#">/stopCamera</a>	GET	Stops 3D camera and returns OK message

<code>/calibrate</code>	GET	<p>Performs calibration and returns calibration success status.</p> <p>Does not work if SW is sealed, a corresponding error message is returned.</p>
<code>/getSerial</code>	GET	<p>Returns the serial number of the Resolution dimensioner. Has no effect on measurement parameters or data.</p>
<code>/getSWVersion</code>	GET	<p>Returns the version number of the Volumizer SW.</p> <p>Has no effect on measurement parameters or data.</p>
<code>/getCalibrationSettingsFile</code>	GET	<p>Returns calibration settings.</p> <p>Has no effect on measurement parameters or data.</p>
<code>/setCalibrationSettingsFile</code>	POST	<p>Overwrites calibration settings. It is advised to get settings first with <code>getCalibrationSettingsFile</code> command, modify settings and resend them with <code>setCalibrationSettings</code> command</p> <p>Does not work if SW is sealed, a corresponding error message is returned.</p>
<code>/getUserSettingsFile</code>	GET	<p>Returns user settings.</p> <p>Has no effect on measurement parameters or data.</p>
<code>/setUserSettingsFile</code>	POST	<p>Overwrites user settings. It is advised to get settings first with <code>getUserSettingsFile</code> command, modify settings and resend them with <code>setUserSettingsFile</code> command</p> <p>Has no effect on measurement parameters or data.</p>
<code>/getMeasurementSettingsFile</code>	GET	<p>Returns the measurement settings file. Legally relevant parameters are stored in this file.</p> <p>Has no effect on measurement parameters or data.</p>
<code>/setMeasurementSettingsFile</code>	POST	<p>Overwrites measurement settings file. It is advised to get settings first with <code>getMeasurementSettingsFile</code> command,</p>

		<p>modify settings and resend them with setMeasurementSettingsFile command</p> <p>Does not work if SW is sealed, a corresponding error message is returned.</p>
/resetDevice	GET	<p>Reboots the device.</p> <p>Has no effect on measurement parameters or data.</p>
/checkForUpdates	GET	<p>Checks for Volumizer SW update and returns an UpdateExists Boolean indicating the check result.</p>
/updateVolumizer	GET	<p>Updates Volumizer. CheckForUpdates must be run in prior to this command</p>
/getRecordsFile	GET	<p>Returns the results of saved previous measurements.</p> <p>Has no effect on measurement parameters or data.</p>
/getEthernetSettings	GET	<p>Returns Ethernet settings.</p> <p>Has no effect on measurement parameters or data.</p>
/setEthernetSettings	POST	<p>Overwrites Ethernet settings. Required fields in JSON data are:</p> <p>interfaceName: Name of the ethernet interface if more than one available. Default name is "Ethernet". Run "getEthernetSettings" first to get interface name.</p> <p>dynamicIP: dynamic or static</p> <p>ip1: first 3 digits of IP</p> <p>ip2: second 3 digits of IP</p> <p>ip3: third 3 digits of IP</p> <p>ip4: fourth 3 digits of IP</p> <p>gateway1: first 3 digits of gateway</p>

		<p>gateway2: second 3 digits of gateway</p> <p>gateway3: third 3 digits of gateway</p> <p>gateway4: fourth 3 digits of gateway</p> <p>subnet1: first 3 digits of subnet</p> <p>subnet2: second 3 digits of subnet</p> <p>subnet3: third 3 digits of subnet</p> <p>subnet4: fourth 3 digits of subnet</p> <p>dns1: first 3 digits of dns</p> <p>dns2: third 3 digits of dns</p> <p>dns3: third 3 digits of dns</p> <p>dns4: fourth 3 digits of dns</p> <p>Example JSON data:</p> <pre>{   "interfaceName": "Ethernet",   "dynamicIP": "static",   "ip1": "10",   "ip2": "2",   "ip3": "1",   "ip4": "110",   "subnet1": "255",   "subnet2": "0",   "subnet3": "0",   "subnet4": "0",   "gateway1": "0",   "gateway2": "0",   "gateway3": "0",   "gateway4": "0",   "dns1": "192",   "dns2": "168",   "dns3": "1",   "dns4": "1" }</pre> <p>If you change IP address with this command, do not forget to update the address of the Resolution dimensioner in your program</p> <p>Has no effect on measurement parameters or data.</p>
/getWifiSettings	GET	<p>Returns Wi-Fi settings</p> <p>Has no effect on measurement parameters or data.</p>
/setWifiSettings	POST	<p>Overwrites Wi-Fi settings. Same JSON fields as in setEthernetSettings command are required with this POST method</p>

		<p>If you change IP address with this command, do not forget to update the address of the Resolution dimensioner in your program</p> <p>Has no effect on measurement parameters or data.</p>
<a href="#">/scanWifiNetworks</a>	GET	<p>Starts a Wi-Fi scan and returns available Wi-Fi networks. This operation takes longer than 4 seconds</p> <p>Has no effect on measurement parameters or data.</p>
<a href="#">/connectToWifi</a>	POST	<p>Connects to the given network name. JSON data must include “network” and “password” key and corresponding values</p> <p>Has no effect on measurement parameters or data.</p>
<a href="#">/downloadRecordsFile</a>	POST	<p>Downloads the previously saved results as an XML file.</p> <p>Requires From, To and Download All fields as in the following example.</p> <pre>{"DownloadAll":1,"From":"","To":""}</pre> <p>If DownloadAll is 1, then whole file is downloaded, otherwise results between given dates are downloaded.</p> <p>From and To parameters date and time format should be like following</p> <pre>yyyy-MM-ddThh:mm           2019-12-25T19:00</pre> <p>Has no effect on measurement parameters or data.</p>
<a href="#">/downloadBatchRecordsZip</a>	GET	<p>Downloads a zip file containing group measurement results</p> <p>Has no effect on measurement parameters or data.</p>

<p><a href="#">/setExternalBarcodeResult</a></p>	<p>POST</p>	<p>Send barcode result data to the device. Required fields in JSON data are:</p> <p>BarcodeList that consists of following nested fields:</p> <ul style="list-style-type: none"> <li>Barcode</li> <li>BarcodeType</li> <li>ImageBase64</li> </ul> <p>An example JSON data:</p> <pre>{   "BarcodeList": [{Barcode:"1231231",BarcodeType:3 }], "ImageBase64":"" }</pre> <p>Has no effect on measurement parameters or data.</p> <p>This function is not supported in static Resolution dimensioners.</p>
<p><a href="#">/setExternalBarcodeImage</a></p>	<p>POST</p>	<p>Send an image of the package. Required fields in JSON data are:</p> <ul style="list-style-type: none"> <li>ImageBase64</li> </ul> <p>Has no effect on measurement parameters or data.</p> <p>This function is not supported in static Resolution dimensioners.</p>
<p><a href="#">/getDateAndTime</a></p>	<p>GET</p>	<p>Returns the date and time of the device in the following scheme:</p> <pre>{"StatusCode":0,"StatusMessage":"Operation successful","Date":"22/05/2019","Time":"00:27:07"}</pre> <p>Has no effect on measurement parameters or data.</p>
<p><a href="#">/setDateAndTime</a></p>	<p>POST</p>	<p>Overwrites the device date and time. It is recommended to get date and time data first with getDateAndTime, modify the data and set it with setDateAndTime</p>

		Has no effect on measurement parameters or data.
<a href="#">/downloadImagesZip</a>	GET	Downloads a zip file containing the images that device saved so far. Same JSON fields as in downloadRecordsFile command are required with this POST method  Has no effect on measurement parameters or data.
<a href="#">/deleteMeasurementImages</a>	GET	Deletes the images saved in the device.  Has no effect on measurement parameters or data.
<a href="#">/deleteMeasurementRecords</a>	GET	Deletes the measurement results saved in the device.  Does not work if SW is sealed, a corresponding error message is returned.
<a href="#">/getDeviceInfo</a>	GET	Returns the device related information as provided below.  <pre>{ "StatusCode":0,"StatusMessage":"Operation successful","DeviceModel":"Resolution 3DT270","SWChecksum":"12d6c1d5322723d59fe81731627ba4fe","CertificateNumber":"N/A","MinWidth":"10","MinLength":"10","MinHeight":"10","MaxWidth":"80","MaxLength":"80","MaxHeight":"80","WidthAccuracy":"1","LengthAccuracy":"1","HeightAccuracy":"1","CertificationStatus":"0" }</pre> Has no effect on measurement parameters or data.
<a href="#">/printZPLLabel</a>	POST	Send a ZPL formatted message to print a label using connected printer. Sample provided below:  <pre>{ "ZPLMessage": "^XA ^FX Top section with company logo, name and address. ^CF0,60 ^FO50,50^GB100,100,100^FS ^FO75,75^FR^GB100,100,100^FS ^FO88,88^GB50,50,50^FS ^FO220,50^FD3DTIM Elektronik^FS ^CF0,40 ^FO220,100^FDODTU^FS ^FO220,135^FDAnkara 06800^FS ^FO220,170^FDTurkey^FS ^FO50,250^GB700,1,3^FS ^XZ"</pre>

/startConveyor	POST	<p>Start the conveyor if available. If runTime_ms value is provided, conveyor automatically stops after provided duration</p> <p>Optional field:</p> <pre>{   "runTime ms": "1000" }</pre>
/startReverseConveyor	POST	<p>Start the conveyor in reverse direction if available. If runTime_ms value is provided, conveyor automatically stops after provided duration</p> <p>Optional field:</p> <pre>{   "runTime ms": "1000" }</pre>
/stopConveyor	GET	Stops the conveyor if available.