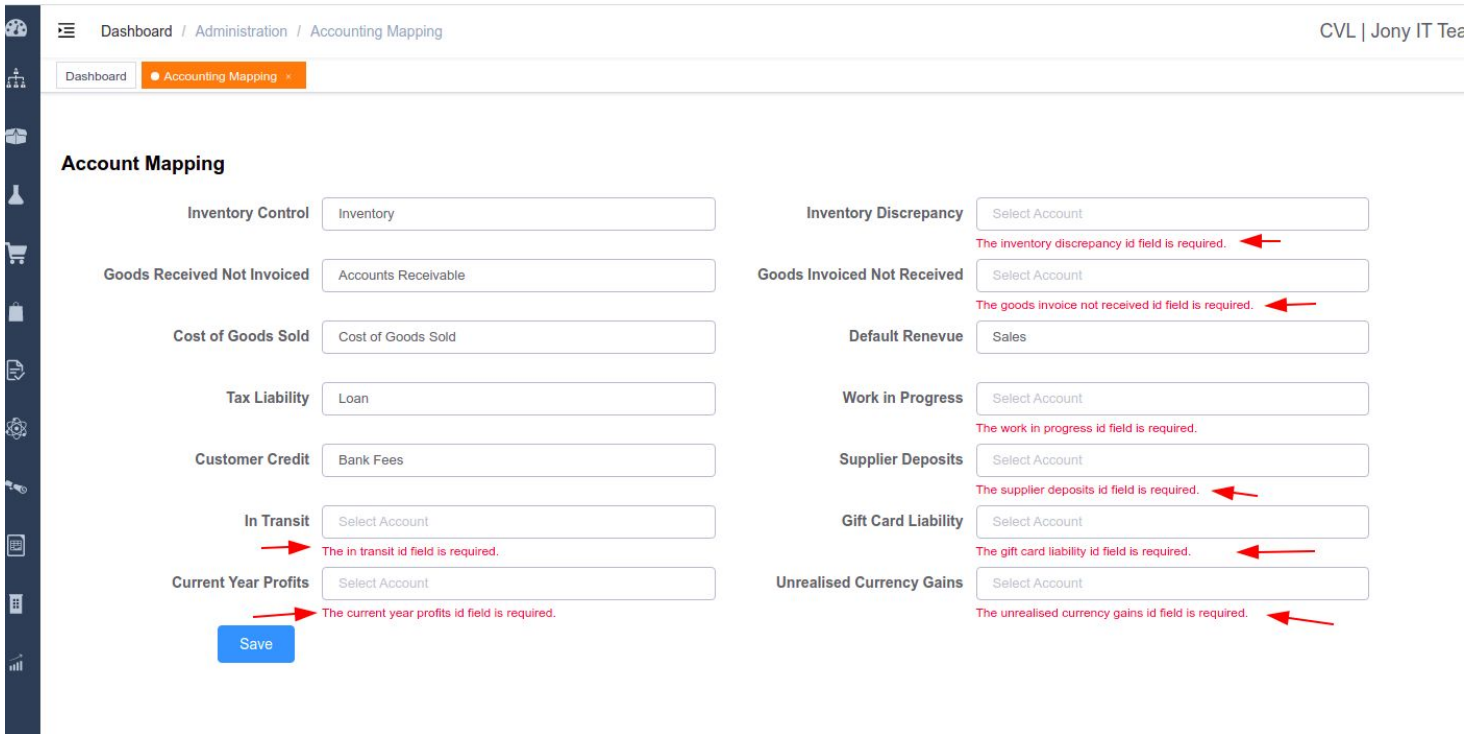

**Show error
messages from
backend below
inputs**

Like this



1) Add rules in backend

```
public function getRules($request, $item = null)
{
    $rules = [
        'inventory_control_id' => ['required', 'exists:tenant.accounts,id'],
        'inventory_discrepancy_id' => ['required', 'exists:tenant.accounts,id'],
        'goods_received_not_invoice_id' => ['required', 'exists:tenant.accounts,id'],
        'goods_invoice_not_received_id' => ['required', 'exists:tenant.accounts,id'],
        'cost_of_goods_sold_id' => ['required', 'exists:tenant.accounts,id'],
        'revenue_id' => ['required', 'exists:tenant.accounts,id'],
        'tax_liability_id' => ['required', 'exists:tenant.accounts,id'],
        'work_in_progress_id' => ['required', 'exists:tenant.accounts,id'],
        'customer_credit_id' => ['required', 'exists:tenant.accounts,id'],
        'supplier_deposits_id' => ['required', 'exists:tenant.accounts,id'],
        'in_transit_id' => ['required', 'exists:tenant.accounts,id'],
        'gift_card_liability_id' => ['required', 'exists:tenant.accounts,id'],
        'current_year_profits_id' => ['required', 'exists:tenant.accounts,id'],
        'unrealised_currency_gains_id' => ['required', 'exists:tenant.accounts,id'],
    ];

    return $rules;
}
```

2) Add error property in inputs on the frontend.

```
center
:closeable="false"
/>
<el-form v-if="accountMappingForm" v-show="isXeroConfigured()" ref="accountMappingForm" :model="accountMappingForm" label-width="150px">
  <el-input v-model="accountMappingForm.id" type="hidden" />
  <el-row>
    <h3>Account Mapping</h3>
  </el-row>
  <el-row>
    <el-col :span="10">
      <el-form-item prop="inventory_control_id" label="Inventory Control" label-width="250px" :error="custom_errors.inventory_control_id">
        <customList
          field-key="inventory_control_id"
          :field-value.sync="accountMappingForm.inventory_control_id"
          :field-label="accountMappingForm.inventory_control_account"
          :config="{ optionLabel: 'label', optionValue: 'id', placeholder: 'Select Account' }"
          :query-function="findAccounts"
          :filter="{category: 'ASSET', status: 'ACTIVE'}"
          @value:updated="handleModel"
        />
      </el-form-item>
    </el-col>
    <el-col :span="10">
      <el-form-item prop="inventory_discrepancy_id" label="Inventory Discrepancy" label-width="250px" :error="custom_errors.inventory_discrepancy_id">
        <customList
          field-key="inventory_discrepancy_id"
          :field-value.sync="accountMappingForm.inventory_discrepancy_id"
          :field-label.sync="accountMappingForm.inventory_discrepancy_account"
        />
      </el-form-item>
    </el-col>
  </el-row>
</el-form>
```

3) Create custom_errors object.

```
30
31 export default {
32   name: 'AccountMappings',
33   components: { customList },
34   data() {
35     return {
36       loading: false,
37       isEditing: false,
38       custom_errors: {},
39       findAccounts,
40       accountMappingForm: {
41         id: null,
42         inventory_control_id:
43         inventory_control acc
```

4) Treat the response for update custom_errors object.

```
create() {  
  let data = this.accountMappingForm  
  this.loading = true  
  this.custom_errors = {}  
  if (this.isEditing) { // UPDATE  
    update(data.id, data).then(response => {  
      this.loading = false  
      if (response.status) {  
        data = response.data  
        this.$notify({  
          title: 'Item updated',  
          message: 'Success!',  
          type: 'success',  
          duration: 5000  
        })  
      } else {  
        if (response.data) {  
          const result = parseApiError(response.message, response.data)  
          for (var x in result) {  
            Vue.set(this.custom_errors, result[x].field, result[x].msg)  
          }  
        } else {  
          Message({  
            message: 'Error: ' + response.message,  
            type: 'error',  
            duration: 10000  
          })  
        }  
      }  
    })  
  } else { // CREATE NEW  
    store(data).then(response => {  
      this.loading = false  
      if (response.status) {  
        data = response.data  
        this.$notify({  
          title: 'Item created',  
          message: 'Success!',  
          type: 'success',  
          duration: 5000  
        })  
      } else {  
        Message({  
          message: 'Error: ' + response.message,  
          type: 'error',  
          duration: 10000  
        })  
      }  
    })  
  }  
}
```

That's it, thanks for your time.