

# Data Binding in Front End for Web Applications

Egnar Özdikililer<sup>1,2\*</sup>

<sup>1</sup> Istanbul Technical University/Faculty of Aeronautics and Astronautics, Istanbul, Turkey,

<sup>2</sup> University of Telecommunications and Posts, IT Department, Sofia, Bulgaria

\*E-mail of corresponding author: ozdikililer@itu.edu.tr

**Abstract:** In this study, Data Binding is discussed. However, in Web technologies, data binding on the front end comes to the fore. That is, it should not be affected by the development of interventions to be made in a department, and it should be aimed to be freed from software. This is a requirement of the rules for a minimal plugin that will not burden the correct code, and it has to maintain itself as renewing itself. The application part is written in JavaScript. The work done is also available to be added to other applications as a library. "Real software applications" as Data Binding are written with this object for the purpose of interconnecting.

**Received** 29 November 2021

**Accepted** 19 December 2021

**Published** 22 December 2021

**Keywords:** Data Binding, jQuery, JavaScript, web application development.

## 1. Introduction

Data binding is a connector between user interface and business logic, between destination and source. Namely, it is a general method that connects two different data sources and keeps the data in sync. This synchronization can be performed bi-directionally (Two-way binding) or unidirectionally (One-way binding). One-way binding occurs as "from model to view"; Changes made in the view are not reflected in the model, data changes are only reflected in the control layer. In two-way binding "from model to view" and vice versa, all changes that are made "from view to model" are reflected in the model and database, changes made through the controller for the model and view updates the view and model.

In the scope of this study, is written a code portion in JavaScript (jquery) [1], that allows manipulating the binding data bidirectionally. If the information in any structure (for example, a web page) is used in more than one area the code portion manipulates in (change everywhere binding data) simultaneously and bidirectionally. When a change is made in a defined binding area of the web page, it updates all affected areas. Likewise, the background changes are also reflected in the front end.

## 2. Data Binding Principles

The issue of data binding has been one of the issues that makes all software developers think from time to time. It is based on the concept of object oriented, and its deep purpose is to write neat code and make it more event-modifiable. Data binding is a primary and preferred method that starts with the change of the code and allows the data to be organized together in the user interface.

It is a process that simultaneously manipulates the

information displayed at more than one point on the web page, without the need for any complex program or command line blocks in code edits. For ease of operation, a code block is converted into a procedure or script. It can be designed based on the web page, as well as the data binding method may vary, taking into account the difference in the browser used.

Nowadays, when using data binding you spend very little time work [4,6]. We can write any extension for API in our programs [3]. This extension or API generate the binded data and finished changes in very little time.

Its undisputed contribution is the ease of manipulation from a single point and the absence of deficiencies in the update process. The downside is that sql injection errors are mostly ignored.

Data binding is used for data matching between documents, from XML, Java, JavaScript etc. It is used to create, create or update elements on web pages, linking the same fields and changing them synchronously. Data Binding implementations are using for implementation presentation layer of business applications [7]. It allows sequential extraction and use of document-based working with the same data and necessary editing steps. By associating the information with a certain object class, the information can be accessed from different areas on the web page and can be updated even by the user who has the right to edit the web page.

## 3. Data Binding – JavaScript Solution

The issue of data binding has been one of the issues that makes all software developers think from time to time. It is based on the concept of object oriented, and its deep purpose is to write neat code and make it more event-modifiable. Data binding is a primary and preferred method that starts with the change of the code and allows the data to be organized together in the user interface.

It is a process that simultaneously manipulates the information displayed at more than one point on the web page, without the need for any complex program or command line blocks in code edits. For ease of operation,



a code block is converted into a procedure or script. It can be designed based on the web page, as well as the data binding method may vary, taking into account the difference in the browser used.

Its undisputed contribution is the ease of manipulation from a single point and the absence of deficiencies in the update process. The downside is that sql injection errors are mostly ignored.

Data binding is used for data matching and/or manipulating between (or in same) documents from XML, Java, JavaScript etc. It is used to create and/or update elements on web pages, linking the same fields and changing them synchronously. It allows sequential extraction and use of document-based working with the same data and necessary editing steps. By associating the information with a certain object class, the information can be accessed from different areas on the web page and can be updated even by the user who has the right to edit the web page.

Editing and updating information in data binding operations is very easy to convert from one format to another. It basically requires writing code and hence the data in the file is more easily accessed. The database-to-user interface makes the process easier when multiple user interface changes are matched in an editor-like structure. It improves user-friendliness and ability to write front end in a specific area. With this method, confusion and negative servers on the web page are avoided.

Data binding is very useful in terms of saving time in processing time and easily converting data from format to format. As mentioned before, hackers can use this tool to change format and data, and security measures should not be mitigated by using this tool carefully.

It will be very useful to prevent original sql queries at the stage of manipulating the data. In summary, model and data binding are used in applications to connect two data or information sources and synchronize data. The JSON model, which I have experienced before, is highly preferred because it provides high performance and the possibility of completion in a minimum time.

Data and user interface are associated and interacted. When the data changes, the associated elements on the screen are updated by interacting. When the elements on the screen change, the data is changed in same time using simple script. For this reason, when an element changes, there is also a change in the affected elements.

(1):

```
<input type='checkbox' id='color_id'
  bind='val:content.header;
  event.change:<?
  console.log(jQuery(this).attr("checked"));
  if(jQuery(this).attr("checked")) {
    content.header_color="red";
  } else {
    content.header_color="blue";
  }
  ?>' />
```

(1)

In the code above(1), the work to be done when the change event of the checkbox element occurs. When selected here, it sets the "content.header\_color" variable to "red". If not selected, it sets it to "blue".

(2):

```
<span
  bind='html:content.header;css.color:content.header_color;attr.al
  ign:content.header_align;'>
```

(2)

When the variable changes; "content.header\_color";, it changes the color of the text written in the "span" field above with the set color. This gives us the flexibility to: Update the elements on the screen when the data changes, or update the data when the element on the screen changes.

The Databind code plugin, which can be seen in the figure below, is arranged as JavaScript. In this context, databind.js and Data.js contents have been added. databind.js is the code that monitors the relationship between data and screen elements in the background, Data.js is the associated data code.

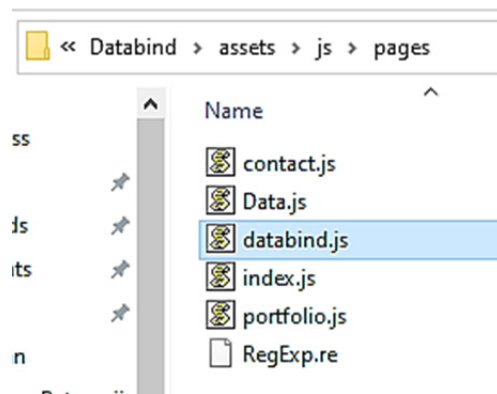


Figure 1. JavaScript Structure.

Databind.js library and Data.js (The data, theme I use inside)

In input type *Input type = 'text'* row,

(3):

```
<input type='text'
  bind='val:content.header;css.color:content.header_color;
  event.keyup:content.header=jQuery(this).val();' />
```

(3)

The relation for the bind operation is defined as above. Green field (css) command; The yellow area (color) is the parameter sent to the command (the visual value to be changed); The gray area (content.header\_color) is the information (data) with which variable this change is associated with.

Data.js> content object has header property, its default value is "EGNAR ÖZDIKILILER" color etc.

(4):

```
content={
  header:'EGNAR OZDIKILILER',
  header_color:'blue',
  header_align:'right',
  test:{a:'egnar' },
  Draw:function(){
    this.header='test';
  }
};
```

(4)

In the Figure 3 you can find the sample page with data binding areas. The binding interactive areas are marked in red. In the code below (Figure 2) we can see the code portions with and without binding parameter.

These fields are bound to the content.header property (Figure 2 - right)). Changing the INPUT field changes all fields in our system. EVENT BIND has been done to



```
});  
Databind.addHelper('val',function(obj,attr,value){ // Allows  
to change the value of an element  
$(obj).val(value);  
});  
Databind.addHelper('foreach',function(obj,attr,value){ //  
Multiple copies of an element are created with the incoming  
value array  
});  
Databind.addHelper('call',Databind.call); // Used to call a  
method (7)
```

Bind MSG function (7), defines the Databind functions used in code. These functions are fundamental for all similar programs. In the web page we allow changing CSS, adding text to sheet, change the values and attributes of elements.

The purpose of the DataBinding.js code portion is to simplify the development processes, to make the visual design fast so that people with little coding knowledge can take action quickly. The case of this study aims to make software development processes shorten and enable designers to make screens in a simpler way. The web page makes it easy to interact with the data coming from the server quickly.

#### 4. Conclusion

In every software project, after the design process, software implementation phases are completed and the concept of easy sustainability comes to the fore. Therefore, over time, common code writing and/or a single point change control in the software system becomes an essential requirement.

Regardless of the small or big changes that affect in the programs, the code portion or variable changes play an important role in the emergence of each project

because periodically changes are necessary. The aim of this study is to reflect the determined data model to the screen quickly and change shape according to the structure of the model.

#### 5. References

- [1] <https://jquery.com/> (Last Access: 08.08.2021).
- [2] <https://wrapbootstrap.com/theme/unify-multipurpose-responsive-template-WB0412697> (Last Access: 12.02.2017).
- [3] McLaughlin, Brett. *Java & XML data binding*, O'Reilly Media, Inc, 2002.
- [4] Liu, T., Lin, Y., Wen, X., Jorissen, R. N., & Gilson, M. K. (2007). *BindingDB: a web-accessible database of experimentally determined protein-ligand binding affinities*. *Nucleic acids research*, **35**(suppl\_1), D198-D201.
- [5] Flores, H., Srirama, S. N., & Buyya, R. (2014). Computational offloading or data binding? bridging the cloud infrastructure to the proximity of the mobile user, in *2014 2nd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering*, pp. 10-18.
- [6] Liu, Z., Thompson, J., Wilson, A., Dontcheva, M., Delorey, J., Grigg, S., & Stasko, J. (2018). Data Illustrator: Augmenting vector design tools with lazy data binding for expressive visualization authoring, in *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pp. 1-13.
- [7] Duarte, A. (2021). Data Binding. In *Practical Vaadin, Apress, Berkeley, CA*, pp. 111-140.

#### Acknowledgements

This paper is sponsored by Project NID-30-137#1/26.04.2021 "Modernization and improvement of UTP technology base and infrastructure for scientific research - UNICLOUD 2021" of the University of Telecommunications and Post (UTP) - Sofia, Bulgaria.