

This is our Django models, they will help you to help people to write and edit and fix jinja template tags:

```
class CoreModel(models.Model):
    author = models.ForeignKey(
        User,
        on_delete=models.SET_NULL,
        verbose_name=_('Author'),
        related_name='author_%(class)s',
        null=True,
        blank=True,
    )
    editor = models.ForeignKey(
        User,
        on_delete=models.SET_NULL,
        verbose_name=_('Last editor'),
        related_name='+',
        editable=False,
        null=True,
        blank=True,
    )
    business_unit = models.ForeignKey(
        BusinessUnit,
        verbose_name=VERBOSES.business_unit,
        on_delete=models.SET_NULL,
        null=True,
        blank=True,
    )
    update_time = models.DateTimeField(
        _('Updated'),
        auto_now=True,
        null=True,
    )
    create_time = models.DateTimeField(
        _('Created'),
        auto_now_add=True,
        null=True,
    )

class Meta:
    abstract = True
```

```

class Consignment(CoreModel, CustomFieldsExtension):
    objects = ConsignmentManager.as_manager()
    bu_relation_fields = ('contract__business_unit',)

    number = models.CharField(
        _('Number (#)'),
        max_length=255,
        blank=True,
        default="",
    )
    contract = models.ForeignKey(
        to=MODELS.CONTRACT_BASE,
        on_delete=models.CASCADE,
        null=True,
        blank=True
    )
    invoice = models.ForeignKey(
        'finances.Finance',
        on_delete=models.SET_NULL,
        verbose_name=_('Invoice'),
        null=True,
        blank=True,
    )
    volume = models.DecimalField(
        VERBOSES.volume,
        max_digits=14,
        decimal_places=4,
        default=0,
        blank=True
    )
    date = models.DateField(
        _('Date'),
        blank=True,
        null=True
    )
    additional_info = models.TextField(
        VERBOSES.additional_info,
        blank=True,
        default="",
    )
    manager = models.ForeignKey(
        MODELS.USER,

```

```

        verbose_name=_('Responsible'),
        on_delete=models.SET_NULL,
        null=True,
        blank=True
    )
    farm = models.ForeignKey(
        'clients.Farm',
        on_delete=models.PROTECT,
        verbose_name=VERBOSES.farm,
        null=True,
        blank=True,
    )
    elevator = models.ForeignKey(
        'clients.Elevator',
        on_delete=models.PROTECT,
        verbose_name=VERBOSES.elevator,
        related_name='consignment_elevator',
        null=True,
        blank=True,
    )
    terminal = models.ForeignKey(
        MODELS.TERMINAL,
        on_delete=models.SET_NULL,
        verbose_name=_('Terminal'),
        blank=True,
        null=True,
    )
    warehouse = models.ForeignKey(
        to=MODELS.WAREHOUSE, on_delete=models.PROTECT, null=True,
blank=True)
    commodity = models.ForeignKey(
        MODELS.COMMODITY,
        on_delete=models.SET_NULL,
        verbose_name=_('Commodity'),
        blank=True,
        null=True,
    )
    documents = contenttypes.fields.GenericRelation(
        'documents.Document',
        verbose_name=_('Documents'),
        related_query_name='consignments'
    )
    receiver = models.ForeignKey(

```

```

        MODELS.CLIENT,
        on_delete=models.PROTECT,
        verbose_name=_('Receiver'),
        related_name='receiver_%(class)s',
        null=True,
        blank=True,
    )
    producer = models.ForeignKey(
        MODELS.CLIENT,
        on_delete=models.PROTECT,
        verbose_name=_('Producer'),
        related_name='producer_%(class)s',
        null=True,
        blank=True,
    )
    type = models.CharField(
        _('Type'),
        max_length=255,
        choices=CONSIGNMENT_TYPES,
        blank=True,
    )

```

```

class Client(
    ApprovableModel,
    CoreModel,
    ExtDataModelMixin,
    CustomFieldsExtension,
    PredictionsOrderMixin
):
    objects = ClientQuerySet.as_manager()

    name = models.CharField(
        _('Name'),
        max_length=255,
        unique=True,
    )
    related_counterparties = models.ManyToManyField(
        'Client',
        verbose_name=_('Related counterparties'),
        related_name='related_companies',
        blank=True,
    )

```

```

custom_status = models.ForeignKey(
    'core.CustomStatus',
    on_delete=models.SET_NULL,
    verbose_name=_('Custom status'),
    limit_choices_to={
        'content_type__app_label': 'clients',
        'content_type__model': 'client',
    },
    blank=True,
    null=True,
)
abbreviation = models.CharField(
    _('Abbreviation'),
    max_length=255,
    null=True,
    blank=True,
)
full_name = models.CharField(
    _('Full name'),
    max_length=255,
    null=True,
    blank=True,
)
full_name_eng = models.CharField(
    _('Full name Eng'),
    max_length=255,
    null=True,
    blank=True,
)
company_code = models.CharField(
    _('Code'),
    max_length=255,
    null=True,
    blank=True,
)
eic = models.CharField(
    _('EIC'),
    max_length=255,
    null=True,
    blank=True,
)
itn = models.CharField(
    _('ITN'),

```

```
        max_length=255,
        null=True,
        blank=True,
    )
    tax_info = models.CharField(
        _('Tax info'),
        max_length=255,
        null=True,
        blank=True,
    )
    country = models.ForeignKey(
        to=MODELS.COUNTRY,
        on_delete=models.SET_NULL,
        verbose_name=_('Country'),
        null=True,
        blank=True,
    )
    current_address = models.CharField(
        _('Address'),
        max_length=255,
        null=True,
        blank=True,
    )
    address = models.CharField(
        _('Legal address'),
        max_length=255,
        null=True,
        blank=True,
    )
    address_eng = models.CharField(
        _('Legal address eng'),
        max_length=255,
        null=True,
        blank=True,
    )

    person = models.CharField(
        _('Person'),
        max_length=255,
        null=True,
        blank=True,
    )
    person_signatory = models.CharField(
```

```
    _('Signatory'),
    max_length=255,
    null=True,
    blank=True,
)
person_signatory_eng = models.CharField(
    _('Signatory Eng'),
    max_length=255,
    null=True,
    blank=True,
)
person_signatory_docs = models.CharField(
    _('Signatory Docs'),
    max_length=255,
    null=True,
    blank=True,
)
person_signatory_eng_docs = models.CharField(
    _('Signatory Eng Docs'),
    max_length=255,
    null=True,
    blank=True,
)
phone = models.CharField(
    _('Phone'),
    max_length=255,
    null=True,
    blank=True,
)
phone_additional = models.CharField(
    _('Phone (additional)'),
    max_length=255,
    null=True,
    blank=True,
)
email = models.CharField(
    _('Email'),
    max_length=255,
    null=True,
    blank=True,
)
website = models.CharField(
    _('Website'),
```

```

        max_length=255,
        blank=True,
        null=True,
    )
    blacklisted = models.BooleanField(
        _('Black list'),
        default=False,
    )
    client_data_is_checked = models.BooleanField(
        _('Company data is checked'),
        default=False,
    )
    bank_data_is_checked = models.BooleanField(
        _('Bank data is checked'),
        default=False,
    )
    documents = GenericRelation(
        'documents.Document',
        verbose_name=_('Documents'),
        related_query_name='clients'
    )
    date_first_confirmation = models.DateField(
        _('First confirmation (date)'),
        null=True,
        blank=True,
    )
    term_of_proxy = models.DateField(
        _('Term of proxy'),
        null=True,
        blank=True,
    )
    days_before_deadline_to_notify = models.PositiveIntegerField(
        _('Days before deadline to notify'),
        default=0,
    )
    date_confirmation = models.DateField(
        _('Confirmation (date)'),
        null=True,
        blank=True,
    )
    confirmation_info = models.TextField(
        _('Confirmation info'),
        blank=True,

```



```
        null=True,
    )
    date_request = models.DateField(
        _('Request (date)'),
        null=True,
        blank=True,
    )
    date_received = models.DateField(
        _('Received (date)'),
        null=True,
        blank=True,
    )
    date_of_incorporation = models.DateField(
        _('Incorporation (date)'),
        null=True,
        blank=True,
    )
    status = models.CharField(
        _('Status'),
        max_length=32,
        choices=STATUS_CHOICES,
        null=True,
        blank=True,
        default='New',
    )
    additional_info = models.TextField(
        VERBOSESES.additional_info,
        blank=True,
        null=True,
    )
    president = models.TextField(
        _('President'),
        blank=True,
        null=True,
    )
    ceo = models.TextField(
        _('CEO'),
        blank=True,
        null=True,
    )
    cfo = models.TextField(
        _('CFO'),
        blank=True,
```

```
        null=True,
    )
    other_executive = models.TextField(
        _('Other executive'),
        blank=True,
        null=True,
    )
    parent_company = models.TextField(
        _('Parent company'),
        blank=True,
        null=True,
    )
    ultimate_parent_company = models.TextField(
        _('Ultimate parent company'),
        blank=True,
        null=True,
    )
    ultimate_beneficiary_owners = models.TextField(
        _('Ultimate beneficiary owners'),
        blank=True,
        null=True,
    )
    activities_traded_products = models.TextField(
        _('Activities & traded products'),
        blank=True,
        null=True,
    )
    trade_suppliers = models.TextField(
        _('Suppliers'),
        blank=True,
        null=True,
    )
    customers = models.TextField(
        _('Customers'),
        blank=True,
        null=True,
    )
    can_confirm = models.BooleanField(
        _('Can confirm clients'),
        default=False,
    )
    kpi_confirmation_success = models.BooleanField(
        _('Don`t use kpi check'),
```

```

        default=False,
    )
    responsible = models.ForeignKey(
        settings.AUTH_USER_MODEL,
        verbose_name=_('Responsible'),
        on_delete=models.SET_NULL,
        related_name='client_responsible_set',
        null=True,
        blank=True,
    )
    responsible_for_confirmation = models.ForeignKey(
        settings.AUTH_USER_MODEL,
        verbose_name=_('Responsible for confirmation'),
        on_delete=models.SET_NULL,
        related_name='client_responsible_for_confirmation_set',
        null=True,
        blank=True,
    )
    is_resident = models.BooleanField(
        _('Client is resident'),
        default=True,
    )
    sap_vendor_number = models.CharField(
        _('SAP vendor number'),
        max_length=20,
        unique=True,
        null=True,
        blank=True,
    )
    business_units = models.ManyToManyField(
        'core.BusinessUnit',
        verbose_name=_('Business Units'),
        blank=True,
        related_name='clients_many'
    )
    equity = models.DecimalField(
        _('Equity'),
        max_digits=16,
        decimal_places=6,
        default=0,
    )
    net_income = models.DecimalField(
        _('Net income'),

```

```

        max_digits=16,
        decimal_places=6,
        default=0,
    )
    total_fixed_assets = models.DecimalField(
        _('Total fixed assets'),
        max_digits=16,
        decimal_places=6,
        default=0,
    )
    turnover = models.DecimalField(
        _('Turnover'),
        max_digits=16,
        decimal_places=6,
        default=0,
    )
    total_assets = models.DecimalField(
        _('Total assets'),
        max_digits=16,
        decimal_places=6,
        default=0,
    )
    annual_traded_volume = models.DecimalField(
        _('Annual traded volume'),
        max_digits=16,
        decimal_places=6,
        default=0,
    )
    info_for_traders = models.TextField(
        _('Info for traders'),
        blank=True,
        null=True,
    )
    # from roles
    future_contact = models.DateField(
        null=True,
        blank=True,
        verbose_name=_('Contact (future)'),
    )

    city = models.ForeignKey(
        'location.City',
        on_delete=models.SET_NULL,

```

```
        verbose_name=_('City'),
        null=True,
        blank=True,
    )
    district = models.ForeignKey(
        'location.District',
        on_delete=models.SET_NULL,
        verbose_name=_('District'),
        null=True,
        blank=True,
    )
    regions = models.ManyToManyField(
        'location.Region',
        verbose_name=_('Regions'),
        blank=True,
    )
    crop_area = models.DecimalField(
        _('Client crop area'),
        max_digits=16,
        decimal_places=6,
        default=0,
    )
    assets = models.CharField(
        VERBOSES.additional_info,
        max_length=256,
        blank=True,
        null=True,
    )
    pledge = models.CharField(
        VERBOSES.additional_info,
        max_length=256,
        blank=True,
        null=True,
    )
    cash_flow = models.CharField(
        _('Cash Flow'),
        max_length=256,
        blank=True,
        null=True,
    )
    transports_count = models.IntegerField(
        _('Client transports count'),
        default=0,
```

```

)
regular_status = models.CharField(
    _('Regular status'),
    max_length=32,
    choices=(
        ('potencial', _('Potencial')),
        ('active', _('Active')),
        ('inactive', _('Inactive')),
        ('closed', _('Closed')),
    ),
    blank=True,
    default='potencial',
)
company_group = models.ForeignKey(
    CompanyGroup,
    on_delete=models.SET_NULL,
    verbose_name=VERBOSES.company_group,
    blank=True,
    null=True,
)
size = models.CharField(
    max_length=32,
    choices=SIZE_CHOICES,
    verbose_name=_('Size'),
    null=True,
    blank=True,
    default='Medium'
)
first_contact = models.DateField(
    _('Contact Date (first)'),
    blank=True,
    null=True,
)
relations = models.CharField(
    max_length=32,
    choices=RELATION_CHOICES,
    verbose_name=_('Relation'),
    null=True,
    blank=True,
    default='Neutral',
)
is_distributor = models.BooleanField(
    default=False

```

```

)
is_agent = models.BooleanField(
    _('Agent'),
    default=False,
)
is_producer = models.BooleanField(
    _('Producer'),
    default=False,
)
is_competitor = models.BooleanField(
    _('Competitor'),
    default=False,
)
use_local_currency = models.BooleanField(
    _('Local currency'),
    default=False,
)
use_foreign_currency = models.BooleanField(
    _('Foreign currency'),
    default=False,
)
clientroles = models.ManyToManyField(
    MODELS.CUSTOM_CLIENT_ROLE,
    through='clients.ClientCustomClientRole',
    blank=True
)
is_own_counterparty = models.BooleanField(
    _('Own cunterparty'),
    default=False
)
outgoing_invoice_number_prefix = models.CharField(
    null=True,
    max_length=32,
    verbose_name=_('outgoing invoice prefix'),
    blank=True
)
client_template = models.ForeignKey(
    'ClientTemplate',
    on_delete=models.SET_NULL,
    verbose_name=_('Template'),
    blank=True,
    null=True,
)

```

```

signatory_position = models.CharField(
    _('Signatory position'),
    max_length=255,
    blank=True,
    null=True,
)
signatory_position_eng = models.CharField(
    _('Signatory position eng'),
    max_length=255,
    blank=True,
    null=True,
)
economic_activities_code = models.CharField(
    _('Code of economic activities'),
    max_length=512,
    blank=True,
    null=True,
)
counterparty_type = models.CharField(
    _('Counterparty type'),
    max_length=255,
    choices=[
        ('legal_entity', _('legal entity')),
        ('physical_person', _('physical person')),
        ('individual_entrepreneur', _('individual entrepreneur'))
    ],
    blank=True,
    null=True
)
mailing_address = models.CharField(
    _('Mailing address'),
    max_length=255,
    null=True,
    blank=True
)
type_of_activities = models.ManyToManyField(
    'core.TypeOfActivity',
    verbose_name=_('Types of activity'),
    related_name='clients',
    blank=True
)
updates = GenericRelation(
    'ClientUpdate',

```



```
verbose_name=_('Updates'),  
content_type_field='updates_entity_content_type',  
object_id_field='updates_entity_id',  
related_query_name='updated_entity'  
)
```