Coding Test

# Task

Use ReactJS or Angular to implement a web application.

The single page web application must present a form for editing a Business Entity. A Business Entity consists of a set of fields that support values in different data types (text, date).

# Deliverable

Deliver the web application in one of the following methods:

* A link to JSFiddler or Codepen
* A war file to run in a Tomcat or another application server
* A set of files ready for deployment

# Comments

If some requirements are unclear, please use your judgement and previous experience to resolve uncertainties. You can list those decisions in a readme.

# Input

Business Entity is represented by two json files:

**entityMeta.json** – defines a list of fields in a Business Entity.

* name –name of an Entity Type.
* label – displayable name of an entity type.

It’s a structure with a property “field” which is an array of field definitions. Important fields of a metadata are:

* name – unique field name.
* label – text that describes a field and must be presented on a UI
* dataType – a type of a field. It defines a widget in a form.
* system – Boolean property that indicates whether a field needs to be exposed to a user. If system=true field must be hidden.

**entityData.json** – defines values for a given Entity instance.

Each field listed in a the **entityMeta.json** file can have a value in the **entityData.json** file. This is an initial value for editing. If there is no field value in the entity data, the editor must be empty by default.

# Requirements

* A text field represented by an input box
* A date field represented by a date picker
* Create the following layout:

<Entity Type label> : <Entity Instance label> [Save button]

Label 1: <Field 1 editor> Label 2: <Field 2 editor> Label 3: <Field 3 editor>

Label 4: <Field 4 editor> Label 5: <Field 5 editor> Label 6: <Field 6 editor>

* **Save button** must produce a JSON with the following format:

{

“field1”:”new value1”,

“field2”: “new value2,

$original: {

“field1: “original value1”,

“field2”: “original value2”

}

}

* “$original” field must contain all fields modified by a user during editing of a form. Values must be the original values, read from **entityData.json**. If a field had no value originally, “null” must be specified.
* Generated JSON can be written into a session storage for simplicity or displayed in a popup.