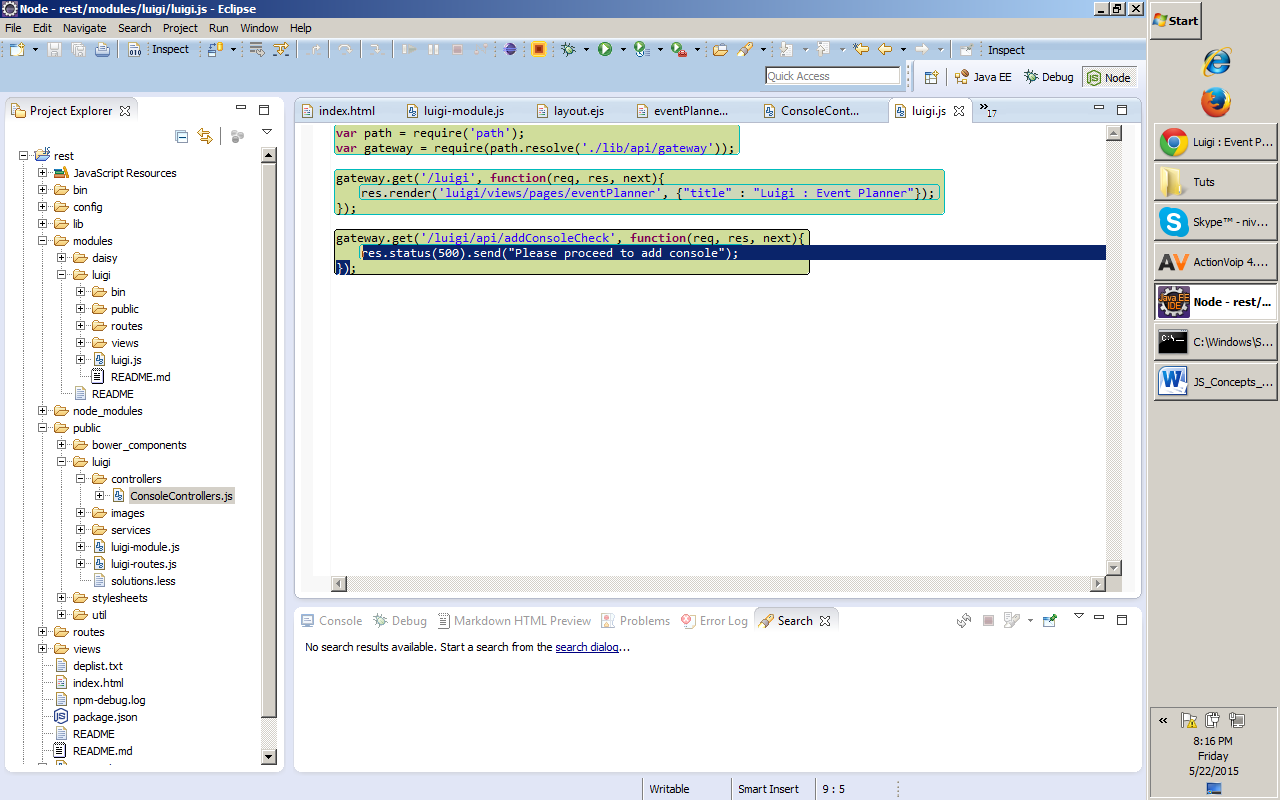
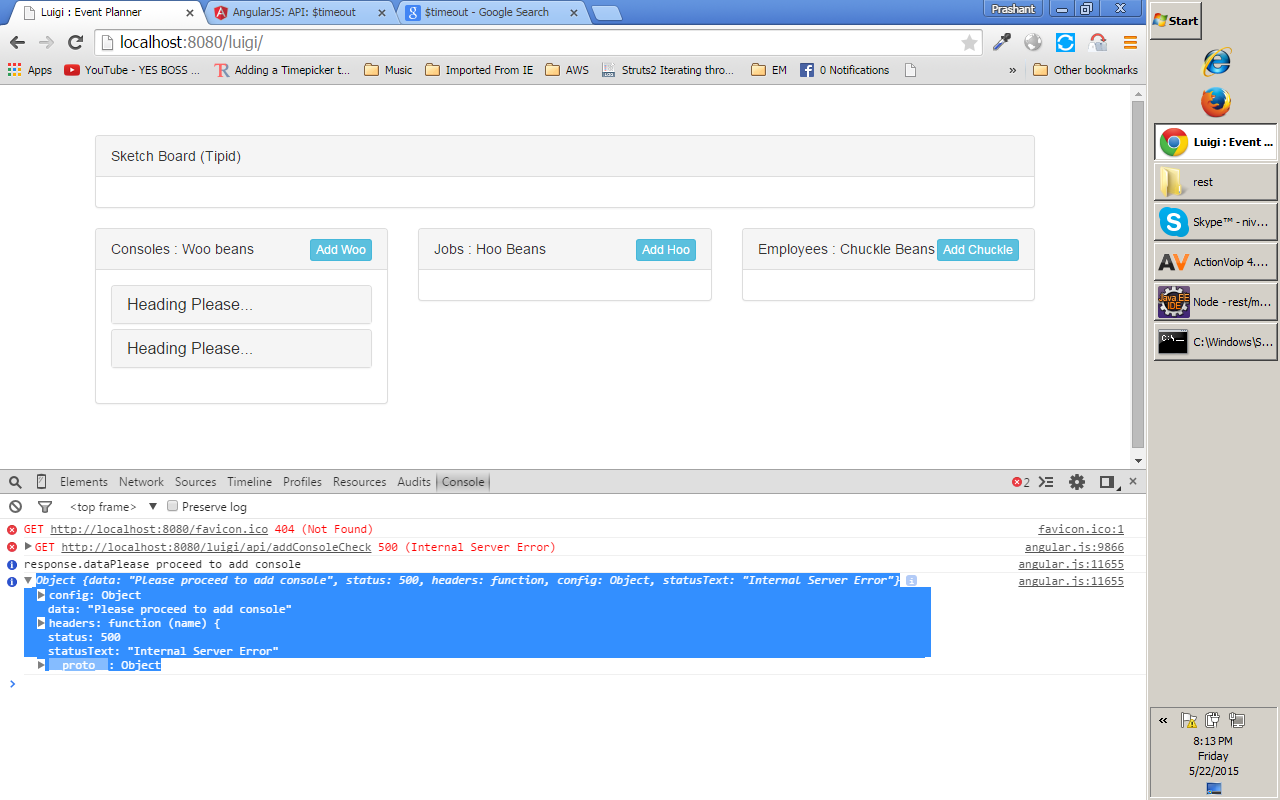
Cases :-

1. When



1. Then



// http://stackoverflow.com/questions/22785775/different-between-scope-and-rootscope

/\*

this.editContentCheck = function(){

return $resource('/luigi/api/editContentCheck').get().$promise.catch(function(res){

$log.info('The resp from EditContentCheckRestService on catch'+ res.status);

return false;

}).then(function(onFulfilled, onRejected, progressBack) {

$log.info(onFulfilled);

$log.info(onRejected);

$log.info(progressBack);

return true;

});

};

in above it will not return true or false , instead it will return as per vbelow

// from angular.js

Promise.prototype = {

then: function(onFulfilled, onRejected, progressBack) {

var result = new Deferred();

this.$$state.pending = this.$$state.pending || [];

this.$$state.pending.push([result, onFulfilled, onRejected, progressBack]);

if (this.$$state.status > 0) scheduleProcessQueue(this.$$state);

return result.promise;

},

"catch": function(callback) {

return this.then(null, callback);

},

"finally": function(callback, progressBack) {

return this.then(function(value) {

return handleCallback(value, true, callback);

}, function(error) {

return handleCallback(error, false, callback);

}, progressBack);

}

};

//Faster, more basic than angular.bind http://jsperf.com/angular-bind-vs-custom-vs-native

function simpleBind(context, fn) {

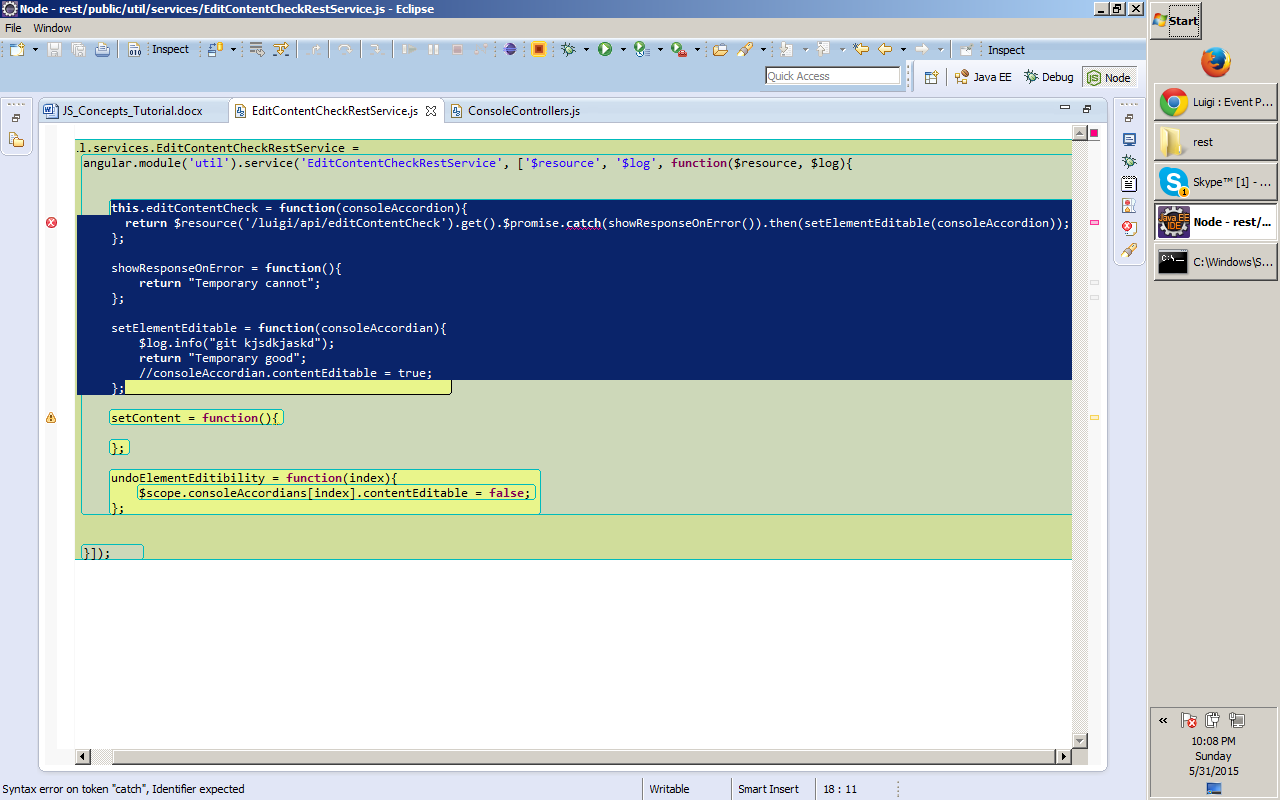
return function(value) {

fn.call(context, value);

};

}

\*/



In above it wont retun temporary good or temporary bad , else it will return as per the angular js code

// from angular.js

Promise.prototype = {

then: function(onFulfilled, onRejected, progressBack) {

var result = new Deferred();

this.$$state.pending = this.$$state.pending || [];

this.$$state.pending.push([result, onFulfilled, onRejected, progressBack]);

if (this.$$state.status > 0) scheduleProcessQueue(this.$$state);

return result.promise;

},

"catch": function(callback) {

return this.then(null, callback);

},

"finally": function(callback, progressBack) {

return this.then(function(value) {

return handleCallback(value, true, callback);

}, function(error) {

return handleCallback(error, false, callback);

}, progressBack);

}

};

//Faster, more basic than angular.bind http://jsperf.com/angular-bind-vs-custom-vs-native

function simpleBind(context, fn) {

return function(value) {

fn.call(context, value);

};

}

\*/

https://www.airpair.com/angularjs/posts/angularjs-promises