TCS — Blockchain Tutorial 3

# Explore a Bitcoin Currency Exchange with API Endpoints

Ok, so far in the Blockchain Tutorial 1 you set up a simple MVC framework with which to implement your blockchain explorer code. In Blockchain Tutorial 2 you explored more deeply the concept of the blockchain by writing some JavaScript code, and you may have in Part 3 of this tutorial converted this JavaScript to a PHP implementation that integrates with your MVC framework.

In this Blockchain Tutorial Part 3 we are going to investigate public API endpoints to explore bitcoin currency exchanges. But, rather than go through a long winded tutorial on how to develop a front end with Bootstrap, I will give you the front end into which you can begin to intgate your results for all parts of this tutorial series, Figure 1

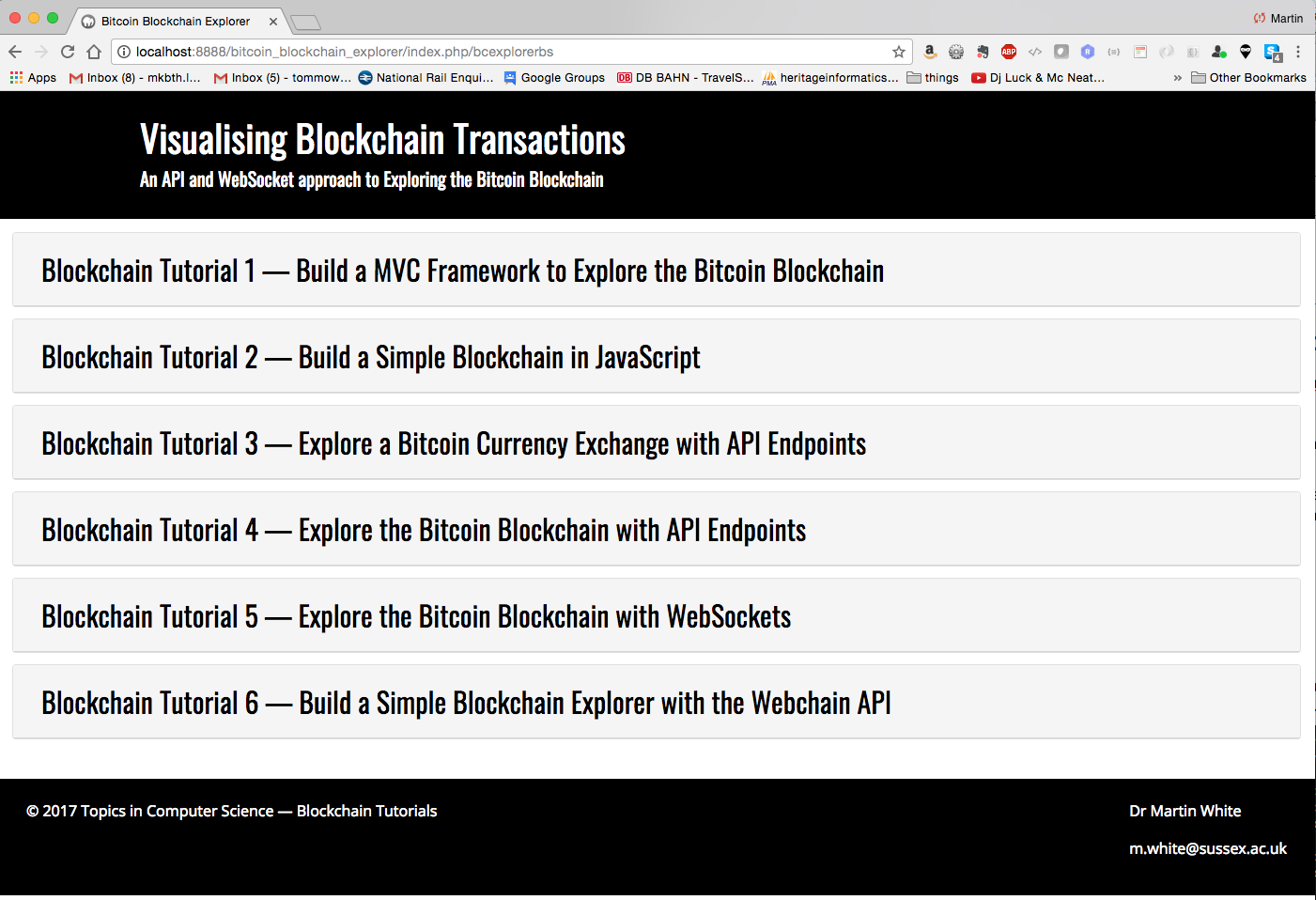


Figure 1: A Bootstrap front end for the Blockchain Tutorial Series …

Now, you have already completed Blockchain Tutorial 1, where we gave you another view (Dreamweaver) to visualise the data in the SQLite table, so all the Bootstrap view above is link to that view and back again. I am using the term visualise here rather loosely, but when, and if you get to the latter tutorials you should see that it would be a logical step to feed real-time or historical blockchain data into a graph function, for example.

To see your result for the Blockchain Tutorial 1 simply link the button BITCOIN\_DW, Figure 2, to the view shown in Figure 3. You can link up the navigation bar in Figure 3 at the same time.

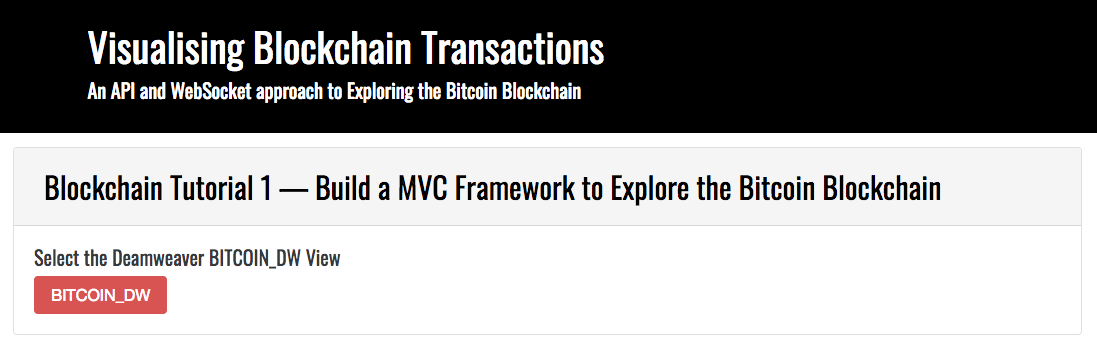


Figure 2: Bootstrap front end to link to Blockchain Tutorial 1 result

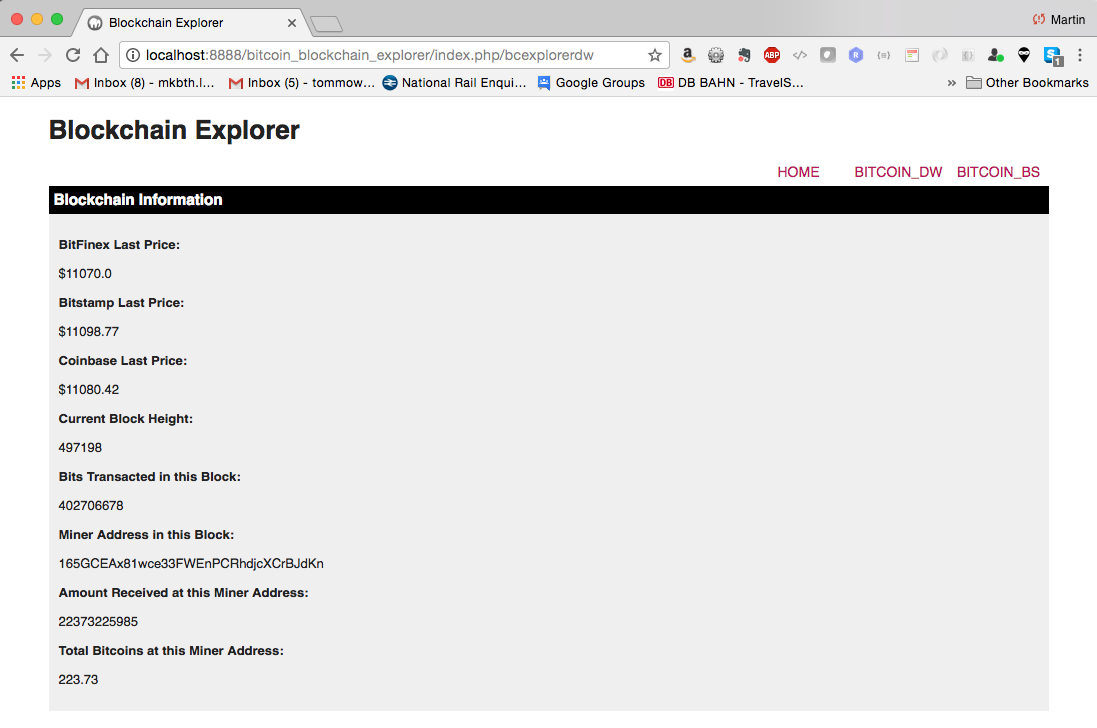


Figure 3: Blockchain Tutorial 1 result

Similarly, if you look at the Blockchain Tutorial 2 view in Figure 4, you can see it is empty because up to Part 2 of this tutorial we were looking at the JavaScript output with the console.log() function. You may recall that we suggested you convert your Part 2 JavaScript blockchain into a PHP blockchain and build a simple, very simple, wallet and simulation front end to play with it — our front end would go here.

So, before we move onto the Blockchain Tutorial 3 — Explore a Bitcoin Exchange with API Endpoints, lets make sure we have this new front end implemented in your MVC framework.

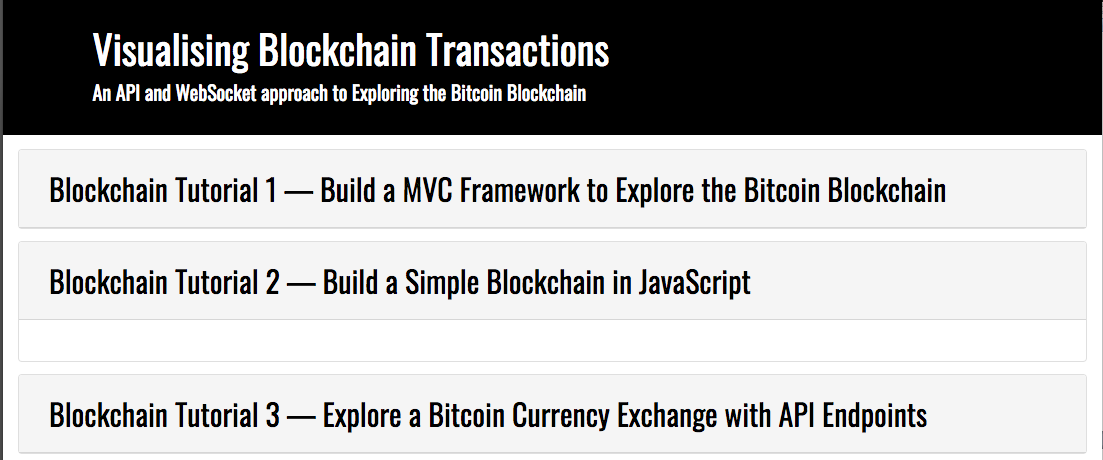


Figure 4: The view for Blockchain Tutorial 2 is empty waiting for you to use for Part 3

## Part 1 — Implement the Bootstrap Front End

You should already have downloaded the code for all these Blockchain Tutorials 1 to 6 for GitHub so that you can cut and paste elements as you go along. I am currently using Visual Studio Code as my development editor; see Figure 5, which shows the bcExplorerBs.php view highlighted. I will list all the required files needed this tutorial and highlight those you have already created, those you need to copy and those that you will update in this tutorial.

* **MVC** folder — This is your model, view, controller folder that you have already created in the Blockchain Tutorial 1. At the same level as this folder you should have the mvc.php file.
* **Controller** folder — This contains your controller.php file with your PHP controller class and associated methods.
* **Model** folder — This folder contains your model.php file with its Model class and associated methods. You may have updated these if you completed Part 3 of the Blockchain Tutorial 2. There are some other files here too: data.json, you will need to copy this from the GitHub download. This is a simple JSON file to allow dynamic update of information at the frontend view — if you look at it, it will be obvious what it does. You can ignore the bitcoin\_model.php file — that was just some procedural PHP code that I wrote to test the blockchain data access with API endpoints. You will use the same API endpoints, but we will use object oriented PHP later in this Blockchain Tutorial 3. Finally, there is also a db folder that contains your SQLite database. — again, you may have already updated this if you did the optional Part 3 of the Blockchain Tutorial 2.
* **Vendor** folder — ignore this folder, I originally installed composer as a PHP package manager, we are not using this at the moment. Associated with this is the composer.lock and composer.json files, again ignore.
* **View** folder —In here we find the bcExplorerBs.php view, which we just saw in Figure 1. You need to copy this file from the GitHub download. This file is already created for the complete set of tutorials, because we will be creating or updating new controller methods and new model methods using PHP and API endpoints in this Blockchain Tutorial 3, rather than focusing on HTML5 and CSS3 development. Where necessary I will refer to elements of this view to illustrate. Also, you already have the bcExplorerDw.php view, and the load.php, and the view\_simple\_message.php views. I can’t recall if you already have the home.php view, if not copy that over too.

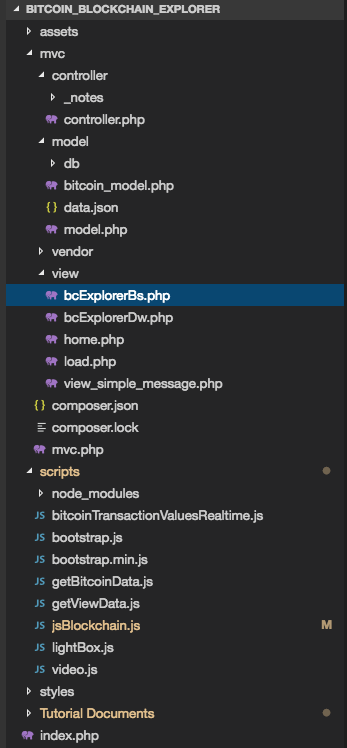


Figure 5: MVC framework structure showing the bcExplorerBs.php view highlighted

* **Scripts** folder — This folder contains all your JavaScript files and the node\_modules folder that you installed in the Blockchain Tutorial 2, when you used the JavaScript SHA256 hash function for creating the current block hash, previous block hash and mining a block. The rest of the files are JavaScript files, copy over the getViewData.js file, you can use this as it is, because we won’t have time to build it. Have a look at it anyway; it simply uses JQuery’s .getJSON() function to dynamically get view information from the data.json file in the Model folder to save cluttering up the view with hardwired text. The lightBox.js and video.js you can copy over, we might be using those later — you should note, that the iews I am giving you are based on views I created for other projects (code reuse, etc.). Obviously you will need the bootstrap JavaScript files, copy those over. You already created the jsBlockchain.js file in the last Blockchain Tutorial 2 — this is your simple JavaScript blockchain that you created to reinforce your knowledge around this subject. That leaves us one more JavaScript file for us to build as part of this Blockchain Tutorial 3, i.e. the getBitcoinData.js file. Finally, there is also a bitcoinTransactionsValuesRealtime.js file, which you can ignoire for now. We will create that in the Blockchain Tutorial 5 where we explore wWebSockets to access real-time bitcoin data.
* **Styles** folder — This folder contains all the CSS files needed to style the views. Just copy over this complete folder. I think you may already have it.

Ok, that is most, if not all, of the stuff needed for this Blockchain Tutorial 3 — Explore a Bitcoin Exchange with API Endpoints.

## Part 2 —Create PHP Model Methods with the API Endpoints

Let’s start by adding some more PHP methods to the Model class to access bitcoin exchange rate data form various bitcoin exchanges.

1. Open your model.php file in your mvc/model folder.b

## Part 4 —Create PHP Controller Methods to Call the PHP Model API Endpoints

## Part 5 — Create AJAX Requests for Bitcoin Exchange Data

## Part 6 — Connect Bitcoin Exchange Data to the View