



Development of new features and automation framework from Silverlight to ExtJS

Manjushree Sahana V¹, UdayaRani V²

¹Student, Reva University, Bangalore, Karnataka

²Associate Professor, Reva University, Bangalore, Karnataka

ABSTRACT

ExtJs is a framework available in a scripting language that is Javascript and is used for constructing useful web applications. It uses the tools and techniques such as Ajax, DHTML, and DOM for scripting. ExtJs includes a set of GUI-based form widgets for use within web applications. Ext JS interacts with jQuery and Prototype. Microsoft Silverlight is a technology that is used in developing internet applications which can be thought similarly as Adobe Flash. Quality is one of the most critical factors which govern the business growth of any company. Software Testing is a fundamental technique that is used to guarantee the quality of the product by finding the bugs in the application. In this growing and competitive market timely delivery of a quality product plays an important role for an organization to be rated high. Therefore this makes the software testing a discipline and a time-consuming task as the applications are getting complex in nature day by day. Test automation gained importance in Industry as it can be used to avoid this situation and helps in improving the quality of the software at the expenditure of optimum cost and time demands continuous and systematic growth. The intent of this paper is to show how MVC with ExtJS has changed the patterns of web development. In this paper, we discuss MVC user interface, enhancements, performance, syntax and productive features which includes pre-built widgets, database migrations and tools for web APIs, asynchronous support and uniform responsive designs.

A Test Automation Framework is a structure where several tools and concepts are made to work together to provide a platform for automating the testing process. The RESTful web services are also known as REST API is a Hybrid framework that incorporates both Keyword-Driven as well as the Data-Driven framework, which works like a plug and play template. This Hybrid framework provides a very fast way of creating test automation and also helps users who have less programming knowledge to perform automation easily.

Keywords: Rest API, MVC, ExtJs, Automation.

1. INTRODUCTION

Complex legacy solutions are divided into small pieces which has become more elegant, simplified and easier to use the solutions today. The progress of technology is apparent in the world of web-based development where e-business is the driving force behind the rapid advancement of websites, web applications, and web services. Many progress are made in web development which has grew from static HTML pages to dynamic data-driven events. Credit for this effort can be attributed to user-friendly web development software, such as Adobe Dreamweaver and Microsoft Visual Studios (VS), which allow developers to utilize diverse and powerful tools for web programming in an easy-to-learn environment. This paper describes the two major technologies that have transformed web computing. These technologies are the Model View Controller (MVC) architecture, along with a SQL server as the database system, and the ExtJS which is a JavaScript framework. [1] An Individual can develop secure, robust and efficient applications by using these two powerful development tools. These are both standalone tools which can be used independently to develop an application. In this paper, we show how both of these technologies can be used in a single application. Our examples use the C# MVC in the backend server side with a SQL server as the database management tool. ExtJS is used on the client side.

eGRC Solutions enables the users to construct an efficient, collaborative and an effective enterprise governance, risk and compliance eGRC program across various domains like finance, IT, operations and legal areas. By the use of eGRC it is possible to automate different business processes, manage the risks and obtain a deeper insight into risks involved in the corporate and security controls. eGRC also provides out of box solutions which are built on eGRC Platform. It helps business users to modify existing solutions to cater their needs and integrate to their existing data sources without even having to compose a single line of code through its out-of-the-box solutions.

2. PROBLEM STATEMENT

eGRC platform provides business level management, covering the basic areas of business and apply the full suite of eGRC solutions to single, multiple or integrated system. The aim of this project is the business value, which is typically the key driver for an accelerated delivery of up graded product with a good quality and customer service that exceeds their expectations.

Motivation- In today's fast-moving world, it is a challenge to persistently keep up and enhance the quality continuously and effectiveness of the product that is being upgraded. An upgraded product launch with an accelerated delivery has many benefits like grabbing the attention towards the company and in turn, helps in performance enhancement of the product and company by reducing the fixes and patches. If the launch is successful, it increases the number of customers and can cover the development costs, generating profit for the company and if the product is evergreen, like something that people will need or use for a long period, then the revenue stream that starts with the product dispatch can be reliable for a long time.

3. SYSTEM ARCHITECTURE

MVC

MVC (Model-View-Controller) is a structural software pattern that combines different and various components of a web application into model, view, and controller. The ASP.NET MVC framework permits the developer to choose a substitute for the ASP.NET web forms pattern for developing MVC-based web applications. The MVC framework is defined in the System.Web.Mvc namespace and is a fundamental, supported part of the System. Web namespace.

Figure 1 shows the flow diagram of the MVC architecture. Model, view, and controller are the three different components. The user sends a request to the controller. The controller looks for the data in the model which directly communicates with the data repository. The model sends the data to the controller. The user's request are processed by the controller and it is sent back to the view in the form of response to the user query.

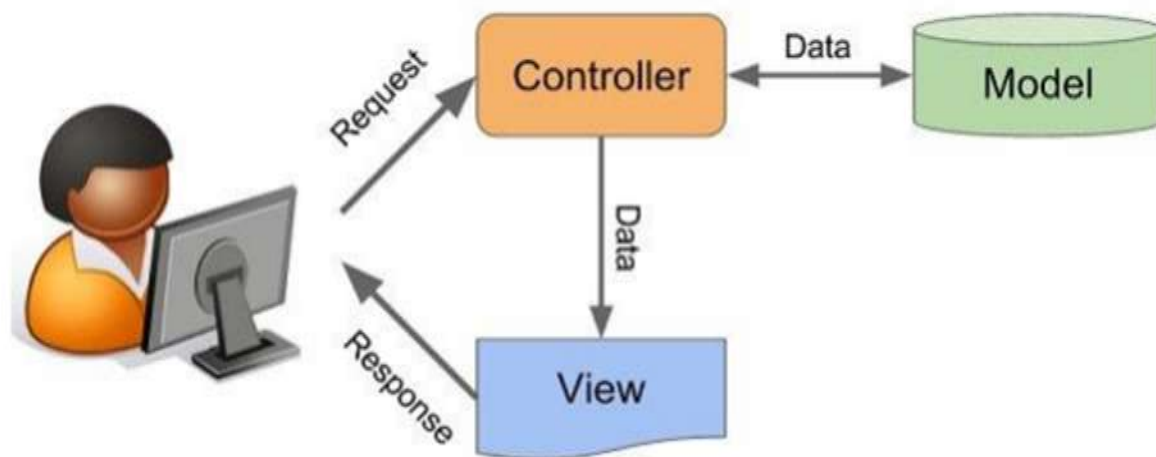


Figure 3.1: MVC flow diagram

3.1.1 Model

The model implements the logic for the application's data domain. The major purpose of the model is to retrieve and store the data to the database. For example, a book object might retrieve information from a database, operate on it, and then write the updated information back to a book table in the SQL server. The objects in the model map to the data in the SQL server. 4

3.1.2 View

Views are those components intended to display what the user requests. Typically, the user interface is created using the model data. An example would be a load, create, edit or delete view of a table that displays text boxes, drop-down lists, check boxes and the action buttons. Generally, the views are HTML pages, but in our case, ExtJS is incorporated in the views to design visually attractive web applications with pre-built and pre-tested components.

3.1.3 Controller

Controllers are the components that work with the model, handle user interaction and select a view to render those displays according to the user request. In an MVC application, the view displays what is requested by the user; the controller handles and responds to user input and interaction. [2] The MVC pattern combines different features of the application which are the business logic, input logic, and UI logic and provides a good bonding with these elements. The pattern helps in finding the location of every kind of elements. The input logic, User Interface logic and business logic each belong to the view, the controller and the model respectively which a person to focus on a particular aspect of development at a time, by which the complexity of the application is handled effectively. Example anybody can focus on the view without depending on the business logic.

3.2 ExtJS

The desired UI is rendered according to the request given by the user and view as a component is used for this purpose. There are different technologies which can be used to change the look and feel of the user interface. Hence ExtJS a highly robust, scalable and open source Java script platform is used. There are different utilities which make the

Document Object Model (DOM) manipulation and DOM traversal very stable and easy. Moreover, cross-browser compatibility is reliable. Grid, charts, tree, menu, panel, form, button, container are commonly used in ExtJS.

4. IMPLEMENTATION

The legitimacy and appropriate functionality of all the modules of the developed application is guaranteed during the implementation phase. Implementation is nothing but the recognition of technical specifications and a process of assuring that the system is operational. It involves careful preparation, inquiry and exploration of the current system and its constraints on implementation; design of methods to accomplish the changeover and an assessment of change over method apart from planning.

The implementation of the integrated architecture is divided into 5 modules based on different functionalities. The development of the modules is done using Java, MS SQL and My SQL is used to have interaction between the various services

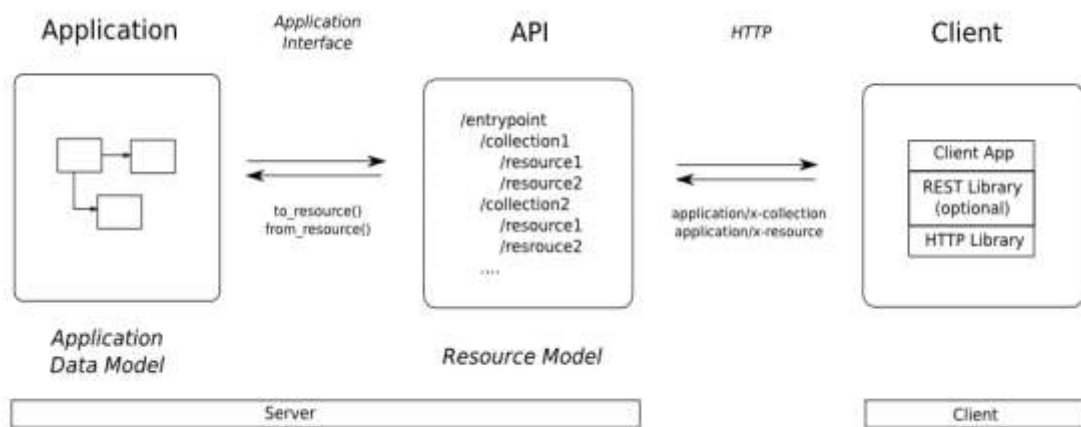


Figure 4.1: REST API Model

The Fig 4.1 talks about how API invokes a call and gets a response from the Server. Example consider I have a test case for the creation of the Global iView page. Application invokes the API from the server side which in turn gets a response from the client side.

5. RESULT ANALYSIS

Run	Manual Test Case ID	Test Level	Known Bug	Description	Driver	Test Function	Solution	Input
Yes	TC5962	Level1			Selenium Driver	Login		
Yes	TC5962	Level1			Selenium Driver	Navigate	Incident Management >Dashboard	
Yes	TC5962	Level1			Selenium Driver	Logout		

- 1) Run: Specifies whether the test needs to be run or not, if Run is set as Yes then the test will be executed else if its set to No then that test case will be skipped.
- 2) Manual Test Case ID: To uniquely identify the test case and to support sequential execution.
- 3) Test Level: To know the level of the test case.

- 4) Known Bug: To provide any related information about the bug (JIRA id's), if it is related to that test.
- 5) Description: Brief description of the bug.
- 6) Driver: This keyword specifies which driver to be called, in order to execute that test. For example: To support navigation Selenium Driver must be called.
- 7) Test Function: This specifies the different functionality provided by the driver. Depending on our requirement we need to fill this keyword.
- 8) Solution: Through this keyword, we need to specify to which application or solution or dashboards that we need to navigate, this will be described in detail while explaining the templates.
- 9) Inputs: Through this keyword, we provide the Meta data that is required while creating records etc.

6. CONCLUSION AND FUTURE WORK

The REST API presented in this thesis is a user-friendly authoring tool which has been tried and tested with various features of eGRC. So far the result is positive and encouraging. By the implementation of this framework, the product quality is realized on daily basis and quality gaps are filled at the earliest and also used for faster delivery of the upgraded product into the market. The REST API has drastically reduced the human intervention in the automation process.

Management is getting a better control over the product releases with adequate test coverage. Manual testers are also able to automate different features of eGRC within a short duration and are also able to manage the test case addition, modification, deletion. The framework is in production and continuous improvement is in progress.

The complexity of eGRC may increase rapidly as the market demands additional features. This calls for the future implementations of Automation scripts in order to test the new features that will be added into eGRC for it to sustain the growth of complexity.

The Future Implementations may include complete QC integration where in the REST API read the test cases from test Set in QC and execute them and update the results back to QC. To support multiple browser validations simultaneously.

7. REFERENCES

- [1] FENG, Y.C., Tang, Y. and Zhou, C., 2004. Theory and implementation of keyword-driven test automation [J]. Computer Applications, 8, pp.140-142.
- [2] Baker, P., Dai, Z.R., Grabowski, J., Haugen, Schieferdecker, I. and Williams, C., 2008. Data-Driven Testing. In Model-Driven Testing (pp. 87-95). Springer Berlin Heidelberg.
- [3] HP Quality Center Tutorials, [http://www.w3schools.com/HP Quality Center/default.asp](http://www.w3schools.com/HP_Quality_Center/default.asp).
- [4] Microsoft SQL Server, https://en.wikipedia.org/wiki/Microsoft_SQL_Server.
- [5] SeleniumHQ Browser Automation <http://www.seleniumhq.org>.
- [6] Eclipse <https://eclipse.org>.
- [7] JUNIT Test “ www.tutorialspoint.com/junit/junit_test_framework.html”
- [8] Selenium Web-Driver, https://seleniumhq.org/projects/web_driver
- [9] SELENIUM with JAVA, automation training material, www.doulos.com