Image by Linoleum Magazine on Unsplash

Wix and IoT: How to show live sensor data on your Wix website using Gravio

公開日: 2020 年 7 月 10 日　(LinkedIn)

Author: Christoph Burgdorfer - Gravio Global Tech/Sales,Hong Kong

I was recently working on a project where we would collect data from a sensor network and display that data on a Wix website. In this particular case, the sensor was counting the number of people in a queue and it would show that very queue length on the company's website in real-time.

I thought it would be a good idea to write up a small tutorial on how that simple solution can be achieved. To build this, you don't need more than 20 minutes.

For the purpose of this tutorial here, we use a sensor to count people in a meeting room and share this information on a standard Wix website. The tutorial has three steps:

First, you need to know about Wix's capabilities ("Corvid") to ingest information via an API into a so-called "collection". A Wix collection is a bit like a database table.

Secondly, you need to know how to use Gravio to HTTP Post the sensor information to the Wix backend.

Thirdly, we will look at how the information in the collection can be displayed on the Wix front-end.

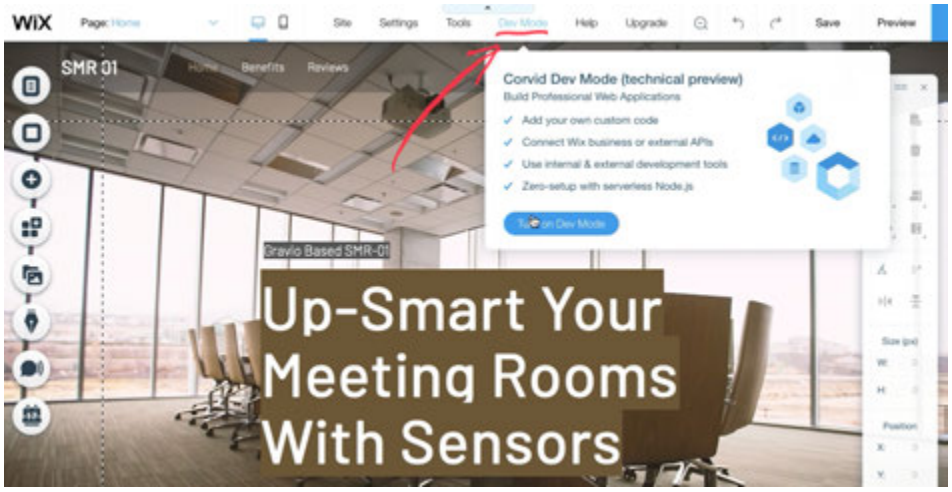
What you need to get started:

- A Wix account for the website you are building
- A Gravio account for the sensors and the IoT platform. The basic account at U$ 5/month is enough, pick 4 of the available sensors that you like to capture data.
- A software called Postman to test the API

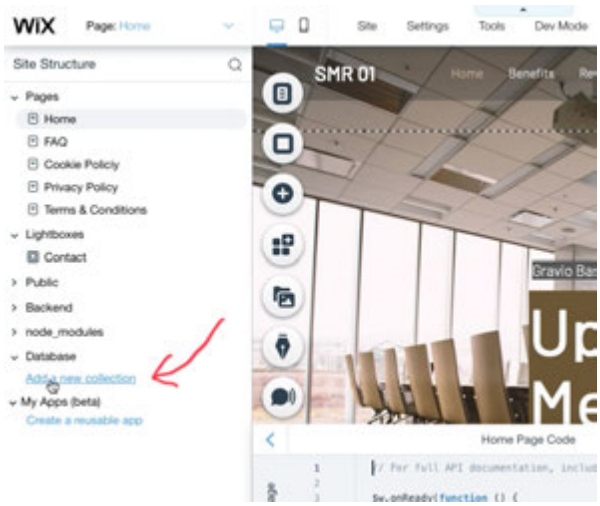
Step 1: Building the Wix API

API stands for “Application Programming Interface” and allows you to exchange data between different independent systems. You will need to create an “Endpoint” in Wix Corvid to receive the data from Gravio. Wix Corvid is the integrated application development environment of Wix to create advanced websites. The “Endpoint” is an URL,
typically *www.yourdomain.com/_functions/functionname/* where the data gets submitted via HTTP Post command.

In order to get started, you will have to enable the “Dev Mode” in Wix:

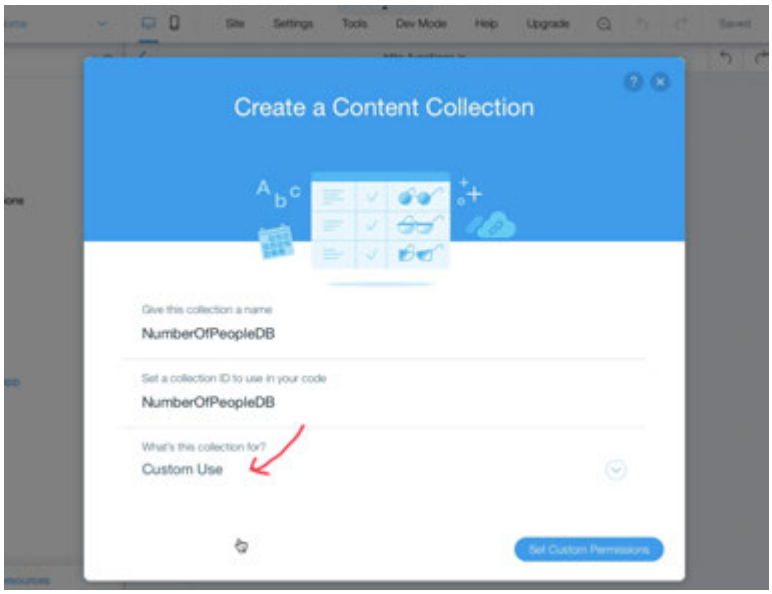


This will give you an additional menu at the left side of the screen where you can unfold the “Database” section and click on “add a new collection:

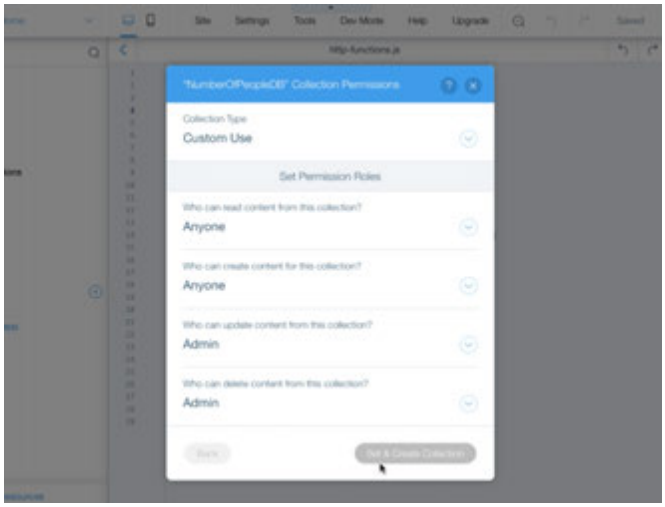


Creating the “Collection”

As a first step, you need to create a database in Wix, where the data received from Gravio gets stored. Wix calls such a data table a “Collection”. Once you clicked on “Add a new collection”, you will see the dialogue box to set the parameters. Start with a “Blank” collection and give it a meaningful name. Note, that because we will need to be able to write into the database, we need to give it custom permissions, therefore please select “Custom Use” in the last drop-down menu:



Then click on the button “Set Custom Permissions”. Here you can set it to the following permissions:



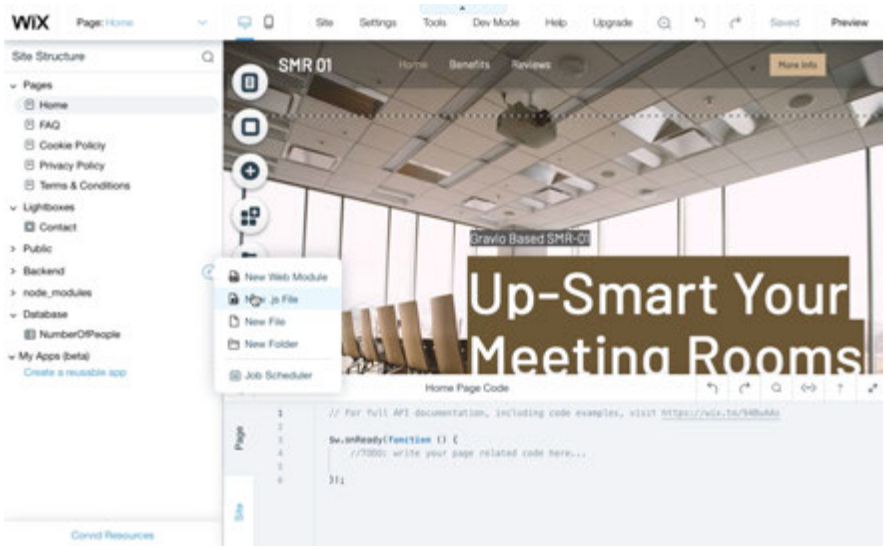
Hide all irrelevant fields and add a new field naming it something sensible that you will recognize:

In this case, we create a number column with the name “numberofpeople”:

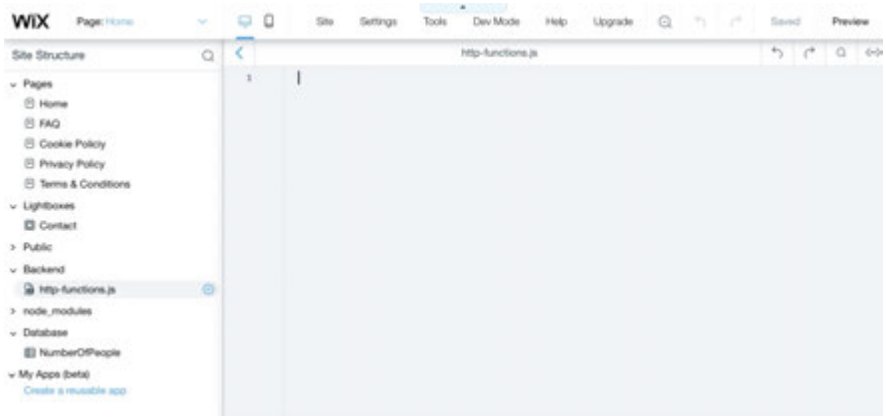
Once the collection is ready, we can start building the API.

Creating the script to push the data into the collection

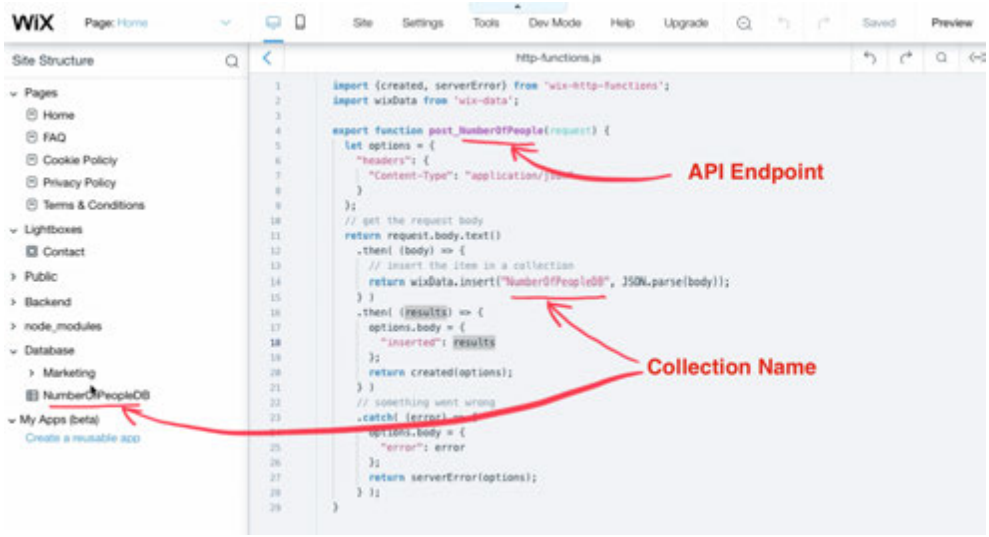
To create the actual Wix API, please find the “Backend” folder on the left and create a new js file by clicking on the + button:



Name the new file ***http-functions.js*** - it is important to call them that way as this is a reserved name for exposing external functions. The editor window will contain some prefilled code, you can delete that and start with a blank file:



Here, the names of the functions in this document determine the name of the API endpoints. You may consider choosing “hard to guess” function names, so nobody can find out the API by trying easy to guess functions. For the purpose of this tutorial, we choose an obvious name for the POST API: ***post_NumberOfPeople()***



The full code is:

```
import {created, serverError} from 'wix-http-functions';

import wixData from 'wix-data';



export function post_NumberOfPeople(request) {

  let options = {

    "headers": {
```

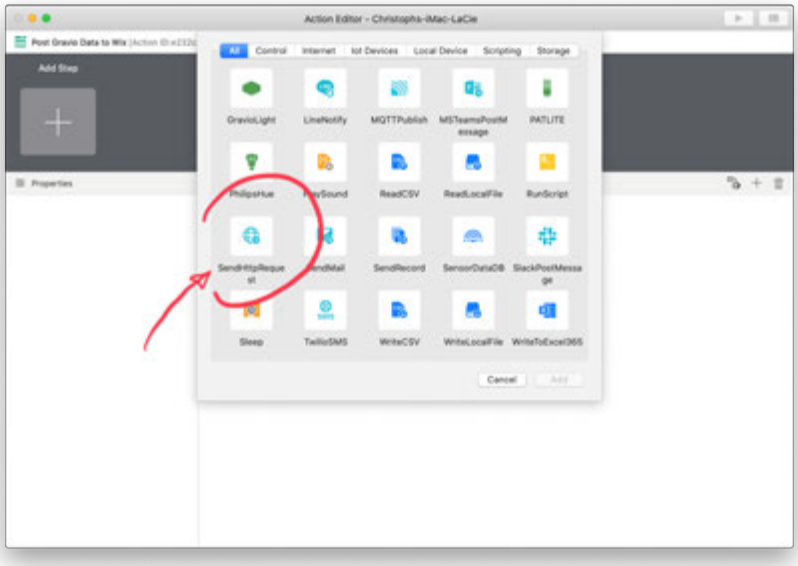
```
        "Content-Type": "application/json"
    }
  };
  // get the request body
  return request.body.text()
    .then( (body) => {
      // insert the item in a collection
      return wixData.insert("NumberOfPeopleDB", JSON.parse(body));
    } )
    .then( (results) => {
      options.body = {
        "inserted": results
      };
      return created(options);
    } )
    // something went wrong
    .catch( (error) => {
      options.body = {
        "error": error
      };
      return serverError(options);
    } );
}
```

This code takes the JSON document posted to the API and will put the data values matching the keys into the collection. This is why the column name and the name of the key in the JSON string must match. (See next step.)

More details about how to create API functions or JavaScript code on Wix Corvid can be found under https://support.wix.com/en/article/corvid-exposing-a-site-api-with-http-functions

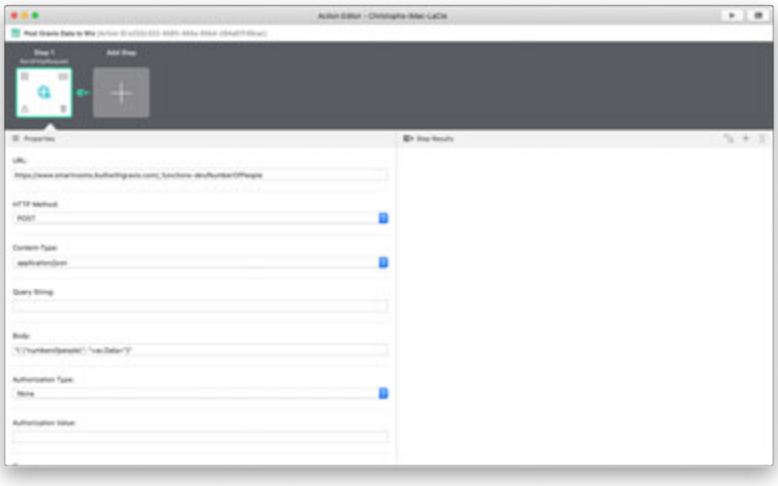
Step 2: Sending the data from Gravio to Wix

Gravio can send information from the sensors to an API using the HTTP command. In order to do that, please create an action and add the HTTP Component to a step:



The HTTP action step contains the information where to post the data to (which is the Wix API url) and what data to post there (which is the JSON string).

If you set it up, it will look like this:



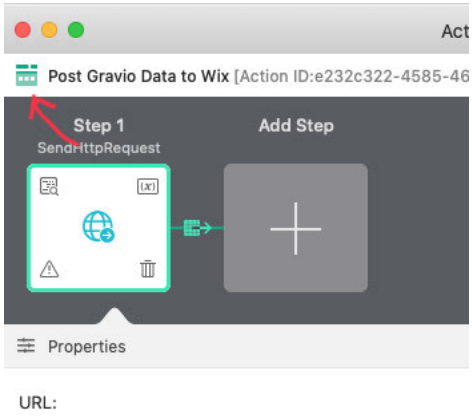
In this case, the function name of Wix is "***post_NumberOfPeople()***" and the column, in which we enter the data is called "***numberofpeople***". Hence the URL (in this case to the staging or "-dev" system) is:

```
https://yourdomain.com/_functions-dev/NumberOfPeople
```

And the JSON string is:

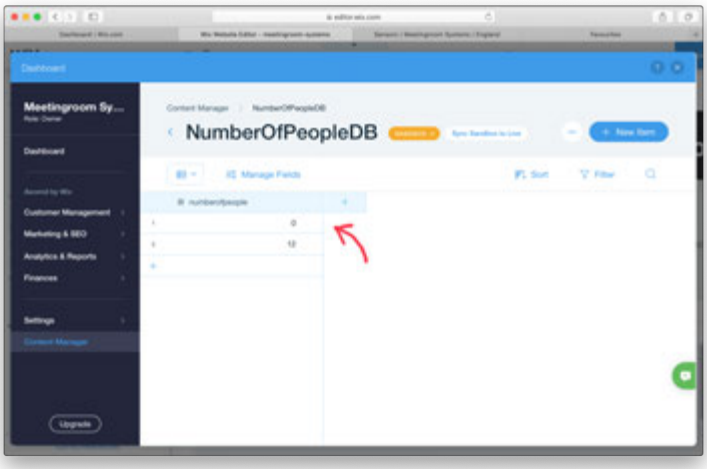
```
"{ \"numberofpeople\": "+av.Data+"}"
```

Note, we have to put the JSON string around "-marks and have to "escape" the inside "-marks with a \. The av.Data part will be replaced by the sensor data and concatenated by the + sign. To view, what variables are available within the action, you can click the action icon at the top left:



Authorization we leave blank.

If you now hit the Play button at the top right, you should see a new entry "0" in your Wix database:



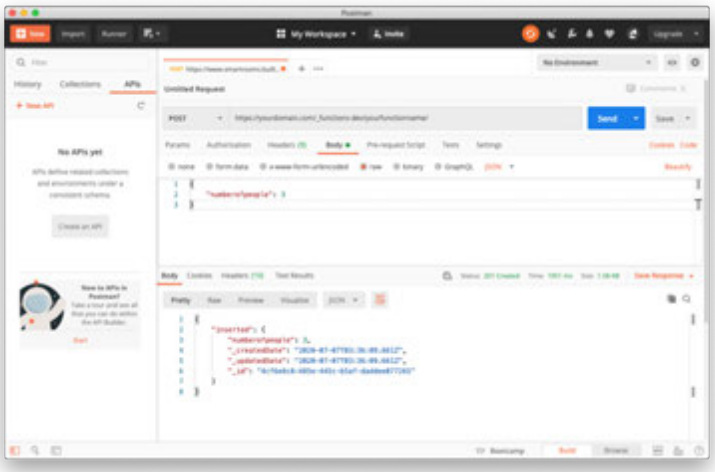
The entry is ***0*** because ***av.Data*** was empty since your action was not triggered by a sensor.

Please note that Wix uses a staging and a live environment. In the staging environment, the URL after ***functions*** will be appended with a ***-dev***. Once the site is published live on Wix, the ***-dev*** part of the URL has to be removed.

More details about how to create API functions on Wix can be found under https://support.wix.com/en/article/corvid-exposing-a-site-api-with-http-functions

Debugging the API:

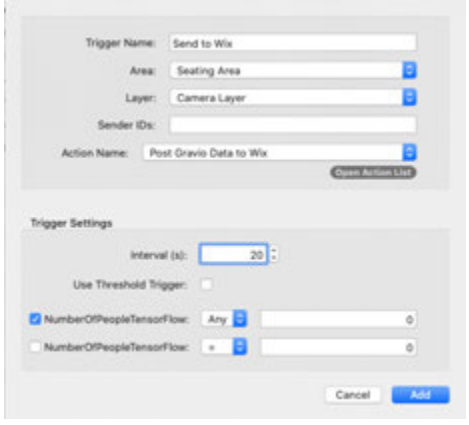
You may want to try the Wix API without Gravio to start with as sensor data may not be available all the time. For this, we use a free software called "Postman" to send HTTP Post requests. To send a test request to the Wix API, try the following settings:



This allows you to simulate Gravio sensor data submissions and is useful to debug. If you get a WD_PERMISSION_DENIED error, try to find out if you got the names of the collection and the name of the data column right. Also, check the actual permissions of the collection. If the error persists, try re-creating the table as something might have gone wrong when you created the table.

Triggering the Action from the Sensor

Now we need to trigger the submission of the data from a sensor. For this, please open the Trigger section in Gravio Studio and create a new trigger. Your trigger settings could look something like this:



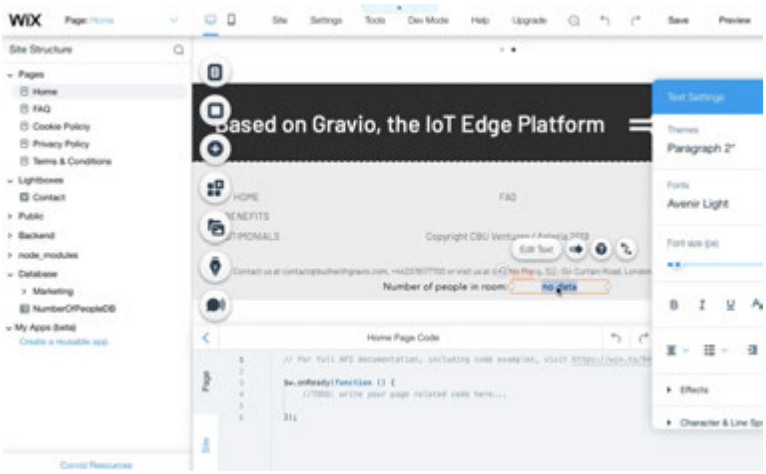
Here we look at the Camera Layer in the Seating area, and trigger the Action we have built before called “Post Gravio Data to Wix”. We do this every 20 seconds and we do this in any case, no matter what the Camera has detected.

The fields of the trigger are depending on what type of sensor you use. In this case, we’re using a Camera sensor with the human counting AI/ML Tensorflow model.

Once the trigger is set and enabled, you can test it by standing in front of the camera. It should now trigger the “Post Gravio Data to Wix” information to the Wix API.

Step 3: Displaying the Data from the Database/Collection in the Wix Front End

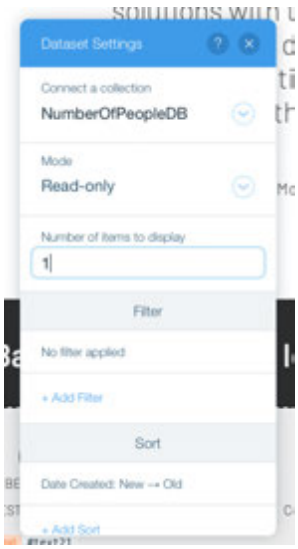
Next, we want to show on the website what the number of people in the room is. For this, we open the Wix editor and create a new text field. The contents of this text field will be replaced by the collection data. If you like some static text next to the data, you will need to create a separate field for that:



You then click on the line icon with the two dots and start creating a new dataset and connecting the content of the textbox with the collection and field:

If you scroll around your website in the editor, you will also find a new floating icon that looks like this:

This represents the data source and if you click the “Settings” button above, you can set the filter criteria for the data to be extracted from the database:



Set it to one record only and ordered by the newest. This way the Wix front end will always show the latest number on record.

Now, save your website first, and check the preview. You should see the latest number.

Going Live

If everything is ok, press the Publish button in Wix and change the Gravio API URL from

```
www.yourdomain.com/_functions-dev/functionname/
```

To

```
www.yourdomain.com/_functions/functionname/
```

By removing the ***-dev*** from the URL, Gravio will post from the test Wix site to the live Wix site.

And here you go, your Wix site is now connected to live sensor data: