

为了方便前端同学开发时更好的与后端沟通，现在把这次开发主要用到的实体后端的设计展现出来。直接阅读即可掌握所有应该提供的信息。

用户

```
USER_TYPE_CHOICES = (
    (0, 'Individual'),
    (1, 'College'),
    (2, 'Company'),
)

#用户状态
USER_STATE_CHOICES = (
    (0, '普通用户'),
    (1, '专家认证中'),
    (2, '企业认证中'),
    (3, '封禁中'),
    (4, '认证专家'),
    (5, '认证企业')
)

class User(AbstractUser):
    nick_name = models.CharField(max_length=20)
    email = models.CharField(max_length=30)
    institution = models.CharField(max_length=20, blank=True, null=True)
    icon = models.ImageField(upload_to= "images/%Y%m/%d/icons",
                             default= 'images/default_user_icon.jpg')
    biography = models.CharField(max_length=50, null=True)
    user_type = models.IntegerField(choices=USER_TYPE_CHOICES, default= 0)
    followers = models.ManyToManyField('User')
    favorites = models.ManyToManyField('PapModel', related_name='favorites')
    is_confirmed = models.BooleanField(default=False)
    #用户目前状态
    state = models.IntegerField(choices=USER_STATE_CHOICES, default=0)
    enterprise_info = models.OneToOneField("Enterprise_info",
on_delete=models.CASCADE, related_name="enterprise_info", null=True, blank=True)
```

企业信息

```
class Enterprise_info(models.Model):
    #企业名
    name = models.CharField(max_length=20, blank=True, null=True)
    #地址
    address = models.CharField(max_length=20, blank=True, null=True)
    #官网
    website = models.CharField(max_length=20, blank=True, null=True)
    #简介
    instruction = models.CharField(max_length=50, blank=True, null=True)
    #电话
    phone = models.CharField(max_length=20, blank=True, null=True)
    #法定代表人
```

```

legal_representative = models.CharField(max_length=10)
#注册资本
register_capital = models.IntegerField()
#主营业务
field = models.CharField(max_length=10)
#营业执照
business_license = models.ImageField(upload_to="images/%Y%m/%d/icons",
                                      default='images/default_user_icon.jpg')
#法人身份证
legal_person_ID = models.ImageField(upload_to="images/%Y%m/%d/icons",
                                    default='images/default_user_icon.jpg')

```

需求（貌似没有找到固定的后端关键词，所以关键词的具体处理可能需要这次需求平台小组组内沟通决定）

```

NEED_FIELD_CHOICES = (
    (0, '信息技术'),
    (1, '装备制造'),
    (2, '新材料'),
    (3, '新能源'),
    (4, '节能环保'),
    (5, '生物医药'),
    (6, '科学创意'),
    (7, '检验检测'),
    (8, '其他')
)

NEED_STATE = (
    (0, '进行中'),
    (1, '已结束')
)

EMERGENCY_CODE = (
    (0, '低'),
    (1, '中'),
    (2, '高')
)

class Need(models.Model):
    #需求标题
    title = models.CharField(max_length=20)
    #需求描述
    description = models.CharField(max_length=50)
    #预计经费（报酬）
    money = models.IntegerField()
    #需求创建时间，自动填充
    start_time = models.DateTimeField(auto_now_add=True)
    #有效时间（预计结束时间）
    valid_time = models.DateTimeField(blank=True, null=True)
    #关键字
    key_word = models.CharField(max_length=10)
    #技术领域
    field = models.IntegerField(choices=NEED_FIELD_CHOICES, default=8)
    #地址
    address = models.CharField(max_length=20)

```

```
#需求公司
enterprise = models.ForeignKey(User, on_delete=models.CASCADE,
related_name="enterprise")
#订单状态
state = models.IntegerField(choices=NEED_STATE, default=0)
#紧急程度
emergency = models.IntegerField(choices=EMERGENCY_CODE, default=0)
#预计招募人数
predict = models.IntegerField()
#实际已招募人数
real = models.IntegerField(default=0)
```

订单

```
ORDER_STATE = (
    (0, "正在合作中"),
    (1, "合作结束")
)

class Order(models.Model):
    #订单对应需求
    need = models.ForeignKey(Need, on_delete=models.CASCADE,
related_name="need")
    #订单接收方专家
    user = models.ForeignKey(User, on_delete=models.CASCADE,
related_name="expert")
    #订单创建时间，自动填充
    create_time = models.DateTimeField(auto_now_add=True)
    #订单结束时间
    end_time = models.DateTimeField(blank=True, null=True)
    #订单目前状态
    state = models.IntegerField(choices=ORDER_STATE, default=0)
```