Ext Js Architecture

Introduction to EXT JS:

Ext JS provides a robust and efficient event-handling system, allowing developers to build interactive and dynamic web applications. Events are fundamental to modern web applications, enabling seamless user interactions, such as clicking buttons, hovering over elements, typing in text fields, and more. Ext JS simplifies event management by providing a structured approach to registering, handling, and propagating events within components.

This document delves into the core aspects of Ext JS event handling, covering essential topics such as event registration, propagation, and delegation. It explores how the framework utilizes the observable pattern, which allows components to fire and listen to events efficiently. Additionally, we discuss the role of event listeners, their attachment to DOM elements, and best practices for managing them to ensure optimal performance.

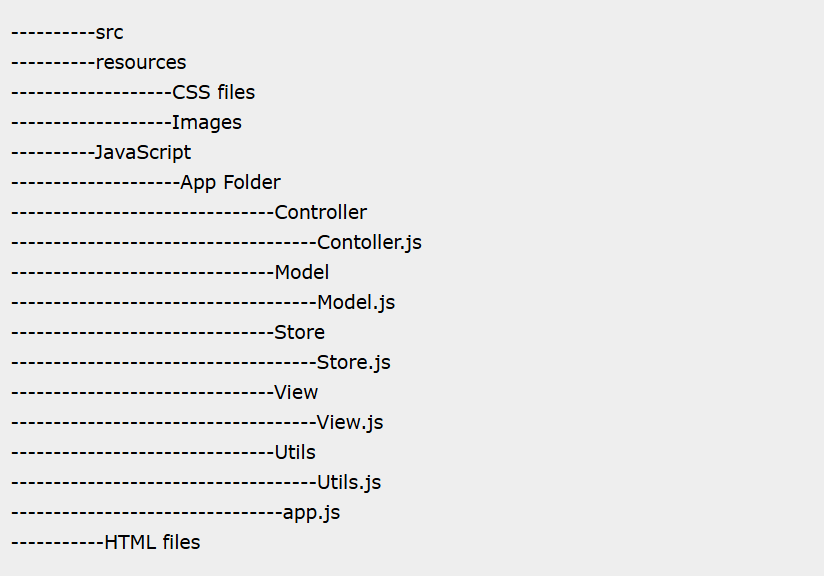
Understanding event propagation is crucial for managing event flow in an application. This document explains the bubbling and capturing phases, enabling developers to control how events move through the DOM hierarchy. Furthermore, we cover techniques such as relay events, which allow one component to listen and re-fire another component's events, enhancing modularity.

To ensure a smooth user experience, we also highlight the importance of animations in event handling. Animations provide visual feedback to user actions, making applications more engaging and responsive. By integrating animations effectively, developers can improve usability and accessibility.

**MVVM** − Model View Viewmodel (version 5)

This architecture is not mandatory for the program, however, it is a best practice to follow this structure to make your code highly maintainable and organized.

Project Structure with Ext JS App



Ext JS app folder will reside in JavaScript folder of your project.

The App will contain controller, view, model, store, and utility files with app.js.

**app.js** − The main file from where the flow of program will start, which should be included in the main HTML file using <script> tag. App calls the controller of application for the rest of the functionality.

**Controller.js** − It is the controller file of Ext JS MVC architecture. This contains all the control of the application, the events listeners, and most of the functionality of the code. It has the path defined for all the other files used in that application such as store, view, model, require, mixins.

**View.js** − It contains the interface part of the application, which shows up to the user. Ext JS uses various UI rich views, which can be extended and customized here according to the requirement.

**Store.js** − It contains the data locally cached which is to be rendered on the view with the help of model objects. Store fetches the data using proxies which has the path defined for services to fetch the backend data.

**Model.js** − It contains the objects which binds the store data to view. It has the mapping of backend data objects to the view dataIndex. The data is fetched with the help of store.

**Utils.js** − It is not included in MVC architecture but a best practice to use to make the code clean, less complex, and more readable. We can write methods in this file and call them in the controller or the view renderer wherever required. It is helpful for code reusability purpose as well.

In MVVM architecture, the controller is replaced by ViewModel.

**ViewModel** − It basically mediates the changes between view and model. It binds the data from the model to the view. At the same time, it does not have any direct interaction with the view. It has only knowledge of the model.

* **When Should You Use MVVM Patterns?**

The Model-View-ViewModel ([MVVM](https://www.sencha.com/blog/how-to-implement-mvvm-architecture-in-ext-js-web-application/mvvm/)) method is suitable for large and complex projects. It promotes better code management. This allows multiple developers to work efficiently. The Model in MVVM handles information and business logic. The View is responsible for the user interface. The ViewModel links the two and manages the application’s logic and state.

For small applications, MVVM might be overkill. The complexity it adds could outweigh its benefits. Simple projects may not need its full structure. In such cases, using MVVM could create unnecessary complications.

If your application lacks multiple data connections or advanced testing needs, MVVM is unnecessary. It can sometimes add complexity without offering enough benefits. Simple applications can be built more easily without it.

MVVM is very useful when you need to manage data binding. It allows changes in the ViewModel to update the View, and vice versa. However, if there’s no strong need for data synchronization, MVVM may be less attractive.

* **What is the Difference Between MVC and MVVM?**

Here are the key differences between the MVC pattern and the MVVM pattern:

MVC is an architectural structure that divides the app into three parts: Model, View, and Controller. MVVM separates the user interface from the app logic.

In MVC, the controller is the cross-point of the app structure. In MVVM, the View is the cross-point.

In MVC, the controller handles all user inputs. In MVVM, the input is represented through the View.

MVC typically has a “one to many” relationship between the Controller and View. In MVVM, the relationship is “one to many” between the View and ViewModel.

* **Why Use MVVM in Ext JS?**

MVVM fits well with Ext JS, a JavaScript framework for building data-centric web applications. It simplifies handling complexity.

* **Separation of Concerns**  
  MVVM separates logic, data, and the UI. In Ext JS, the Model handles data and business logic. The View handles the user interface. The ViewModel connects the two. This separation makes large applications easier to manage.

* **Enhanced Data Binding**  
  Data binding ensures that changes in the ViewModel affect the View, and vice versa. This reduces manual updates and the chance for errors.

* **Improved Testability**  
  MVVM makes testing easier. The ViewModel holds the business logic, which can be tested without the UI. This improves the ease and effectiveness of unit testing.

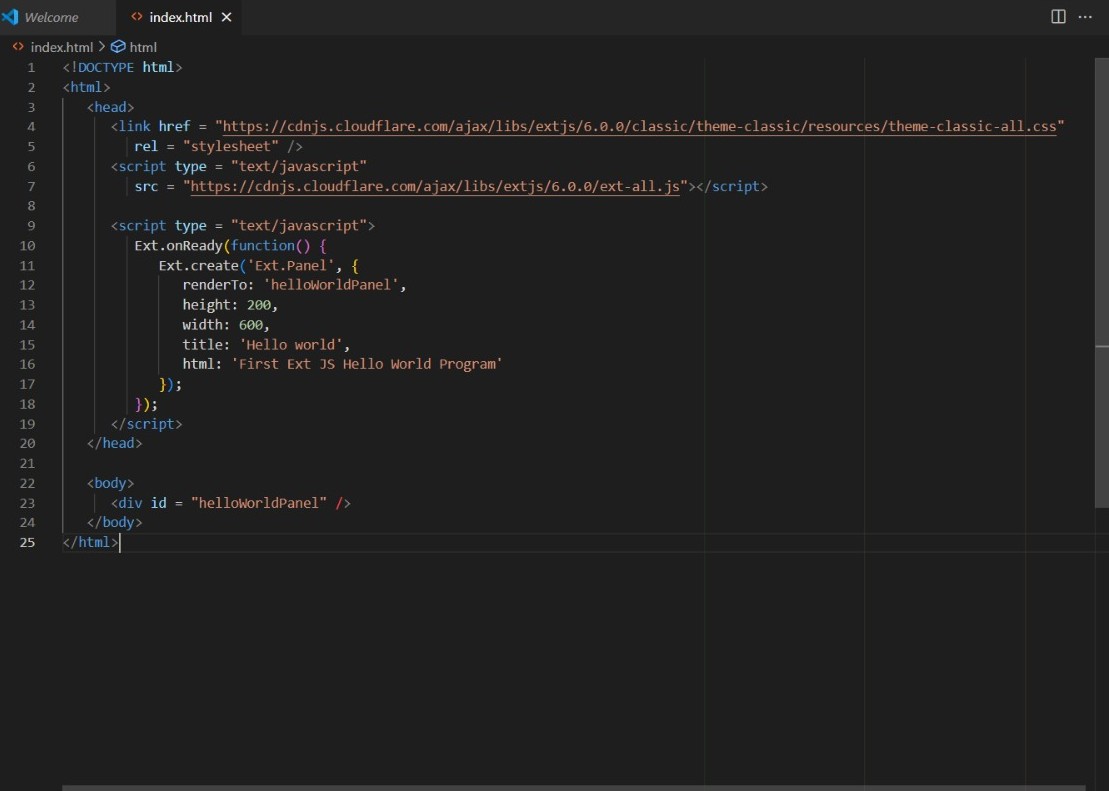
* **Scalability and Maintainability**  
  MVVM supports application growth without losing structure. It allows teams to focus on different parts of the application independently.

* **Code Refactoring**  
  MVVM enforces cleaner code. Developers can focus on specific areas without mixing logic and presentation. This makes the code more readable and easier to debug.

Coding :

Step 1

Create an index.htm page in the editor of our choice. Include the required library files in the head section of html page as follows.



Explanation

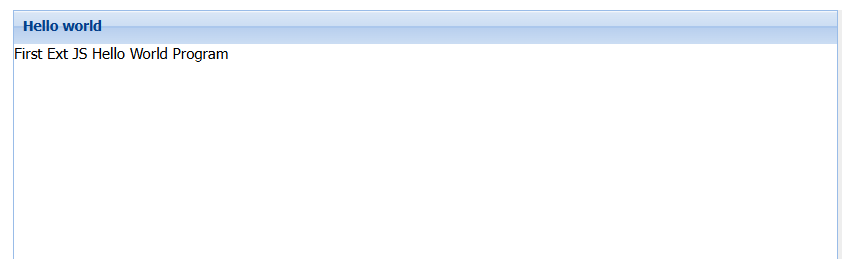
* Ext.onReady() method will be called once the Ext JS is ready to render the Ext JS elements.
* Ext.create() method is used to create an object in Ext JS. Here we are creating an object of simple panel class Ext.Panel.
* Ext.Panel is the predefined class in Ext JS for creating a panel.
* Every Ext JS class has different properties to perform some basic functionalities.

Ext.Panel class has various properties such as −

* **renderTo** is the element where this panel has to render. 'helloWorldPanel' is the div id in Index.html file.
* **Height** and **width** properties are for customizing the size of the panel.
* **Title** property is to provide the title to the panel.
* **Html** property is the html content to be shown in the panel.

Step 2

* Open the index.htm file in a standard browser and you will get the following output on the browser.



**Conclusion**

MVVM is a software development design pattern that enables the separation of the application logic from its user interface. It makes app testing and maintenance simpler and easier. Moreover, it also allows us to easily add new features to our app.

Hence, the MVVM pattern is best suited for large, complex apps, such as enterprise-level apps. And since Ext JS allows you to develop enterprise-grade apps, it makes use of the MVVM architecture to enable you to build high-performance apps quickly. This article shows how to implement the MVVM pattern in your [Ext JS](https://www.youtube.com/watch?v=3LYenkIIX8g) web app.