* Forms
  + The <lightning:recordForm> component has some limitations
    - We cannot populate any field values to give default values
    - We can’t customize the layout very much
    - This is where <lightning:recordViewForm> and <lightning:recordEditForm> come into play
  + <lightning:recordViewForm>
    - Has 2 required attributes
      * Object API name
      * Record ID
    - This component is not entirely premade for you
    - You can use <lightning:outputFiled> components within the view form opening and closing tags
    - These fields have a few attributes
      * fieldname
        + Dictates what that field will display
  + lightning:recordViewForm>
    - Very similar to view form
    - Allows you to customize the layout more and will allow you to set default vales for fields for record creation
    - Has a required attribute objectApiName
    - If recordId is probided you can edit a record and if not you will create one
    - Inside the form we can include <lightning:inputField> components that allow is to populate the value of a field
    - It has a fieldname attribute and a value attribute
      * Used to determine auto-populated values
    - With lightning button components, if you set the type attribute to submit, it will create a button that save automatically
    - If type is reset, then it will automatically clear the form
  + <lightning:recordData>
    - For when you need the most amount of customization
    - You have to build the form and the functionality ourselves
    - No required attributes
    - Parameters that we should specify that are determined by the purpose of the componenet
      * Display: we must use the recordId attribute
        + Databind an aura attribute to the targetFields attribute
        + Give a value for fields or layout type parameters
      * Edit/create
        + We are going to have to create our own actions in combination with an apex controller to be able to fully interact with the database
* Handling errors in aura
  + Javascript also has try, catch functions
  + For apex errors. It will display a pop up saying that something went wrong
    - To get more helpful error message you should throw the AuraHandledException with your own custom message inside of it
* Security in aura
  + Lightning locker
    - Client side security is mainly handled by lightning locker
    - It is automatically added to all of our aura components
    - Forces JS to run in strict-mode
    - Limits DOM access
    - Prevents access of unsupported JS methods
    - Restricts the use of the eval function
    - Places HTML DOM elements and global objects in secure wrappers