

Test Example Requirements:

The purpose of this is to determine how quickly you can pick up and use AngularJS. There are many examples of similar Angular fragments on the web. You may ask questions of clarification, and if you get stuck you may ask for suggestions or help in the form of a suggestion of where to get clarification of the problem or a course of action that should be taken. Return the link for the Plunker to me.

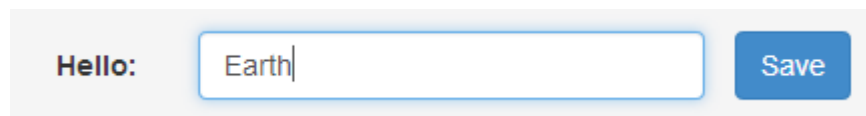
<http://plnkr.co> Angular should be used with a controller and a filter. You may use a directive if you are feeling brave (bonus points) but don't get hung up on trying to do that. The following tutorial will help a lot: <http://www.thinkster.io/pick/GtaQ0oMGII/>

The fragment should contain the following entry field.



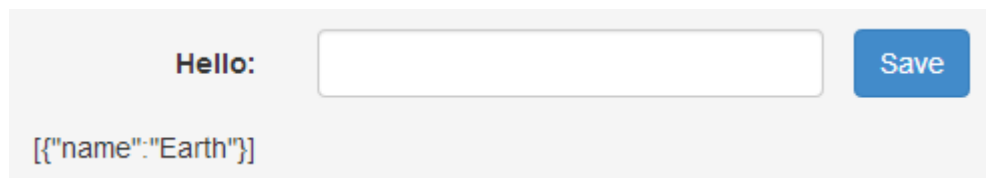
A form with a label "Hello:" followed by an empty text input field and a blue "Save" button.

The items that are typed into the Hello input should be save into an object when the save button is selected, and the input should be set back to empty ("").



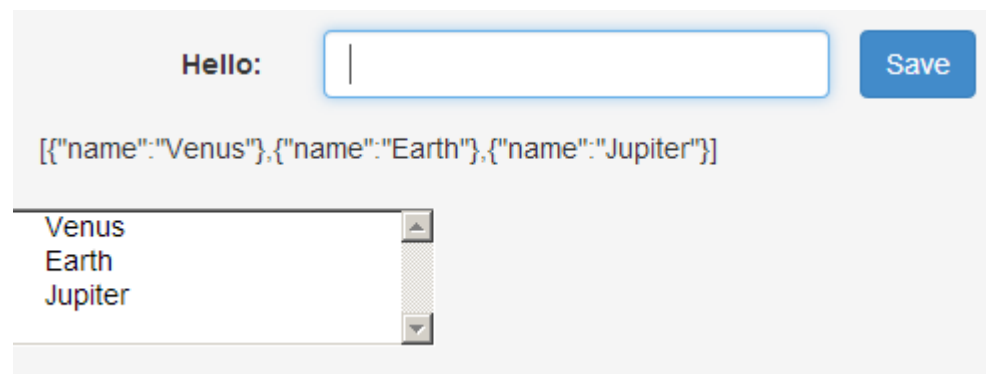
The form shows the text "Earth" entered in the input field. The "Save" button is still present.

After Save:



The form shows the input field is empty. Below the input field, the JSON object `[{"name":"Earth"}]` is displayed. The "Save" button remains.

The elements of the array of object should be used to dynamically create a multiselect control.



The form shows the input field is empty. Below the input field, the JSON array `[{"name":"Venus"}, {"name":"Earth"}, {"name":"Jupiter"}]` is displayed. Below the JSON, a multiselect dropdown menu is shown with three options: "Venus", "Earth", and "Jupiter". The "Save" button remains.

When the items of the select are highlighted it should say hello to the items selected.

Form with a label "Hello:" and an input field. A "Save" button is present. Below the input field, a JSON array is displayed: [{"name": "Earth"}, {"name": "Mars"}, {"name": "Saturn"}]. A multi-select dropdown menu is shown, containing the options "Earth", "Mars", and "Saturn". Below the dropdown, the selected values are displayed as an array: ["Mars", "Saturn"]. At the bottom, the output text is shown: "Hello Mars" and "Hello Saturn".

A custom filter should be written so that if Planets are entered then the Multi-select sorts the options in planet order Mercury, Venus, Earth, Mars, etc...

Form with a label "Hello:" and an input field. A "Save" button is present. Below the input field, a JSON array is displayed: [{"name": "Earth"}, {"name": "Mars"}, {"name": "Venus"}, {"name": "Mercury"}]. A multi-select dropdown menu is shown, containing the options "Mercury", "Venus", "Earth", and "Mars".