

formsets.py

```
from django.core import exceptions

from oscar.apps.dashboard.catalogue.formsets import \
    ProductCategoryFormSet as CoreProductCategoryFormSet, \
    ProductImageFormSet as CoreProductImageFormSet, \
    ProductRecommendationFormSet as CoreProductRecommendationFormSet, \
    StockRecordFormSet as CoreStockRecordFormSet
from oscar.apps.partner.models import StockRecord

class StockRecordFormSet(CoreStockRecordFormSet):

    def __init__(self, product_class, metal, user, *args, **kwargs):
        self.user = user
        self.require_user_stockrecord = not user.is_staff
        self.product_class = product_class
        self.metal = metal

        if not user.is_staff and \
            'instance' in kwargs and \
            'queryset' not in kwargs:
            kwargs.update({
                'queryset': StockRecord.objects.filter(product=kwargs['instance'],
                                                         partner__in=user.partners.all())
            })

        super().__init__(*args, **kwargs)
        self.set_initial_data()

    def _construct_form(self, i, **kwargs):
        kwargs['product_class'] = self.product_class
        kwargs['metal'] = self.metal
        kwargs['user'] = self.user
        return super()._construct_form(
            i, **kwargs)

class ProductCategoryFormSet(CoreProductCategoryFormSet):

    def __init__(self, product_class, metal, user, *args, **kwargs):
        # This function just exists to drop the extra arguments
        super().__init__(*args, **kwargs)

class ProductImageFormSet(CoreProductImageFormSet):

    def __init__(self, product_class, metal, user, *args, **kwargs):
        super().__init__(*args, **kwargs)

class ProductRecommendationFormSet(CoreProductRecommendationFormSet):

    def __init__(self, product_class, metal, user, *args, **kwargs):
        super().__init__(*args, **kwargs)
```