|  |
| --- |
| SCJ Software Solutions |
| Logging in AngularJs |
|  |
|  |
| **Sallap C Joseph** |
| **6/3/2014** |

|  |
| --- |
| How to implement the server side logging in AngularJs applications |

Table of Contents

Contents

[1. Introduction 2](#_Toc389579662)

[1.1. Purpose 2](#_Toc389579663)

[1.2. Proposed Plan 2](#_Toc389579664)

[2. Logical view 2](#_Toc389579665)

[3. Design view 3](#_Toc389579666)

[4. External packages and libraries used. 4](#_Toc389579667)

[5. Implementation 5](#_Toc389579668)

## Introduction

This document provides a high level overview and explains the whole architecture of Server side Logging in JavaScript MVC framework.

## Purpose

The Architecture document provides an architectural overview of the Server side Logging in JavaScript MVC framework AngularJS

## Proposed Plan

* 1. All the logs from the browser is pushed to the server side using an ajax call.
  2. Catch all the exceptions happening in the device/client side and push the stack trace to the server side.
  3. Separate log files will be generated for specific puposes.

## Logical view



## Design view



## External packages and libraries used.

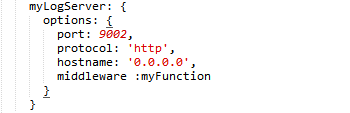
* **log4js node** - This is a conversion of the [log4js](https://github.com/stritti/log4js) framework to work with node. This package is used with node to generate the log files.

Reference**:** <https://www.npmjs.org/package/log4js>

* **Jquery** - used to trigger the ajax request to the server to write the log. The external library is used for this because even if any exception happens in the angularJs code we need to push the details to the log file

## Implementation

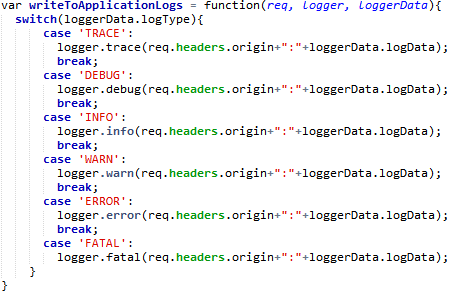
* 1. NodeJs server - A separate application is dedicated to service the server side logging. This application will be listening to the port 9002 and all the exceptions and log messages are pushed to this application.



* 1. **Server side service** – This function will accept the request url with **‘/log’** and read the data pushed from the client side and write to the log files.

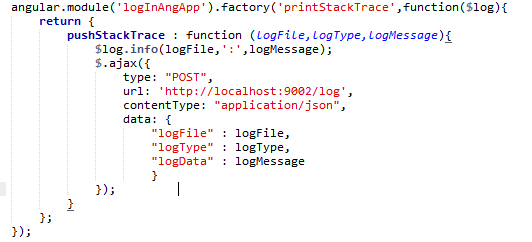




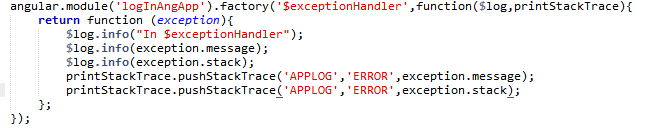


The log file configuration will be read from a log4j configuration file. 

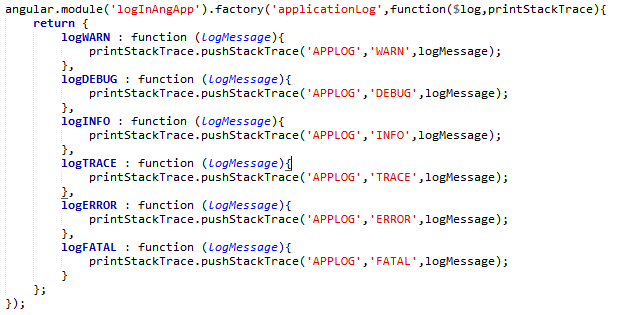
* 1. **AngularJs Factory Methods** 
     1. **printStackTrace** - This fuction will trigger an ajax call to push the log data to the server.



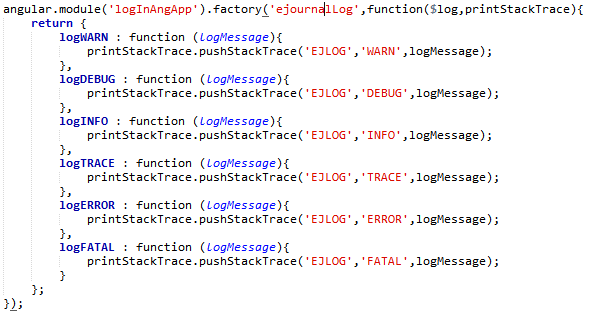
* + 1. **$exceptionHandler** - This function will override the existing AngularJs exceptionHandler method so that whenever there is an error thrown from the angular/javascript , exception and stack trace will be pushed to the server side.



* + 1. **applicationLog** – This factory will expose methods to write different levels of logs in the application specific log files

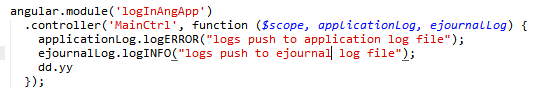


* + 1. **ejournalLog -** This factory will expose methods to write different levels of logs in the ejournallog file

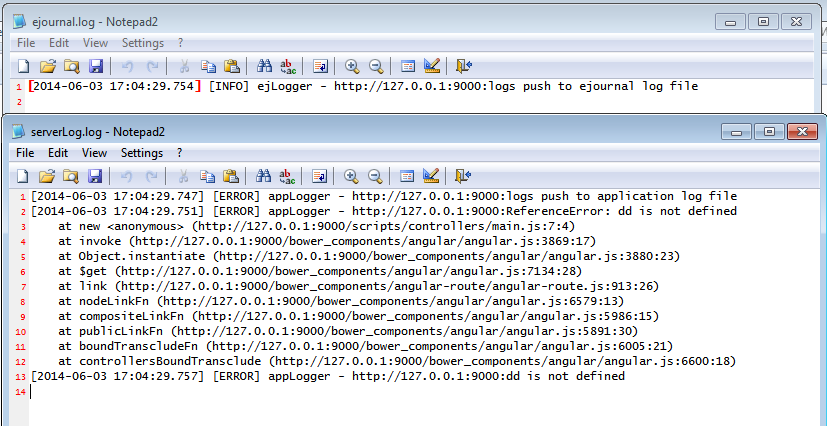


* + 1. **Calling the custom services**

Inject the custom services to the controller/services/directives and call the custom service to log the data from browser



* 1. **Output**
* **Log files**



* **Browser console**

