```
/** Angular app - all condensed for my repo but will seperate the files for modularity in
production **/

	var vhApp = angular.module( 'dashboardApp', [] );

		vhApp.module( 'dashboardApp' ).directive( 'weatherWidget', function() {
		return {
			restrict: 'E',
			templateUrl: 'template-parts/widget-weather.html'
		}
	} );

	vhApp.module( 'dashboardApp' ).directive( 'timeWidget', function(){
		return {
			restrict: 'E',
			templateUrl: 'template-parts/widget-time.html'
		}
	} );


/***** API Calls - Need to break out and update these if used in projects **/

vhApp.module( 'dashboardApp' )
		.factory( 'dashboardAPI', [ '$http', dashboardAPI ] )

	//contains all of our api calls from with in this object and it will be passed
	function dashboardAPI( $http ) {
		var svc = {};

		// ***************************** Scoped Functions *****************************
		svc.getOptions = getOptions;
		svc.getForecastWeather = getForecastWeather;
		svc.getCurrentWeather = getCurrentWeather;
		svc.getIpLocation = getIpLocation;

		// ***************************** Get App Options *****************************
		function getOptions() {
			return $http({
				method:		'POST',
				url:		ajaxurl,
				params:		{
					'action':	'cfg_build_json'
				}
			});
		};

		function getForecastWeather( city , state ){
				return $http( {
					method: 'GET',
					url:
'http://api.wunderground.com/api/5b12d0991f13efd1/forecast10day/q/' + state + '/' + city +
'.json',
					type: 'GET'
				} );
			};


		function getCurrentWeather( city, state ){
				return $http( {
					method: 'GET',
```

```
                                url:
'http://api.wunderground.com/api/5b12d0991f13efd1/conditions/q/' + state + '/' + city +
'.json',
                                type: 'GET'
                        } );
                };

                function getIpLocation(){
                        return $http({
                                method: 'GET',
                                url: 'http://ip-api.com/json',
                                type: 'GET'
                        } );
                };

                // Deploy
                return svc;
        }


/****  Timer App - Break out into seperate if used in projects **/

        vhApp.module( 'dashboardApp' )
                .controller( 'timeAppController', [ '$scope','$timeout', timeAppController ] );

        function timeAppController( $scope, $timeout, dashboardAPI ) {
                var time = this;

                // ***************************** Local Variables *****************************
                time.text = 'Current Time';
                time.mountain = {
                        hour : 0,
                        day : 0,
                        dayOfWeek: 0,
                        month: 0,
                        minutes: 0,
                        ampm : 0,
                        clients : [
                                'Big Chill',
                                'Clear Choice',
                                'Chipotle',
                                'Crocs',
                                'Flatirons Church',
                                'Bold Earth',
                                'Brunton',
                                'Broncos',
                                'Charity Vision',
                                'Ecospire',
                                'Grenadier',
                                'Karsh Hagan',
                                'Kong',
                                'Leopard',
                                'Smartwool'
                        ]
                };
                time.pacific = {
                        hour : 0,
                        day : 0,
                        dayOfWeek: 0,
                        month: 0,
                        minutes: 0,
```

```
                ampm : 0 ,
                clients : [
                        'adidas',
                        'Hewescraft',
                        'Lucy'
                ]
        };
        time.central = {
                hour : 0,
                day : 0,
                dayOfWeek: 0,
                month: 0,
                minutes: 0,
                ampm : 0,
                clients: [
                        'Worthy Publishing'
                ]
        };
        time.eastern = {
                hour : 0,
                day : 0,
                dayOfWeek: 0,
                month: 0,
                minutes: 0,
                ampm : 0,
                clients : [
                        'Case',
                        'Rita Hazan',
                        'Reebok',
                        'Blue Star',
                        'Life is Good',
                        'Soft Science'
                ]
        };

        // **************************** Controller Functions
****************************

        time.getCurrentTime = function(){

                //internal time functions

                //check for daylight savings time
                function stdTimezoneOffset( currentDate ) {
                        var jan = new Date( currentDate.getFullYear(), 0, 1 ),
                                jul = new Date( currentDate.getFullYear(), 6, 1 );
                        return Math.max( jan.getTimezoneOffset(), jul.getTimezoneOffset()
);
                }

                function dst( currentDate ) {
                        return currentDate.getTimezoneOffset() < stdTimezoneOffset(
currentDate );
                }

                //convert the number to a day of the week
                function dayConversion( day ) {
                        switch ( day ) {
                                case 1 :
                                day =  'Monday';
                                break;
```

```
                case 2 :
                day = 'Tuesday';
                break;
                case 3 :
                day = 'Wednesday';
                break;
                case 4 :
                day = 'Thursday';
                break;
                case 5 :
                day = 'Friday';
                break;
                case 6 :
                day = 'Saturday';
                break;
                default :
                day = 'Sunday';
                break;
        }

        return day;
}

//convert the number to the month
function monthConversion ( month ) {
        switch ( month ) {
                case 0 :
                month = 'January';
                break;
                case 1 :
                month = 'Feburary';
                break;
                case 2 :
                month = 'March';
                break;
                case 3 :
                month = 'April';
                break;
                case 4 :
                month = 'May';
                break;
                case 5 :
                month = 'June';
                break;
                case 6 :
                month = 'July';
                break;
                case 7 :
                month = 'August';
                break;
                case 8 :
                month = 'September';
                break;
                case 9 :
                month = 'October';
                break;
                case 10 :
                month = 'November';
                break;
                default :
                month = 'December';
```

```
                        break;
                }

                return month;
        }

        //time format conversion
        function hourConversion ( hour ){
                if ( hour <= 12 ){
                        hour = hour;
                } else {
                        hour = hour - 12;
                }
                return hour;
        }

        //minute conversion
        function minuteConversion ( minute ) {
                if ( minute < 12 ){
                        minute = '0' + minute;
                } else {
                        minute = minute;
                }
                return minute;
        }

        //am pm coversion
        function amPmConversion ( hour ) {
                if ( hour < 12 ){
                        hour = 'am';
                } else {
                        hour = 'pm';
                }
                return hour;
        }

        date = new Date();

        //getting the offsets for each time zone you can find those values here
http://www.timetemperature.com/tzus/gmt_united_states.shtml
        var mtOffset = 7,
                pOffset = 8,
                cOffset = 6,
                eOffset = 5;

        //subtract and hour from the offset if it's daylight savings
        if ( dst( date ) ) {
                mtOffset = 6;
                pOffset = 7;
                cOffset = 5;
                eOffset = 4;
        }

        //getting the UTC Time for correct conversion
        var localTime = date.getTime(),
                localOffset = date.getTimezoneOffset() * 60000,
                utc = localTime + localOffset;

        //setting the new timezones with the correct offset
        var mountainTime = utc - ( 3600000 * mtOffset ),
                pacificTime = utc - ( 3600000 * pOffset ),
```

```
                        easternTime = utc - ( 3600000 * eOffset ),
                        centralTime = utc - ( 3600000 * cOffset );

                //getting the current UTC time to make up for the other time zones
                var timeOffset = date.getTimezoneOffset(),
                        mountainTime = new Date( mountainTime ),
                        centralTime = new Date( centralTime ),
                        pacificTime = new Date( pacificTime ),
                        easternTime = new Date( easternTime );

                        //mountain time variables
                        time.mountain.hour = hourConversion( mountainTime.getHours() );
                        time.mountain.day = mountainTime.getDay();
                        time.mountain.dayOfWeek = dayConversion( time.mountain.day );
                        time.mountain.month = monthConversion( mountainTime.getMonth() );
                        time.mountain.minutes = minuteConversion(
mountainTime.getMinutes() );
                        time.mountain.ampm = amPmConversion( mountainTime.getHours() );
                        ;

                        //pacific time variables
                        time.pacific.hour = hourConversion( pacificTime.getHours() );
                        time.pacific.day = pacificTime.getDay();
                        time.pacific.dayOfWeek = dayConversion( time.pacific.day );
                        time.pacific.month = monthConversion( pacificTime.getMonth() );
                        time.pacific.minutes = minuteConversion( pacificTime.getMinutes()
);
                        time.pacific.ampm = amPmConversion( pacificTime.getHours() );

                        //central time variables
                        time.central.hour = hourConversion( centralTime.getHours() );
                        time.central.day = centralTime.getDay();
                        time.central.dayOfWeek = dayConversion( time.central.day );
                        time.central.month = monthConversion( centralTime.getMonth() );
                        time.central.minutes = minuteConversion( centralTime.getMinutes()
);
                        time.central.ampm = amPmConversion( centralTime.getHours() );

                        //eastern time variables
                        time.eastern.hour = hourConversion( easternTime.getHours() );
                        time.eastern.day = easternTime.getDay();
                        time.eastern.dayOfWeek = dayConversion( time.eastern.day );
                        time.eastern.month = monthConversion( easternTime.getMonth() );
                        time.eastern.minutes = minuteConversion( easternTime.getMinutes()
);
                        time.eastern.ampm = amPmConversion( easternTime.getHours() );


                $scope.$apply();

        };

        init = function() {

                $timeout(function(){
                        time.getCurrentTime();
                });

                //reset the time every minute
                setInterval( function() {
                        time.getCurrentTime();
```

```
                }, 40000 );

        }

        // ***************************** Scoped Functions ****************************
        time.init = init;

        time.init();
}



/**** Weather App - Break out into seperate if used in projects **/



vhApp.module( 'dashboardApp' )
        .controller( 'weatherAppController', [ 'dashboardAPI', '$http', weatherApp ] );

function weatherApp( dashboardAPI, $http ) {
        var weather = this;
        weather.days = [];

        // ************Scoped Variables ******************

        weather.info = {
                text : 'Welcome to the Weather App',
                currentCity : 'Louisville',
                currentState : 'CO'
        };

        weather.getIpLocation = function () {
                dashboardAPI.getIpLocation()
                .then ( function mySuccess ( response ) {
                        weather.info.currentCity = response.data.city;
                        weather.info.currentState = response.data.region;

                        weather.currentWeatherFunction();
                        weather.forecastWeatherFunction();

                } );
        };

        //Call the api to get the current weather and update the page via Angular 2 way
databinding
        weather.currentWeatherFunction = function () {

                dashboardAPI.getCurrentWeather( weather.info.currentCity,
weather.info.currentState )
                .then( function mySucces( response ) {
                        if( response.data.current_observation ){
                                weather.info.currentTemp =
response.data.current_observation.temp_f;
                                weather.info.city =
response.data.current_observation.display_location.city;
                                weather.info.state =
response.data.current_observation.display_location.state;
                                weather.info.iconUrl =
response.data.current_observation.icon_url;
                        } else {
                                $( '.not-valid-city' ).show();
```

```
                        }
                } );
        };

        //Call the api to get the 10 day weather forecast and update the page via
Angular 2 way databinding
        weather.forecastWeatherFunction = function () {
                //emptying the array if it has values
                weather.days = [];

                //making the api call for the current weather in the city and state -
Louisville CO is the default
                dashboardAPI.getForecastWeather( weather.info.currentCity,
weather.info.currentState )
                .then( function mySucces( response ) {

                        if( response.data.forecast ) {
                                var dayArray =
response.data.forecast.simpleforecast.forecastday

                                //looping through the 10 day returned array and populating
the weather.days with new daily weather object for each day
                                for( day in dayArray ) {
                                        weather.days.push( {
                                                day: dayArray[ day ].date.day,
                                                month: dayArray[ day ].date.monthname,
                                                weekday: dayArray[ day ].date.weekday,
                                                iconUrl: dayArray[ day ].icon_url,
                                                conditions: dayArray[ day ].conditions,
                                                highTemp:  dayArray[ day ].high.fahrenheit,
                                                lowTemp: dayArray[ day ].low.fahrenheit,
                                                snowTotal: dayArray[ day ].snow_allday.in
                                        } );
                                }
                        } else {
                                $( '.not-valid-city' ).show();
                        }
                } );
        };

        //On form submit run the new functions with the new values that angular has
databinded
        weather.formChangeCity = function () {
                weather.currentWeatherFunction();
                weather.forecastWeatherFunction();
        };

        weather.geoLocation = function () {
                weather.info.latitude = '',
                weather.info.longitude = ''
        };

        // **************************** Controller Functions
****************************
        initFunction = function() {
                //get the ip loction then run the other two functions
                weather.getIpLocation();

                //Update the current weather every 5 minutes
                setInterval( function() {
                        //weather.currentWeatherFunction();
```

```
        }, 300000 );
    }

    // *************************** Scoped Functions ***************************
    weather.initFunction = initFunction;

    weather.initFunction();
}
```