

HR System code documentation

Q Misell

September 30, 2018

Contents

1	Base information on the system	1
2	Adding a new app	1
3	Adding a model	3

1 Base information on the system

The webserver is Django 2.1. Documentation for Django is available [here](#).

The HTML templates support the Django templating language, whose documentation is available [here](#). Bootstrap is also included in the template base, whose documentation is [here](#).

The **hr_system** run configuration in PyCharm (shown in the top-right of the IDE) will start a webserver running locally using a local database on `http://localhost:8001`. This configuration also runs a database migration before starting.

A migration can also be applied without restarting the webserver using the **Migrate** run configuration.

The **Make migrations** run configuration will instruct Django to create the database migrations based off of the models defined in python. This should be run after making any changes to the models (usually in the *models.py* file of every app).

2 Adding a new app

Open a shell (Command prompt, terminal, etc). Make sure you are in the project venv.

```
./venv/bin/activate
```

Create a new Django app. The app name must be all one word. In the following steps holiday will be used however this should be replaced with the chosen app name where appropriate.

```
python3 manage.py startapp <app name here eg. holiday>
```

Add the app to the Django settings file.

File: *hr_system/settings.py*

```
INSTALLED_APPS = [  
    ...  
    'holiday.apps.HolidayConfig',  
]
```

Add a human readable name to the app, and tell the HR system that this app should be in the menu.

File: *holiday/apps.py*

```
from django.apps import AppConfig
```

```
class HolidayConfig(AppConfig):  
    name = 'holiday'  
    # Add these two  
    verbose_name = 'Holiday'  
    in_menu = True
```

Create the index template file.

File: *holiday/templates/holiday/index.html*

```
{% extends "base.html" %}  
  
{% block content %}  
    <!-- Page content goes here -->  
    <div class="container">  
        <h1>Holiday</h1>  
    </div>  
{% endblock %}
```

Create the view for the index page.

File: *holiday/views.py*

```
from django.shortcuts import render
```

```
def index(request):  
    return render(request, "holiday/index.html")
```

Create app's url definitions file and add the index page to it.

File: *holiday/urls.py*

```
from django.urls import path  
from . import views  
  
app_name = 'holiday'  
  
urlpatterns = [  
    path('', views.index, name='index'),  
]
```

3 Adding a model

A model is used to store data in the database. Django handles the interfacing to the database, so you don't have to. Let's create an model to store holiday.

In the models file create an empty model class.

File: *holiday/models.py*

```
from django.db import models

class HolidayLeave(models.Model):
    pass
```

Add some fields, these could be various things like a textbox, an email, a number, etc. For a complete list see the Django documentation [here](#).

```
class HolidayLeave(models.Model):
    start = models.DateField()
    end = models.DateField()
```

You may wish to link your model to a different model, this can be accomplished with a *ForeignKey*. If this model is in a different app you can simply import it. Don't forget to tell Django what to do when the referenced model is deleted from the database (see [here](#) for a description of possible options), and to give a way to access this model from the referenced model (see [here](#) for documentation on that).

```
import employees.models

class HolidayLeave(models.Model):
    ...
    employee =
        models.ForeignKey(employees.models.Employee,
                           on_delete=models.CASCADE,
                           related_name="holiday_leave")
```

After defining your model don't forget to run the **Make migrations** and **Migrate** run configurations in PyCharm to update the database.