# Title: Account failure to load after save of Module Customizations(Sticky fields)

**History of sticky field implementation**

The custom sticky fields as JSON are stored in the configs  column module customization table. The only way to change sticky fields was through a rake task. We appended the sticky ‘true’ flag in the filter\_options API response.

However, recently we made changes to generate filter\_options on frontend for Leads As a part of the change, we included in configs of module customization in the ember model. This was done to identify sticky fields on the frontend. If a save was triggered for module customization in the UI, sticky fields of module customization would be converted to hash. This would cause an issue, since the module customization serializer on backend, expected the sticky fields to be in JSON format. This dormant issue did not occur, since the customers using custom sticky fields did not trigger save for module customization.

**Cause of incident:**

There was a new ask to display the sticky fields in specified order. Deal has default sticky fields namely deal\_pipeline and deal\_stage. The default sticky had to be displayed above the custom sticky fields.

The sticky fields would be sorted in the frontend, in the same order in which configs were sent in the module customization API response. The default sticky fields were appended to the custom sticky fields in the module customization serializer on the backend. The default sticky were sent to the frontend, regardless of whether an account had custom sticky fields configured. Due to this change, all the accounts had default sticky fields sent to frontend.

This caused the existing dormant issue to occur, since the custom sticky were configured only for few accounts. However, due to new changes, all the accounts had default sticky fields sent in the response, making the probability of the issue occurring much higher.

**Impact of the incident:**

Some of the customers triggered save for module customization and module customization configs were converted into hash. This caused the module customization serializer, to throw error since it expected configs in JSON format.

**Fix for the issue:**

The configs of the module customization were sanitized both on backend and frontend. As of now, we cannot change the configs from the UI, so we can sanitize the configs param.