Описание решения

Содержание

Эбщее описание	1
База данных	
Серверная часть	
Клиентская часть	
Гестирование	
composition	

Общее описание

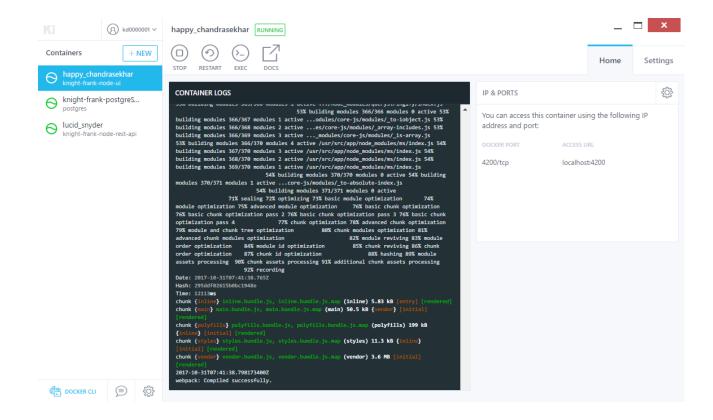
Приложение, опубликованное в облаке (digitalocean.com), можно увидеть по ссылке: http://138.68.185.190

Решение размещено по трём Docker-контейнерам Linux, содержащим уровни приложения.

Для развёртывания приложения на локальной машине достаточно запустить _docker-compose.cmd (при условии, что на локальной машине установлен Docker). По команде будут запущены контейнеры:

- \Database\Scripts\Dockerfile сервер базы данных (PostgreSQL)
- \Service\Rest\Dockerfile веб-сервис (Node.js)
- \UI\ Dockerfile
 интерфейс пользователя (Angular 4, на веб-сервере Ngnix)

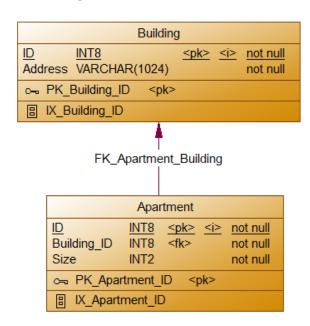
Используя Kitematic, можно наблюдать на локальной машине три запущенных контейнера:



База данных

В прилагаемом архиве папка \Database

Согласно требованию, используется PostgreSQL, привожу диаграмму физической модели (Sybase PowerDesigner):



В прилагаемом архиве (папка "Database") имеется диаграмма и скрипты pdplSQL создания структуры таблиц, а также генерации тестовых данных, со случайным их распределением.

Серверная часть

В прилагаемом архиве папка \Service\Rest

Согласно требованию, используется Node.js в качестве платформы для REST-сервиса. Используются библиотеки Express.js и основанная на ней Loopback.io, добавляющая возможности ORM-системы для реляционных данных, а также возможность генерации VIEW-модели.

Loopback позволяет как читать данные из реляционных таблиц, так и выполнять хранимые функции и SQLзапросы с параметрами, при необходимости выборок на стороне сервера.

Для данного решения это оказалось необходимым, т.к. для выборок понадобилась агрегация и соединение таблиц, что оптимально делать в реляционной СУБД, а не на стороне сервиса.

Структура серверной части проекта:

```
server.js ×
                                                              building.js ×
                                                                             building.json ×
                                                                                              datasources.json ×
                                                                                                                  apartment.json
--- Rest [WebApi] D:\Work\KnightFrank\Service\Rest
                                                        'use strict':
  in node_modules library root
  Ė- ■ server
                                                       var loopback = require('loopback');
                                                3
     ⊨... boot
                                                4
                                                       var boot = require('loopback-boot');
        authentication.js
                                                5
                                                6
                                                       var app = module.exports = loopback();
     .... ■ models
                                                7
         apartment.js
                                                8
                                                       app.start = function() {
           apartment.json
          building.js
building.json
                                                9
                                                         // start the web server
                                                         return app.listen(function() {
                                                            app.emit('started');
        component-config.json
        config.json
                                                            var baseUrl = app.get('url').replace(/\/$/, '');
        datasources.json
                                                            console.log('Web server listening at: %s', baseUrl);
        middleware.development.json
                                               14
                                                            // loopback routing:
        middleware.json
                                               15
                                                            if (app.get('loopback-component-explorer')) {
        model-config.json
                                                             var explorerPath = app.get('loopback-component-explorer').mountPath;
                                               16
         server.js
                                               17
                                                              console.log('Browse your REST API at %s%s', baseUrl, explorerPath);
     .editorconfig
     .eslintianore
                                               19
                                                            // custom routing:
     eslintrc
                                                           app.post('/api/requests', function(req, res) {
     gitignore.
                                                             res.send(200, 'Просмотр успешно заказан');
        .vo-rc.json
     Dockerfile
                                                            1);
                                               23
                                                         });
     Dockerignore
                                               24
                                                      (A):
     DockerStart.cmd
                                               25
     package.json
                                                      // Bootstrap the application, configure models, datasources and middleware.
                                               26
       package-lock.json
                                               27
                                                      // Sub-apps like REST API are mounted via boot scripts.
  |||| External Libraries
                                               28
                                                      boot(app, __dirname, function(err) {
                                                         if (err) throw err;
                                                          // start the server if `$ node server.js`
                                               31
                                                         if (require.main === module)
                                               32
                                               33
                                                            app.start();
                                               34
                                                      <u>`</u>11);
                                              35
```

Клиентская часть

В прилагаемом архиве папка \UI

Структура клиентской части проекта:

```
⊕ ÷ | ÷ 
Project
                                                    app.module.ts ×
🗐 ·· 🖿 🔰 D:\Work\KnightFrank\UI
                                                           import { BrowserModule } from '@angular/platform-browser';
  🖶 - 🖿 dist
                                                           import { NgModule } from '@angular/core';
  ⊕-- 📄 e2e
                                                           import { FormsModule, ReactiveFormsModule } from '@angular/forms';
  ibrary root
                                                           import { HttpModule, JsonpModule } from '@angular/http';
  i- src
                                                           import { BrowserAnimationsModule } from '@angular/platform-browser/animations
     6
        import { AppRoutes } from './app.routing';
           - apartments
                                                    8
                apartment-list.component.html
                                                    9
                                                           import { AppComponent } from './app.component';
           --- buildings
                                                           import { BuildingListComponent } from './components/buildings/building-list co
               building-list.component.html
                                                           import { ApartmentListComponent } from './components/apartments/apartment-list
           i layout
                                                   13
                                                           import { AppMenuComponent } from './components/layout/app.menu.component';
               ··· 🚚 app.content.component.ts
                                                   14
                                                           import { AppContentComponent } from './components/layout/app.content.compon
                 app.menu.component.ts
                                                   15
             abase.component.ts
                                                   16
                                                           import { NgbModule } from '@ng-bootstrap/ng-bootstrap';
        - models
              apartment.model.ts
                                                           import { BuildingService } from './services/buildings/building.service';
                                                   18
              base.model.ts
                                                   19
                                                           import { ApartmentService } from './services/apartment/apartment.service';
              🚚 building.model.ts
                                                           import { OrderService } from './services/order/order.service';
             . and order.model.ts
        ... services
                                                           @NgModule({
           - apartment
                                                             declarations: [
                apartment.service.ts
                                                   24
                                                               AppComponent,
           - buildings
               ..... # building.service.ts
                                                   25
                                                               AppMenuComponent,
                                                               AppContentComponent,
           - order
                                                               BuildingListComponent,
               ···· 🚚 order.service.ts
              abase.service.ts
                                                               ApartmentListComponent
           app.component.css
                                                             ],
             app.component.html
                                                             imports: [
             app.component.spec.ts
                                                               BrowserModule.
             app.component.ts
                                                               BrowserModule,
             app.module.ts
                                                               BrowserAnimationsModule.
           app.routing.ts
                                                               FormsModule,
                                                               ReactiveFormsModule,
                           +- - 2 ÷ | 4- 1-
                                                               HttpModule,
□ II \package.json
                                                               JsonpModule,
      o ng
                                                               NgbModule.forRoot(),
     start
                                                               AppRoutes

    open

                                                   40
                                                             1.

    build

                                                             providers: [BuildingService, ApartmentService, OrderService],
                                                   41
      prod
                                                   42
                                                             bootstrap: [AppComponent]

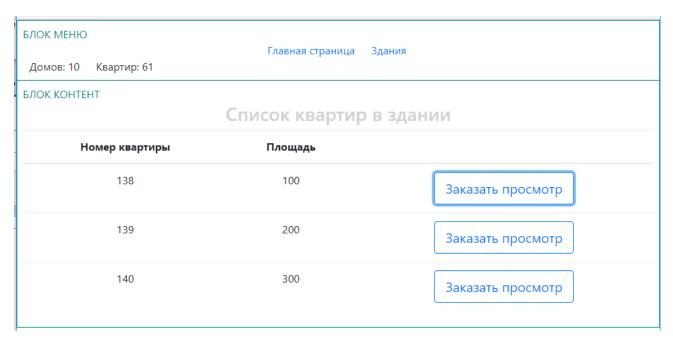
    test

                                                   43
                                                           ЭĐ
      lint
                                                   44
                                                           export class AppModule { }
      。 e2e
                                                   45
```

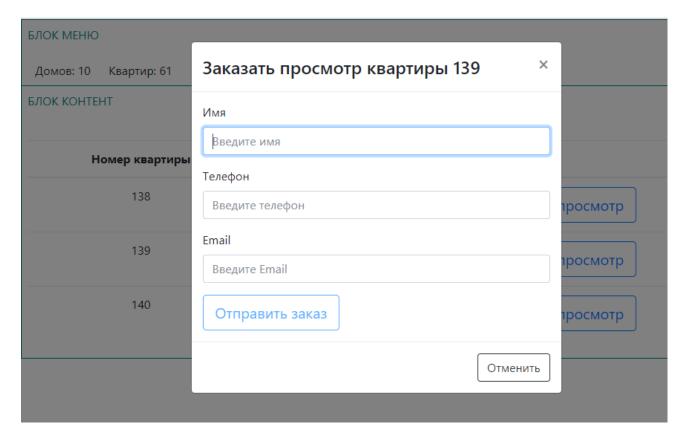
Ниже скриншот формы "Список зданий, в html-разметке специально показаны границы областей — "БЛОК МЕНЮ" и "БЛОК КОНТЕНТ" (в контенте располагается router-outlet, в который подгружаются все прочие компоненты, вызываемые из меню):

БЛОК МЕНЮ Главная страница Здания Домов: 10 Квартир: 61			
Список зданий			
Идентификатор дома	Адрес дома	Количество квартир в доме	
64	Address 64	8	
65	Address 65	3	
66	Address 66	3	
67	Address 67	6	
68	Address 68	7	
69	Address 69	3	
70	Address 70	7	
71	Address 71	8	
72	Address 72	6	
73	Address 73	10	

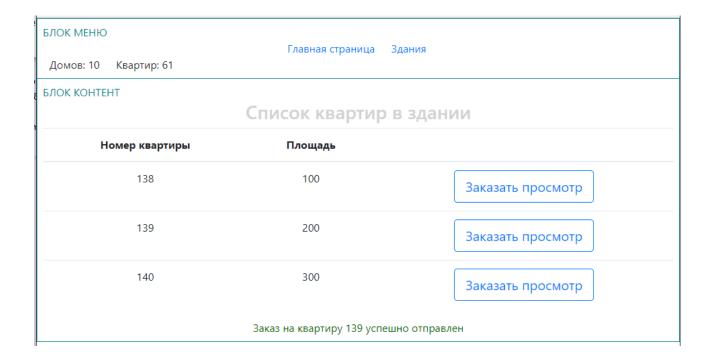
Ниже скриншот формы со списком квартир, вызываемой по клику на идентификаторе дома:



По клику на кнопке "Заказать просмотр" открывается модальное окно ng-bootstrap с формой, позволяющей оправить привязанную к форме модель Order методом Post на Rest-сервис.



В случае успешной отправки, (после отработки Promise), на форме будет показано сообщение об этом.



Тестирование