Tableau Extension API

Write-back to Database

Contents

Extension API

In version 2018.2, Tableau has introduced a feature called “Extension API” which allows Tableau dashboard developers to embed third party applications inside Tableau dashboard (just like any other dashboard object) to extend Tableau’s capabilities.

The extensions are nothing but web application, built using HTML, JavaScript, PHP etc.

Write-back

Developers can create an extension which can write-back to the database and easily embed it in Tableau so that dashboard users can use it to make updates directly in databases without having to visit any other portals or applications.

Some real time scenarios where write-back is needed are:

a. When Sales People want to set their target while analyzing the Sales Dashboard

b. When Marketing teams need to change a point of contact of their leads

c. When Data Quality people need to correct or clean some required data etc.

Creating write-back extension

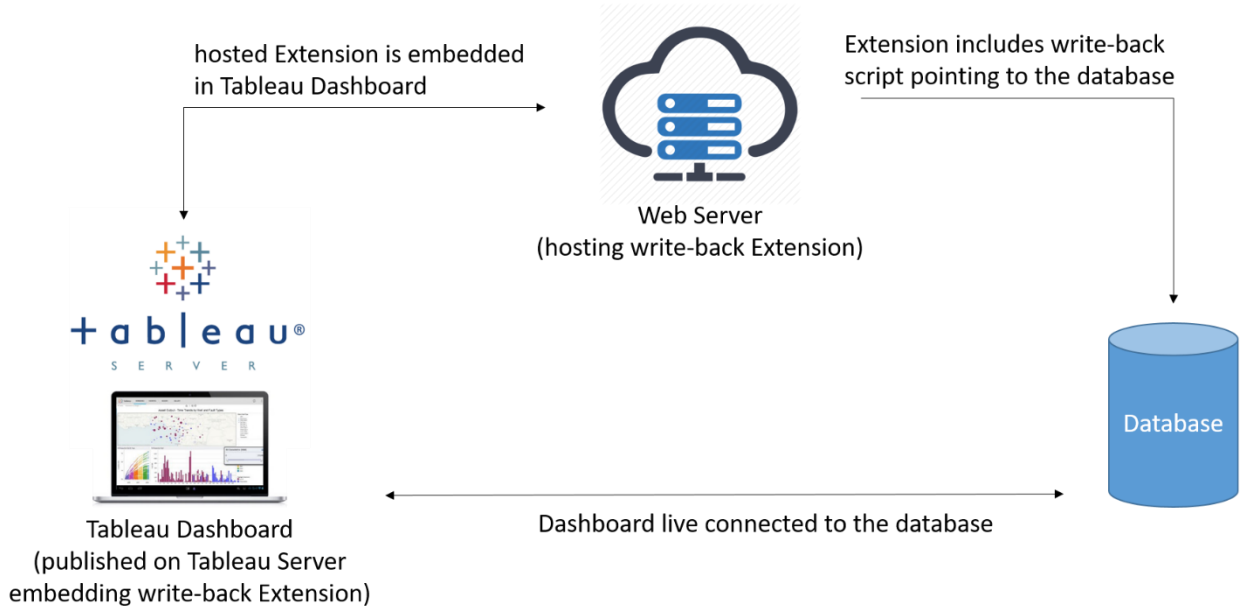
Pre-requisites

- **Tableau Desktop & Server version 2018.2 or later**
- **Understanding of web development**
    - HTML (Optionally CSS, Bootstrap) for front end designing
    - JavaScript (Optionally jQuery) for actions/events/interacting with Tableau data
    - PHP/Python/Ruby for backend processing i.e. writing back to databases
- **Web-Server**
    - To host the Extension
- **Database**
    - To which the extension will be connected for write-back operations
- **SQL knowledge**
    - To prepare update/insert queries for write-back operations

Components / Files

- **A manifest file (.trex)**
  An XML file which describes the extension & provides the information while adding onto dashboard
- **HTML file(s)**
  Front end of the extension. Can be combined with CSS, Bootstrap etc.
  Minimum one HTML file is required which should link with Extension API JavaScript library.
- **Extension API JavaScript library**
  Need to include this library to use the Tableau provided functions which provide interactivity with Tableau objects (can be downloaded from https://github.com/tableau/extensions-api/tree/master/lib)
- **JavaScript file(s)**
  For interacting with Tableau data (using Tableau Extension API), adding actions / events, linking to backend processing scripts (like PHP)

Technology Landscape



There are total 3 different servers used in the process:

1. **Tableau Server**
   On which the Tableau dashboards are published
2. **A Web-Server**
   On which the Extensions are hosted
3. **Database Server**
   On which the DBMS is running

Flow of write-back extension

Overview of steps from developer's perspective

The process starts when a user clicks on any data point/mark on Tableau dashboard that he/she wants to change the required value of -

1. Get the data point / mark selected by user
2. Get the new data value which is entered in text box or selected in drop down/checkboxes/radio boxes
3. Wait for user to click on submit/confirm
4. Once user clicks on submit/confirm, connect to the database
5. Make changes in the database (update/insert) using pre-written SQLs with received data as parameters
6. Get the success message from script after successful operation and show it to the front end
7. Auto refresh the live data source in Tableau so that user will see latest data

From user perspective

1. While analyzing the dashboard, user will want to change a value of a data field for one particular row
    a. User will select the data point/mark
    b. User enter new value for that mark
    c. User confirms the change
2. After updating the value, dashboard will automatically refresh the live data source
3. User will see new value for that data point

Example

Scenario

Let’s take a scenario where user need a write-back feature while using Tableau dashboard and how we can extend Tableau’s capability for this:

Let’s consider that a Sales team is using a Tableau Dashboard which tells the Sales People about their target customers & engagements with the targeted customers.
The “target flag” (Y/N) tells the Sales People whether the Customer is target or not.
This “target flag” of the customer is static and the users can only change this by using some third party application or request to IT team.
When users are analyzing the dashboard, at the same time, they want a functionality which can set/unset the target flag of the customers.
This allows Sales People to quickly change the target flag of customers at any point of time.

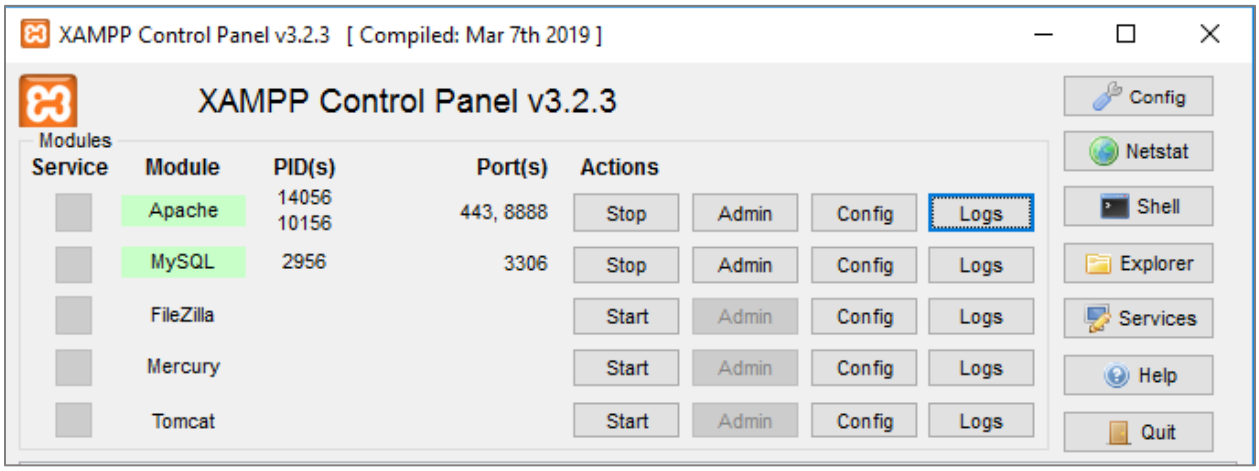
Write-back Extension Demo

A demo write-back extension is published with this. Please download and follow below steps to run it.

Running the extension

Step 1: Install a web server & MySQL Database (you can install XAMPP, get it from here https://www.apachefriends.org/download.html)
Note the installation directory

Step 2: After installing XAMPP, start Apache & MySQL services



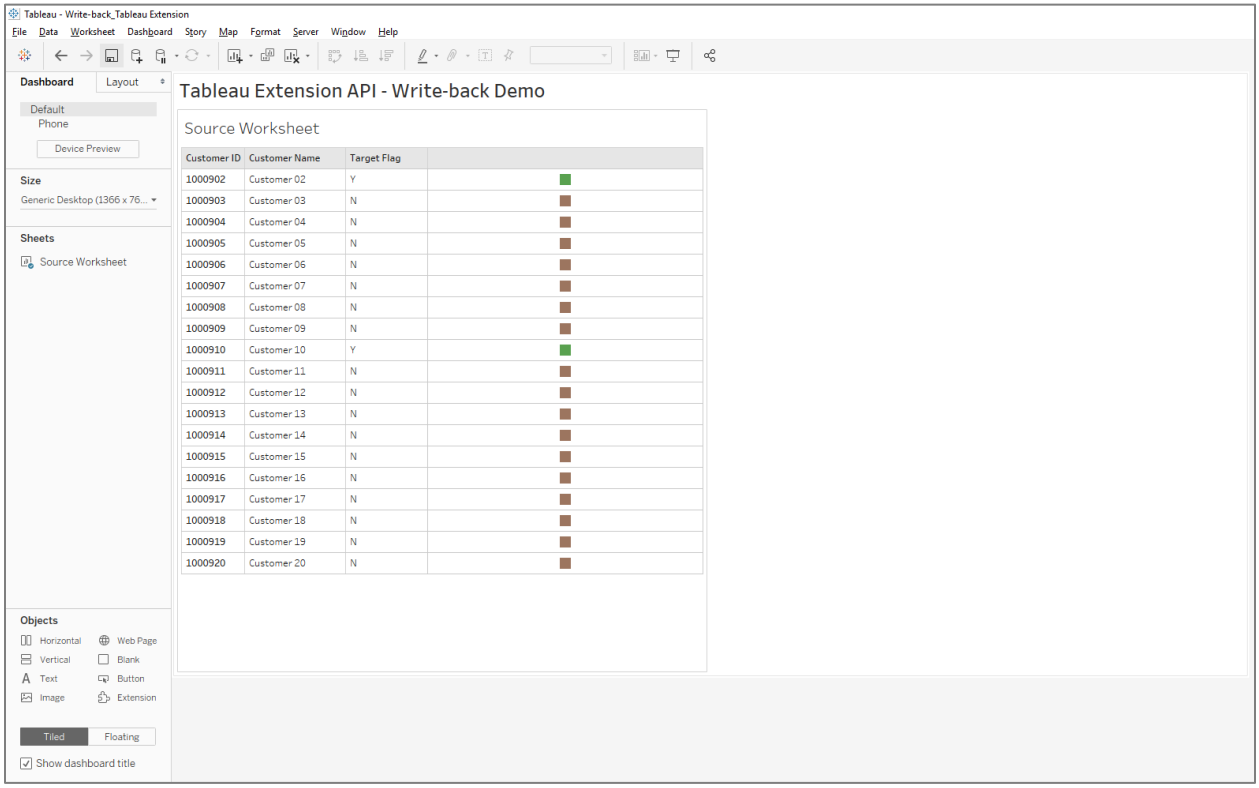
Note that Apache Server is started at port 8888. If your default port is different than this, please set your port as 8888. Follow this to change port: https://stackoverflow.com/questions/11294812/how-to-change-xampp-apache-server-port

Step 3: Once both of your services are started, go to localhost:8888/phpmyadmin which is the GUI for MySQL. Now create a new user, database & a table in it as below:

User: admin
Password: admin
Privileges: All
Database Name: tableau_poc
Table Name: target_customers (use script MySQL Table Script.sql from MySQL folder)

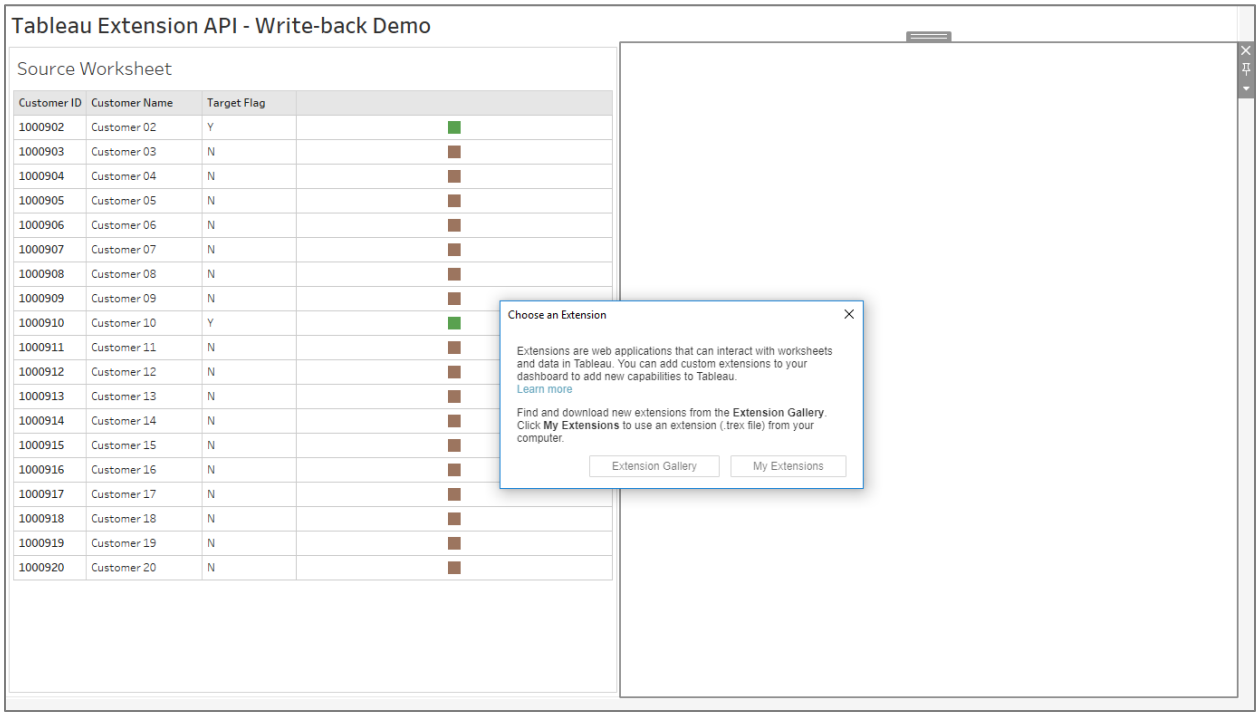
Step 4: Now go to you XAMPP installation folder and then “htdocs” folder, and unzip the given “Write-back Demo.zip” file

Step 5: Open the “Write-back Tableau Extension.twbx” file from unzipped folder->Tableau Dashboard folder (use Tableau 2018.2 or later version)

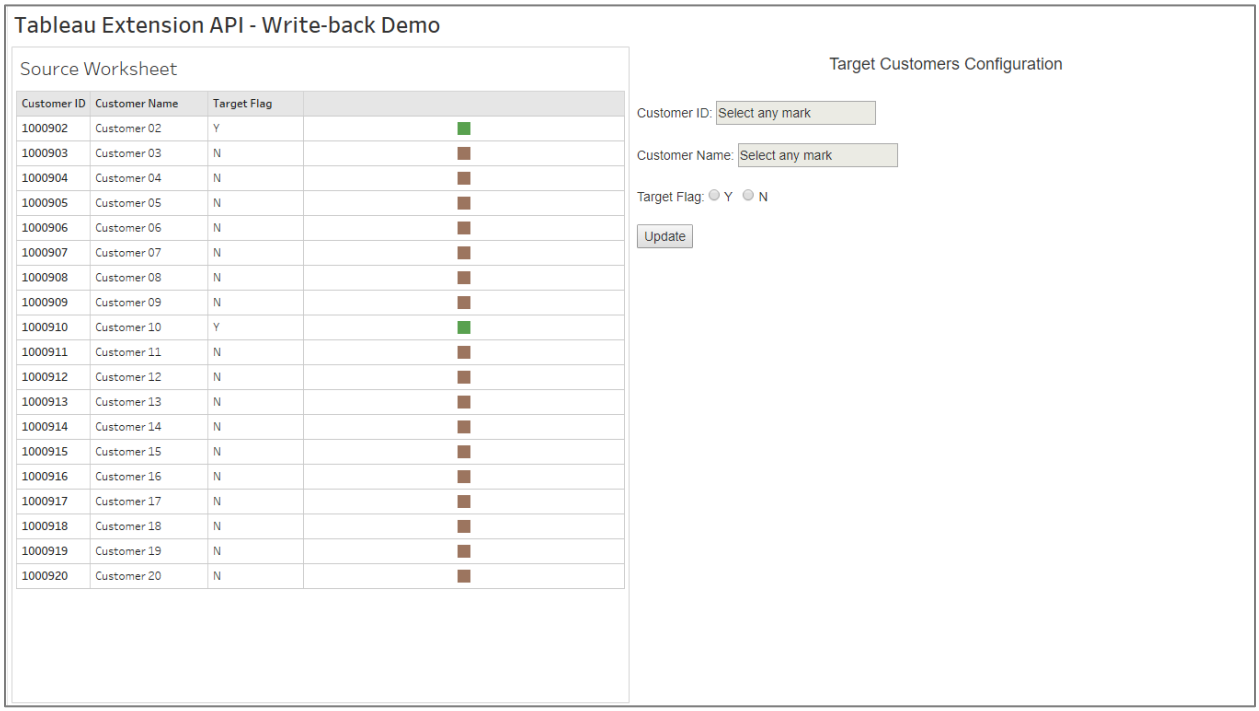


On right side, you will see a container in which you need add the Extension object

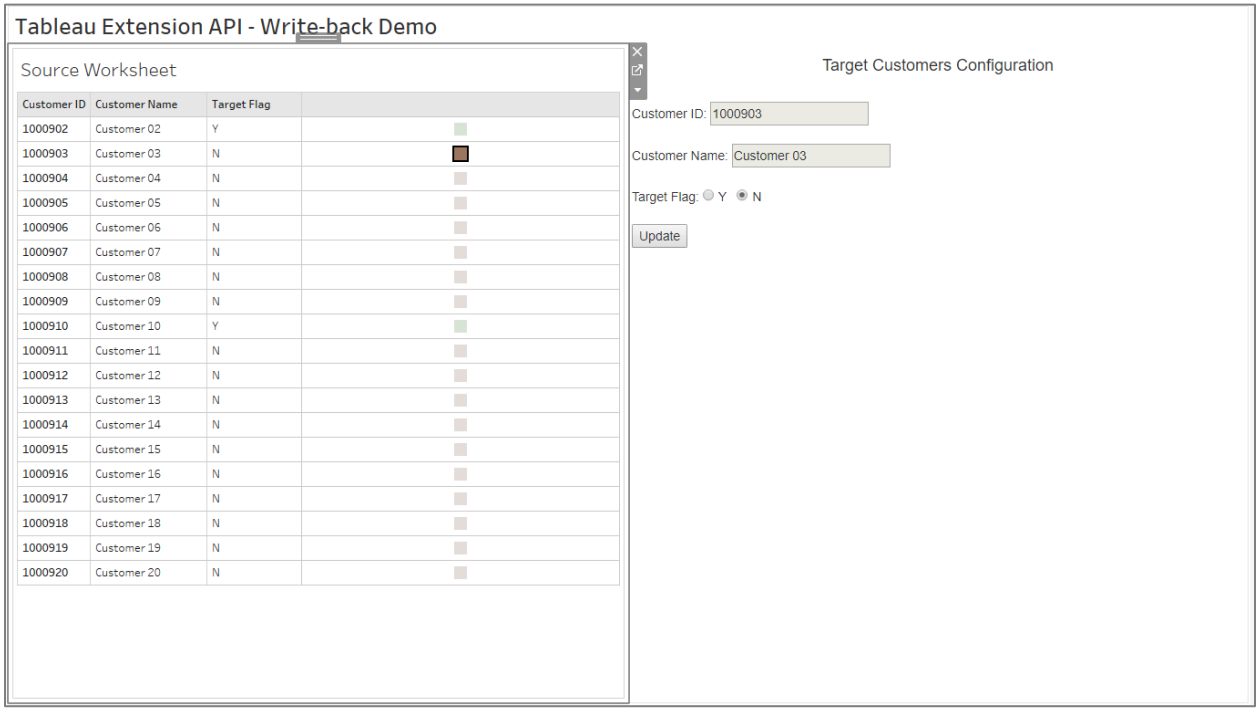
Step 6: Add “Extension” Object from the dashboard and place it on the empty container on the right side. Select “My Extensions”. After that, It will ask for .trex file. Now point to the unzipped folder location and select “WriteBack_Demo.trex” file



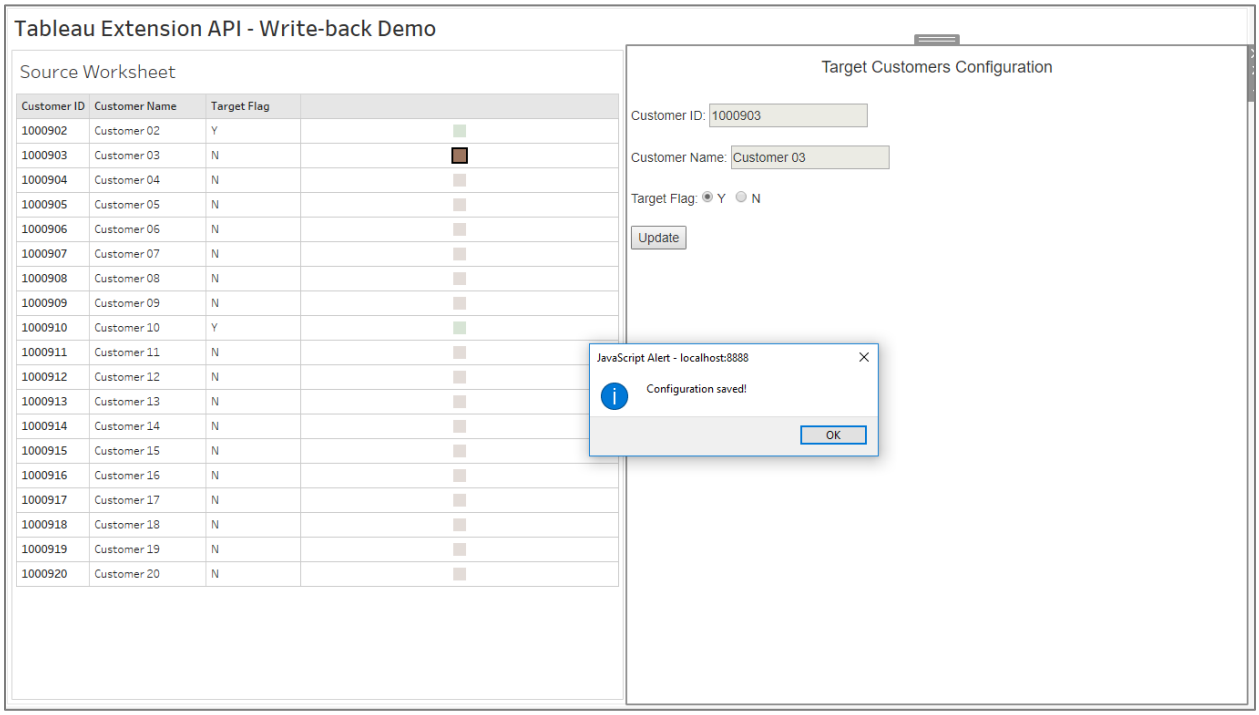
Step 7: After adding the extension, you will see dashboard as below:



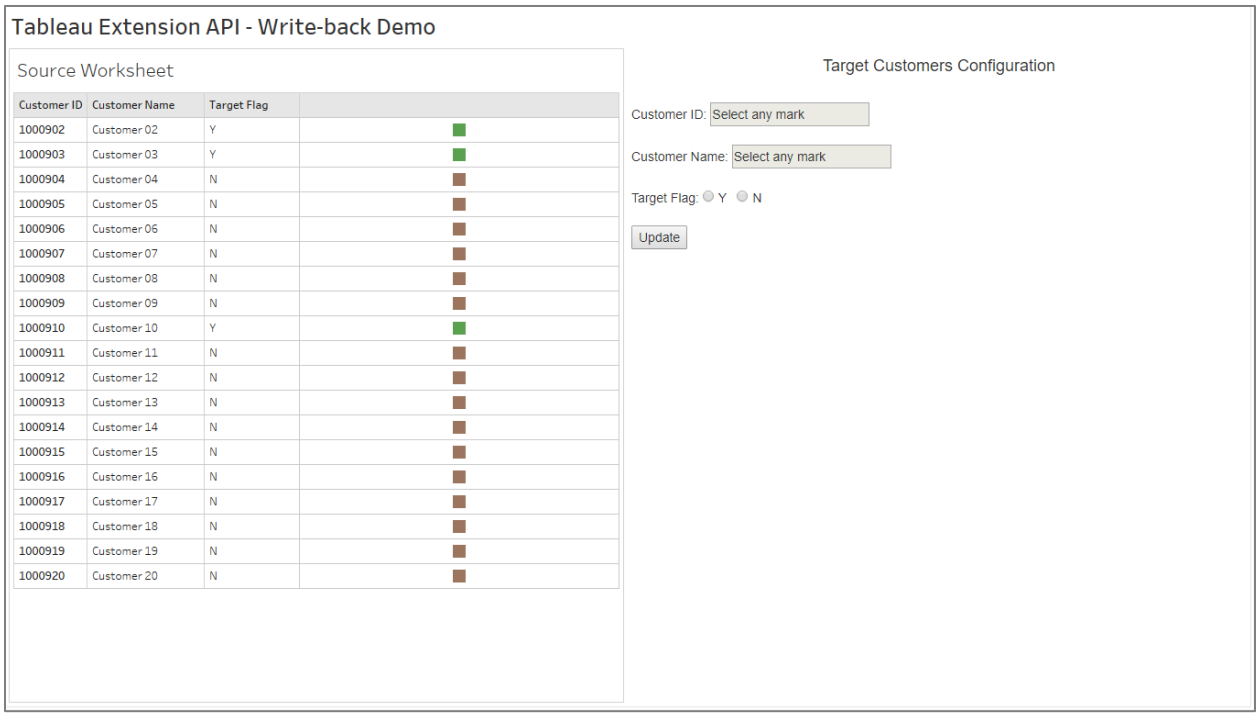
Step 8: Now click on any mark (colored square), and you will see that mark details on the right side (auto fetched by JavaScript on click)



Let's change the Target Flag of this Customer to Y (Procedure: Click on Mark, select Y as Target Flag on right side, and then click Update):



After success message, dashboard will auto refresh itself and you will see updated data:



Using the Extension on Tableau Server

All above steps are done by extension developer & dashboard developer. When you decide to publish your dashboard in which the extension is embedded, the most important thing to do from Tableau Server point of view is to allow that extension to be run on Tableau Server machine.
For this, the Tableau Server Administrator needs to safelist the extension URL and configure the extension so that it will not prompt for permission every time users will use it.

Things to keep in mind:

1. The Extension is hosted on a web-server which should be different from Tableau Server machine
2. The Dashboard Developer only has to include the extension like any other object
3. Tableau Server Administrator
    a. Must allow extensions for the server (add URL of extension to the safe list)
    b. Can enable the default policy that allows custom or unknown extensions to run in case the extension is not added explicitly in the safe list or blocked list
    c. Can configure if users should see the “Allow extension” prompt or not

Challenges / Risks

1. **Connections**
    a. Between Tableau Server & Extension hosted web-server
    The web-server should be reachable to Tableau Server
    b. between web-server & database hosting server
    The web-server on which the extension is hosted should be able to reach to the database on which it has to make updates

    The connections can be made reachable by opening firewalls in case all servers are not in same network.
2. **Validation**
    Since the write-back functionality takes inputs from business users, sometimes the values are typed incorrectly.
    Validation of the input boxes using regex / in-built functions of JS, PHP etc. is must for write back feature
3. **Hosting of extension**
    Since the extension is a web app and need to be hosted on web-server, your Tableau Server machine should have Internet connection (depending on where you are hosting the web server) and should be able to reach to the web server
4. **Data Request Methods**
    The Tableau Dashboard needs to send data to web-server and hence the proper request method (GET/POST etc.) should be configured so that the data shouldn't be read by any third party application other than extension
5. **Live updates to Production tables**
    The write-back features generally writes back to the table which is live connected to the Tableau dashboard which is little risky
    To minimize the risk, store the write-back updates to the different table with primary key and join this table with original table at Tableau end. Whenever you need to show updated data, use the columns from 2nd table.