

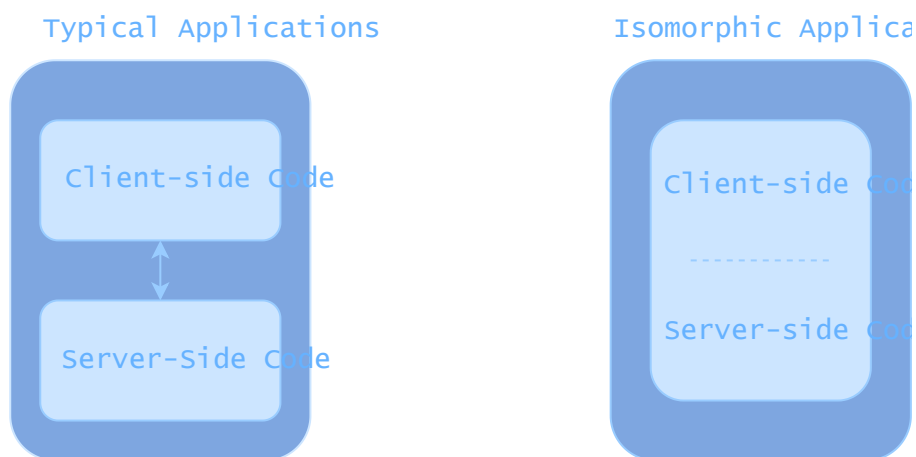
Isomorphic Frameworks

An overview of the popular isomorphic frameworks.

WE'LL COVER THE FOLLOWING ^

- Meteor JS
- Lazo.js
- Rendr

So far, we've covered both front-end and back-end frameworks, and we have learned how they significantly speed up and abate the development process. Now, we will look into a third type of framework that allows developers to avoid the cumbersome process of developing the front end and the back end separately and then integrating them; client-server frameworks. Client-server frameworks, or *isomorphic frameworks*, are frameworks that support both front end and back end development, thus allowing developers to write their entire application in a single framework without having to worry about integrating multiple components written in different frameworks.



Typical client/server frameworks vs. client-server frameworks

Meteor JS

Meteor, or Meteor JS, is an open-source client-server JavaScript framework

Meteor, or Meteor.js, is an open source client-server javascript framework that has been written in Node.js. In addition to having the benefit of being a stand-alone framework that allows for development on both the front and back end of an application, Meteor also allows for prototyping, which means that it allows for versions of the same application to be stored and tested. In addition to this, Meteor also produces code that can span multiple platforms including Android, iOS, and the web itself.

On the server-side, Meteor integrates with MongoDB and follows protocols to create a database management mechanism that enables data changes to be propagated directly to the client-side without having to write any synchronization code on both ends explicitly.

On the client-side, Meteor has its own templating engine called Blaze that allows users to choose from templates for the user interface to customize, thus making the front end development much easier. Alternatively, Meteor can also be used with some of the front-end frameworks we have already studied. It can be integrated with the Angular or React frameworks to seamlessly produce user interfaces that don't require too much work.

Lazo.js

Lazo.js is a client-server web development framework built on Node.js that has the benefit of providing front-end developers with a simple, and relatively familiar syntactical structure that they can use to create MVC structured web applications that are separated as multiple sophisticated components that come together to form the complete front-end. LazoJS has the added benefit of providing an optimized first-page load, something that is often a key determinant of user experience. These benefits are achieved using a combination of tools comprised of Backbone.js, RequireJS, and jQuery.

Lazo was initially created to address the issues that spring from creating an entire website front ends as a single unit or single page applications instead of separating the user interface into mutually exclusive components. Lazo, in turn, not only provides front-end engineers with a familiar environment for creating complete web applications but also ensures that components are separated as such to avoid overly complicated application logic. Pages are constructed via reusable, testable components that have their own life cycles, thus allowing developers to easily create complex views for the interface

while providing excellent encapsulation and separation of concerns.

Rendr

Rendr is a small library that allows you to run your Backbone.js applications on both the client and the server, depending on the specific needs of the web application you are developing. This means that Rendr provides users with mechanisms to build entire applications on either the client or server end in addition to the ability to distribute the code on both the client and server and using Rendr on both ends. Essentially, Rendr allows your web server to serve fully-formed HTML pages to any link within your application, while simultaneously preserving the feel of a traditional Backbone.js client-side MVC application.

In recent times the bulk of applications have been moving increasingly to the client side. However, this is not always suitable and may cause some problems. As a solution to this, Rendr is intended to be a building block along the way to create web applications that can be run on either side of the wire according to the specific needs of your application.

The logo for lazojs, featuring the word "lazo" in blue and "js" in orange.The logo for Meteor, featuring the word "METE" in black and "R" in red with a stylized red and black graphic element.The logo for Rendr, featuring the word "Rendr" in blue.

Isomorphic Frameworks

That covers all the different types of frameworks that are available to choose from. Now, we will move on to look into how these frameworks can be used together to build a fully functional website!