**Server:**

**Router:** One piece which has not been covered is that the router should be configurable.

The router is a function module and it has a function to search for controllers and actions (functions) of a particular controller. One way of looking for functions of names on a controller would be by using the hasProperty function provided by JavaScript.

The router can expect that the url format will roughly follow - "{controller}/{action}/?{param1=value}&{param2=value}”

Example: /Account/logonwithparam?fname=kevin&lname=cain

If the router finds a controller with the specified function, it will route the request and response to the controller’s “handleRequest(req, res, action?)” function. It is still undetermined what, if anything, the function will return.

Router error scenarios: It does not find a controller, or it finds a controller but the specified action (function) for that controller does not exist. In both cases the router lets the server know that an error has occurred and the server knows how to handle the response.

**Controller:** The controller is an extendable function object. A controller is contained in the server and the servers router calls the controller function “handleRequest(req, res, action?)”. The controller will then perform the appropriate action referenced in the request url. It is necessary for the router to call this generic function instead of the action defined in the url, because we need to remove the burden of the developer from having to define and deal with the request and response params themselves. This and the url may still need to be parsed to populate any model parameter specified by the developer.

**Model:**

**ViewPage:** A way to preload templates into memory?

**Demo:**