**Step by Step create a Backbone.js project**

1. **Basic requirement:**
2. HTML 5
3. CSS3
4. Jquery
5. Bootstrap
6. **Browser support:**
7. Mozila FireFox - V.12 or upwards
8. Google Chrome – V.13 or upwards
9. Safari - V.5 or upwards
10. IE – V.9 or upwards (Json not supported)
11. **Create Folder Structure**

Like any other MCV framework backbone.js works on basis of model view and controller. So when we create a folder structure for any type of backbone.js supported project we must be aware basic three folders for Model, View and Controller. Folder structure should look like as bellow. Here we are going to create a backbone.js project naming ‘BBProject’. This is multi device and browser supported project (we are already declare list of supported browser’s ).

*Folder Structure for ‘BBProjects’*

* **project\_folder (BBProjects)** folder.png
  + **WebContent** folder.png
    - **Css (folder)** folder.png
    - **Images (folder)** folder.png
    - **Js (folder)** folder.png
      * **Collections (folder)** folder.png
      * **Models (folder)** folder.png
      * **Views (folder)** folder.png
      * **Scrpts (folder)** folder.png
      * **main.js (file)text.png**
      * **router.js (file) text.png**
    - **Json (folder)** folder.png
    - **META-INF (folder)** folder.png
    - **Templates (folder)** folder.png
      * **Dashboard (folder)** folder.png
      * **Home.html (file) text.png**
      * **Login.html (file) text.png**
    - **WEB-INF (folder)** folder.png
    - **Index.html (file) text.png**

Suppose we are creating a project for viewing the daily data-transferring status. So we create a dynamic dashboard for serving the same.

Please navigate <https://github.com/tanmoy1976/BBproject> for project’s folder structure.

Problem statement:

1. Select a date (not future date, today or past day only) from front end (UI) side
2. Select data-transfer time slot like All Day, Last 4 hours, Last 8 hours, Last 12 hours, Last 16 hours
3. In basis of date and time selection, front-end received data-transfer status and print the same visually.

Solutions:

1. Post the date and time slot as a json to db
2. Db returns a json string to frond-end
3. Chart (d3.js) showing the status visually

So the above solutions is clearly mention that its completely depends on post and receiving service between front-end and database, and Backbone.js exactly doing the same thing. Now the question is how it is doing, what is the process?

**The Process**

Model: Like any other MCV framework, Backbone.js communicate between database and front-end with Model.

**Client**

**Server**

Model View DOM

Data

Pass the data

Render as HTML