views.py

ここではsystem/views.pyについて説明する.
views.pyは画面表示する為のデータを管理するファイルである.

テンプレートビュー

参考ページ

- テンプレートビューのパターン : https://qiita.com/ytyng/items/7cb3c3a5605974151678
- 汎用ビューの使い方 : https://qiita.com/felyce/items/7d0187485cad4418c073

関数

- form_valid
  フォームに値を入れたり，内容チェックなどの処理.
- get_context_date
  models.pyからモデルを持ってきて，そのデータを表示するための関数
- get_query_set models.pyから持ってきたデータの順番を操作する.

TOPページ

try以下はtopページのカレンダーに予約時間を表示する為に，予約時間を整形する為の処理である.

ログイン系

ログイン処理.
クラスの引数にLoginRequireMixinが含まれるクラスはログイン中でないと見れないページである.

予約系

よくあるCREATEVIEW．参考ページ参照.

form_valid

form_validはランダム文字列を生成して，それを予約モデルに格納する.

```
def form_valid(self, form):
  # 入力欄にないフィールドを追加
  form.instance.owner_id = self.request.user
  # 予約フォームを読み込む
  random_string = ''.join(random.choices(string.ascii_letters +
string.digits, k=200))
  # ランダム文字列生成
  form.instance.rdm_str = random_string
  # 予約フォームのrdm_strに生成したランダム文字列を格納
  qr_code = qrcode.make( random_string )
  # qr_code作成
  qr_code.save("qr.png")
```

```
  # qrコードをを画像ファイルで出力
  ...
```

メールの処理はこれ : https://narito.ninja/blog/detail/64/を参考.

フォームの初期値の設定

```
def get_initial(self):
  initial = super().get_initial()
  user = User.objects.get(id=self.request.user.pk)
  # ログインユーザの情報取得
  if self.request.GET.get("facility"):
  # requestにfacilityが存在するとき = 施設ページからきた時
    facility_from_form = self.request.GET.get("facility")
    # facility_from_formに施設idを格納
    facility = Facility.objects.get(facility=facility_from_form)
    # 施設オブジェクトをfacilityに格納
    initial["facility"] = facility
    # facilityの初期値にfacilityオブジェクトの施設名を格納（def __str__より）
  initial["last_name"] = user.last_name
  # last_nameの初期値にログインユーザーの苗字を格納
  ...
```

施設検索

エラーのでる魔境.
参考 : https://takaxtech.com/2018/09/23/article260/

ランダム文字列

参考 : http://note.crohaco.net/2018/django-rest-framework-view/

```
# ランダム文字列受け取り : http:~~/random_string?rdm_str="~~~"
class Random_string(APIView):
    def get(self, request, format=None):
        if "rdm_str" in request.GET:
        # http:~~/random_string?rdm_str<-が存在するとき
            # query_paramが指定されている場合の処理
            random_string = str(request.GET.get("rdm_str"))
            # http:~~/random_string?rdm_str="~~~" <- "~~"をrandom_stringに格
納
            try :
                Reservation.objects.filter(rdm_str=random_string)[0]
            except IndexError:
            # IndexErrorが出た時
                return Response({"Ans":
"False_none"},status=status.HTTP_200_OK)
            if random_string ==
```

```
Reservation.objects.filter(rdm_str=random_string)[0].rdm_str :
            # GETで受け取ったランダム文字列が含まれるReservationオブジェクトがあると
き
                reservation =
Reservation.objects.filter(rdm_str=random_string)[0]
                # ここからタイムゾーン処理
                tz = pytz.timezone('Asia/Tokyo')
                now_nozone = datetime.now()
                now = tz.localize(now_nozone)
                # ここまで
                if reservation.date_select - timedelta(minutes=15) <= now
and \
                reservation.date_select +
timedelta(hours=reservation.time_for) + timedelta(minutes=15) >= now :
                    # 今が 予約時間-15分以上　かつ 予約時間+利用時間+15分以下 のと
き
                    return Response({"Ans":
"True"},status=status.HTTP_200_OK)
                else :
                    return Response({"Ans":
"False_time"},status=status.HTTP_200_OK)
            else:
                return Response({"Ans":
"False_none"},status=status.HTTP_200_OK)
        else:
            # query_paramが指定されていない場合の処理
            return Response({"Ans": "False"},status=status.HTTP_200_OK)
```

前へ 目次 次へ