

Aura Component Controller.js – Detailed Explanation -----

This JavaScript controller is for an Aura component with a lightning:datatable that loads Contacts, allows inline editing, and lets the user save or cancel changes.

1) doInit – Set up the table and fetch data ----- - Define columns for the datatable: First Name, Last Name, Phone, Email (editable), and Account Name (read-only). - Calls the Apex method getContacts to retrieve a list of contacts from the server. - Processes the returned contacts to flatten Account.Name into a simple AccountName field for display. - Sets v.contacts with the processed list. - Uses \$A.enqueueAction(action) to send the request.

2) handleSave – Persist inline edits ----- - Reads pending edits from the datatable's draftValues parameter (only includes Id + changed fields). - Calls the Apex method saveContacts with the updated data. - On success: * Clears v.draftValues to remove edit markers. * Calls doInit again to reload the data from the server.

3) handleCancel – Discard changes ----- - Clears v.draftValues to remove edits. - Calls doInit to reload fresh data from the server.

Apex Controller (Implied) ----- Example: @AuraEnabled(cacheable=true) public static List getContacts() { return [SELECT Id, FirstName, LastName, Phone, Email, Account.Name FROM Contact LIMIT 200]; }

@AuraEnabled public static void saveContacts(List updatedContacts) { update updatedContacts; }

Key Points ----- - Flatten related fields for display (Account.Name → AccountName). - v.columns, v.contacts, v.draftValues must be defined as attributes. - Always refresh the table after save or cancel to match the server data. - Consider adding toast messages and spinners for better UX.