package com.avp.cdai.web;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.ApplicationContext;

@SpringBootApplication

public class WebApplication {

private static final Logger logger = LoggerFactory.getLogger(WebApplication.class);

public static void main(String[] args) throws Exception {

ApplicationContext ctx = SpringApplication.run(WebApplication.class, args);

}

}

package com.avp.cdai.web.entity;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.Id;

import javax.persistence.Table;

import java.util.Date;

/\*\*

\* Created by guo on 2017/8/10.

\*/

@Entity

@Table(name = "line\_cumulative\_passenger\_flow",schema = "afccd")

public class LineCumulativeFlow {

@Id

private Integer id;

@Column(name = "line\_id")

private Integer lineId;

@Column(name = "direction")

private Integer direction;

@Column(name = "passenger\_flow")

private Integer passengerFlow;

@Column(name = "section")

private Integer section;

@Column(name = "flow\_timestamp")

private Date flowTime;

@Column(name = "insert\_time")

private Date insertTime;

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

public Date getFlowTime() {

return flowTime;

}

public void setFlowTime(Date flowTime) {

this.flowTime = flowTime;

}

public Date getInsertTime() {

return insertTime;

}

public void setInsertTime(Date insertTime) {

this.insertTime = insertTime;

}

public Integer getLineId() {

return lineId;

}

public void setLineId(Integer lineId) {

this.lineId = lineId;

}

public Integer getDirection() {

return direction;

}

public void setDirection(Integer direction) {

this.direction = direction;

}

public Integer getPassengerFlow() {

return passengerFlow;

}

public void setPassengerFlow(Integer passengerFlow) {

this.passengerFlow = passengerFlow;

}

public Integer getSection() {

return section;

}

public void setSection(Integer section) {

this.section = section;

}

}

package com.avp.cdai.web.controller;

import com.avp.cdai.web.entity.LineTimeShareFlow;

import com.avp.cdai.web.entity.LineTimeShareFlow\_;

import com.avp.cdai.web.repository.LineTimeShareFlowRepository;

import com.avp.cdai.web.rest.ResponseBuilder;

import com.avp.cdai.web.rest.ResponseCode;

import com.avp.cdai.web.rest.RestBody;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.data.jpa.domain.Specification;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.RequestParam;

import org.springframework.web.bind.annotation.RestController;

import javax.persistence.criteria.CriteriaBuilder;

import javax.persistence.criteria.CriteriaQuery;

import javax.persistence.criteria.Predicate;

import javax.persistence.criteria.Root;

import java.text.SimpleDateFormat;

import java.util.\*;

import static org.springframework.data.jpa.domain.Specifications.where;

/\*\*

\* Created by guo on 2017/8/8.

\*/

@RestController

public class LineTimeShareFlowController {

@Autowired

LineTimeShareFlowRepository lineTimeShareFlowRepository;

private final Logger logger = LoggerFactory.getLogger(this.getClass());

@RequestMapping(value = "lineShareFlow",method = RequestMethod.GET)

ResponseEntity<RestBody<LineTimeShareFlow>> findAll(){

ResponseBuilder builder = ResponseBuilder.createBuilder();

try {

List<LineTimeShareFlow> lineShareFlows = lineTimeShareFlowRepository.findAll();

if(lineShareFlows != null) {

builder.setResultEntity(lineShareFlows, ResponseCode.RETRIEVE\_SUCCEED);

logger.debug("线路分时客流数量为：{}",lineShareFlows.size());

return builder.getResponseEntity();

}

}catch (Exception e){

logger.error(e.getMessage());

builder.setErrorCode(ResponseCode.RETRIEVE\_FAILED);

}

return builder.getResponseEntity();

}

@RequestMapping(value = "lineShareByConditions",method = RequestMethod.GET)

ResponseEntity<RestBody<LineTimeShareFlow>> findByConditions(@RequestParam("ids") List<Integer> ids,@RequestParam(value = "direct",required = false) Integer direct,Integer section, Date time){

ResponseBuilder builder = ResponseBuilder.createBuilder();

try {

List<LineTimeShareFlow> list = lineTimeShareFlowRepository.findAll(where(byConditions(ids, direct,section, time)));

if (list.size() <= 0) {

logger.debug("线路客流分时多条件查询失败，结果为空");

} else {

logger.debug("线路客流分时多条件查询成功，数据量为:({})", list.size());

}

builder.setResultEntity(list, ResponseCode.RETRIEVE\_SUCCEED);

} catch (Exception e) {

logger.error(e.getMessage());

builder.setErrorCode(ResponseCode.RETRIEVE\_FAILED);

}

return builder.getResponseEntity();

}

// Dynamic Query Utils

public Specification<LineTimeShareFlow> byConditions(List<Integer> ids, Integer direct,Integer section, Date time) {

return new Specification<LineTimeShareFlow>() {

public Predicate toPredicate(Root<LineTimeShareFlow> root, CriteriaQuery<?> query, CriteriaBuilder builder) {

Predicate predicate = builder.conjunction();

logger.debug("lineShareByConditions请求的参数ids值为:{}", ids);

if (ids != null) {

// predicate.getExpressions().add(builder.equal(root.get(LineTimeShareFlow\_.id), ids));

// predicate.getExpressions().add(builder.in(root.get(LineTimeShareFlow\_.lineId)).in(ids));

predicate.getExpressions().add(root.<Integer>get(LineTimeShareFlow\_.lineId).in(ids));

}

//

logger.debug("lineShareByConditions请求的参数direct值为:{}", direct);

if (direct != null) {

predicate.getExpressions().add(builder.equal(root.get(LineTimeShareFlow\_.direction), direct));

}

logger.debug("lineShareByConditions请求的参数section值为:{}", section);

if (section != null) {

predicate.getExpressions().add(builder.equal(root.get(LineTimeShareFlow\_.section), section));

}

//

logger.debug("lineShareByConditions请求的参数time值为:{}", time);

if (time != null) {

/\*

//测试改变传入参数的值 - (调试中)

Date start = new Date();

Date end = new Date();

logger.debug("转换前：{},{}",start,end);

GetDayStartEndDate.CreateStartEndDate(start,end,time);

logger.debug("转换后：{},{}",start,end);

\*/

SimpleDateFormat format1 = new SimpleDateFormat("yyyy-MM-dd 00:00:00");

String strStart = format1.format(time);

SimpleDateFormat format2 = new SimpleDateFormat("yyyy-MM-dd 23:59:59");

String strEnd = format2.format(time);

logger.debug("起始时间为：{}，结束时间为：{}",strStart,strEnd);

SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

Date startTime = null;

Date endTime = null;

try {

startTime = format.parse(strStart);

endTime = format.parse(strEnd);

}catch (Exception e){

logger.error(e.getMessage());

}

// predicate.getExpressions().add(builder.equal(root.get(LineTimeShareFlow\_.timestamp), time));

predicate.getExpressions().add(builder.between(root.get(LineTimeShareFlow\_.timestamp),startTime,endTime));

}

return predicate;

}

};

}

// Dynamic End

}

package com.avp.cdai.web.controller;

import com.avp.cdai.web.entity.ObjLine;

import com.avp.cdai.web.repository.ObjLineRepository;

import com.avp.cdai.web.rest.ResponseBuilder;

import com.avp.cdai.web.rest.ResponseCode;

import com.avp.cdai.web.rest.RestBody;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.\*;

import java.util.List;

/\*\*

\* Created by guo on 2017/8/4.

\*/

@RestController

public class ObjLineController {

@Autowired

ObjLineRepository objLineRepository;

private final Logger logger = LoggerFactory.getLogger(this.getClass());

@RequestMapping(value = "lines",method = RequestMethod.GET)

ResponseEntity<RestBody<ObjLine>> findAll(){

ResponseBuilder builder = ResponseBuilder.createBuilder();

try {

List<ObjLine> stations = objLineRepository.findAll();

logger.debug("查询站点成功,数据量为:{}", stations.size());

// System.out.println("查询站点成功,数据量为:" + stations.size());

for(int i = 0;i<stations.size();i++){

ObjLine obj = stations.get(i);

}

builder.setResultEntity(stations, ResponseCode.RETRIEVE\_SUCCEED);

return builder.getResponseEntity();

} catch (Exception e) {

logger.error(e.getMessage());

builder.setErrorCode(ResponseCode.RETRIEVE\_FAILED);

}

return builder.getResponseEntity();

}

@RequestMapping(value = "lines/{id}",method = RequestMethod.GET)

ResponseEntity<RestBody<ObjLine>> findById(@PathVariable("id") Integer idParam){

ResponseBuilder builder = ResponseBuilder.createBuilder();

try {

ObjLine objLine = objLineRepository.findById(idParam);

if(objLine != null){

builder.setResultEntity(objLine,ResponseCode.RETRIEVE\_SUCCEED);

return builder.getResponseEntity();

}

}

catch (Exception e){

logger.error(e.getMessage());

builder.setErrorCode(ResponseCode.RETRIEVE\_FAILED);

}

return builder.getResponseEntity();

}

@RequestMapping(value = "lines/findByName",method = RequestMethod.GET)

ResponseEntity<RestBody<ObjLine>> findByName(@RequestParam("linename") String lineName){

ResponseBuilder builder = ResponseBuilder.createBuilder();

try {

ObjLine objLine = objLineRepository.findBylineName(lineName);

builder.setResultEntity(objLine,ResponseCode.RETRIEVE\_SUCCEED);

return builder.getResponseEntity();

}catch (Exception e){

logger.error(e.getMessage());

builder.setErrorCode(ResponseCode.RETRIEVE\_FAILED);

}

return builder.getResponseEntity();

}

}

package com.avp.cdai.web.controller;

import com.avp.cdai.web.entity.ObjStation;

import com.avp.cdai.web.repository.ObjStationRepository;

import com.avp.cdai.web.rest.ResponseBuilder;

import com.avp.cdai.web.rest.ResponseCode;

import com.avp.cdai.web.rest.RestBody;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.\*;

import java.util.List;

/\*\*

\* Created by guo on 2017/8/8.

\*/

@RestController

public class ObjStationController {

@Autowired

ObjStationRepository objStationRepository;

private final Logger logger = LoggerFactory.getLogger(this.getClass());

@RequestMapping(value = "stations",method = RequestMethod.GET)

ResponseEntity<RestBody<ObjStation>> findAll(){

ResponseBuilder builder = ResponseBuilder.createBuilder();

try {

List<ObjStation> stations = objStationRepository.findAll();

if (stations != null) {

builder.setResultEntity(stations, ResponseCode.RETRIEVE\_SUCCEED);

logger.debug("站点的数量为：{}",stations.size());

return builder.getResponseEntity();

}

}catch (Exception e){

logger.error(e.getMessage());

builder.setErrorCode(ResponseCode.RETRIEVE\_FAILED);

}

return builder.getResponseEntity();

}

@RequestMapping(value = "stations/{id}/lines/{lineId}",method = RequestMethod.GET)

ResponseEntity<RestBody<ObjStation>> findById(@PathVariable("id") Integer id,@PathVariable("lineId") Integer lineId){

ResponseBuilder builder = ResponseBuilder.createBuilder();

try{

ObjStation station = objStationRepository.findBystationIdAndLineId(id,lineId);

// ObjStation station = null;

if(station != null) {

builder.setResultEntity(station, ResponseCode.RETRIEVE\_SUCCEED);

logger.debug("已找到对应站点,站点名称--{}！",station.getStationName());

return builder.getResponseEntity();

}

}catch (Exception e){

logger.error(e.getMessage());

builder.setErrorCode(ResponseCode.RETRIEVE\_FAILED);

}

return builder.getResponseEntity();

}

@RequestMapping(value = "stations/findbyname",method = RequestMethod.POST)

ResponseEntity<RestBody<ObjStation>> findByName(@RequestParam("stationName") String name){

ResponseBuilder builder = ResponseBuilder.createBuilder();

try{

ObjStation station = objStationRepository.findBystationName(name);

// ObjStation station = null;

logger.debug("已找到对应站点，站点名称--{}",station.getStationName());

builder.setResultEntity(station,ResponseCode.RETRIEVE\_SUCCEED);

}catch (Exception e){

logger.error(e.getMessage());

builder.setErrorCode(ResponseCode.RETRIEVE\_FAILED);

}

return builder.getResponseEntity();

}

}

package com.avp.cdai.web.controller;

import com.avp.cdai.web.entity.StationCumulativeFlow;

import com.avp.cdai.web.entity.StationCumulativeFlow\_;

import com.avp.cdai.web.repository.StationCumulativeFlowRepository;

import com.avp.cdai.web.rest.ResponseBuilder;

import com.avp.cdai.web.rest.ResponseCode;

import com.avp.cdai.web.rest.RestBody;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.data.jpa.domain.Specification;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.RequestParam;

import org.springframework.web.bind.annotation.RestController;

import javax.persistence.criteria.CriteriaBuilder;

import javax.persistence.criteria.CriteriaQuery;

import javax.persistence.criteria.Predicate;

import javax.persistence.criteria.Root;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.List;

import static org.springframework.data.jpa.domain.Specifications.where;

/\*\*

\* Created by guo on 2017/8/15.

\*/

@RestController

public class StationCumulativeFlowController {

@Autowired

StationCumulativeFlowRepository stationCumulativeFlowRepository;

private final Logger logger = LoggerFactory.getLogger(this.getClass());

@RequestMapping(value = "stationSumFlow",method = RequestMethod.GET)

ResponseEntity<RestBody<StationCumulativeFlow>> findAll(){

ResponseBuilder builder = ResponseBuilder.createBuilder();

try {

List<StationCumulativeFlow> list = stationCumulativeFlowRepository.findAll();

logger.debug("车站客流累计查询记录数量为：{}",list.size());

builder.setResultEntity(list, ResponseCode.RETRIEVE\_SUCCEED);

return builder.getResponseEntity();

}catch (Exception e){

logger.error(e.getMessage());

builder.setErrorCode(ResponseCode.RETRIEVE\_FAILED);

}

return builder.getResponseEntity();

}

@RequestMapping(value = "stationSumByConditions",method = RequestMethod.GET)

ResponseEntity<RestBody<StationCumulativeFlow>> findByConditions(@RequestParam("ids") List<Integer> ids, @RequestParam(value = "direct",required = false) Integer direct, Integer section, Date time){

ResponseBuilder builder = ResponseBuilder.createBuilder();

try {

List<StationCumulativeFlow> list = stationCumulativeFlowRepository.findAll(where(byConditions(ids, direct,section, time)));

if (list.size() <= 0) {

logger.debug("车站客流累计多条件查询失败，结果为空");

} else {

logger.debug("车站客流累计多条件查询成功，数据量为:({})", list.size());

}

builder.setResultEntity(list, ResponseCode.RETRIEVE\_SUCCEED);

} catch (Exception e) {

logger.error(e.getMessage());

builder.setErrorCode(ResponseCode.RETRIEVE\_FAILED);

}

return builder.getResponseEntity();

}

// Dynamic Query Utils

public Specification<StationCumulativeFlow> byConditions(List<Integer> ids, Integer direct, Integer section, Date time) {

return new Specification<StationCumulativeFlow>() {

public Predicate toPredicate(Root<StationCumulativeFlow> root, CriteriaQuery<?> query, CriteriaBuilder builder) {

Predicate predicate = builder.conjunction();

logger.debug("stationSumByConditions请求的参数ids值为:{}", ids);

if (ids != null) {

// predicate.getExpressions().add(builder.equal(root.get(LineTimeShareFlow\_.id), ids));

// predicate.getExpressions().add(builder.in(root.get(LineTimeShareFlow\_.lineId)).in(ids));

predicate.getExpressions().add(root.<Integer>get(StationCumulativeFlow\_.stationId).in(ids));

}

//

logger.debug("stationSumByConditions请求的参数direct值为:{}", direct);

if (direct != null) {

predicate.getExpressions().add(builder.equal(root.get(StationCumulativeFlow\_.direction), direct));

}

logger.debug("stationSumByConditions请求的参数section值为:{}", section);

if (section != null) {

predicate.getExpressions().add(builder.equal(root.get(StationCumulativeFlow\_.section), section));

}

//

logger.debug("stationSumByConditions请求的参数time值为:{}", time);

if (time != null) {

SimpleDateFormat format1 = new SimpleDateFormat("yyyy-MM-dd 00:00:00");

String strStart = format1.format(time);

SimpleDateFormat format2 = new SimpleDateFormat("yyyy-MM-dd 23:59:59");

String strEnd = format2.format(time);

logger.debug("起始时间为：{}，结束时间为：{}",strStart,strEnd);

SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

Date startTime = null;

Date endTime = null;

try {

startTime = format.parse(strStart);

endTime = format.parse(strEnd);

}catch (Exception e){

logger.error(e.getMessage());

}

// predicate.getExpressions().add(builder.equal(root.get(StationCumulativeFlow\_.flowTime), time));

predicate.getExpressions().add(builder.between(root.get(StationCumulativeFlow\_.flowTime),startTime,endTime));

}

return predicate;

}

};

}

// Dynamic End

}

package com.avp.cdai.web.controller;

import com.avp.cdai.web.entity.ObjStation;

import com.avp.cdai.web.entity.StationShareFlow;

import com.avp.cdai.web.entity.StationShareFlow\_;

import com.avp.cdai.web.entity.ViewData;

import com.avp.cdai.web.repository.ObjStationRepository;

import com.avp.cdai.web.repository.StationShareFlowRepository;

import com.avp.cdai.web.rest.ResponseBuilder;

import com.avp.cdai.web.rest.ResponseCode;

import com.avp.cdai.web.rest.RestBody;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.data.jpa.domain.Specification;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.RequestParam;

import org.springframework.web.bind.annotation.RestController;

import javax.persistence.criteria.CriteriaBuilder;

import javax.persistence.criteria.CriteriaQuery;

import javax.persistence.criteria.Predicate;

import javax.persistence.criteria.Root;

import java.text.SimpleDateFormat;

import java.util.\*;

import static org.springframework.data.jpa.domain.Specifications.where;

/\*\*

\* Created by guo on 2017/8/11.

\*/

@RestController

public class StationShareFlowController {

@Autowired

StationShareFlowRepository stationShareFlowRepository;

@Autowired

ObjStationRepository objStationRepository;

private final Logger logger = LoggerFactory.getLogger(this.getClass());

@RequestMapping(value = "stationShareFlow",method = RequestMethod.GET)

ResponseEntity<RestBody<StationShareFlow>> findAll(){

ResponseBuilder builder = ResponseBuilder.createBuilder();

try {

List<StationShareFlow> list = stationShareFlowRepository.findAll();

logger.debug("车站客流分时记录查询成功，数量为{}",list.size());

builder.setResultEntity(list,ResponseCode.RETRIEVE\_SUCCEED);

return builder.getResponseEntity();

}catch (Exception e){

logger.error(e.getMessage());

builder.setErrorCode(ResponseCode.RETRIEVE\_FAILED);

}

return builder.getResponseEntity();

}

@RequestMapping(value = "stationShareByConditions",method = RequestMethod.GET)

ResponseEntity<RestBody<StationShareFlow>> findByConditions(@RequestParam("ids") List<Integer> ids, @RequestParam(value = "direct",required = false) Integer direct, Integer section, Date time){

ResponseBuilder builder = ResponseBuilder.createBuilder();

try {

List<StationShareFlow> list = stationShareFlowRepository.findAll(where(byConditions(ids, direct,section, time)));

if (list.size() <= 0) {

logger.debug("车站客流分时多条件查询失败，结果为空");

} else {

logger.debug("车站客流分时多条件查询成功，数据量为:({})", list.size());

}

builder.setResultEntity(list, ResponseCode.RETRIEVE\_SUCCEED);

} catch (Exception e) {

logger.error(e.getMessage());

builder.setErrorCode(ResponseCode.RETRIEVE\_FAILED);

}

return builder.getResponseEntity();

}

@RequestMapping(value = "viewDataFindByConditions",method = RequestMethod.GET)

ResponseEntity<RestBody<ViewData<ObjStation>>> viewDataFindByConditions(@RequestParam("ids") List<Integer> ids, @RequestParam(value = "direct",required = false) Integer direct, Integer section, Date time){

ResponseBuilder builder = ResponseBuilder.createBuilder();

try {

/\*

List<StationShareFlow> list = stationShareFlowRepository.findAll(where(byConditions(ids, direct,section, time)));

ViewData<StationShareFlow> viewData = new ViewData<StationShareFlow>();

Map<Integer,List<Integer>> flowMap = new HashMap<Integer,List<Integer>>();

if (list.size() <= 0) {

logger.debug("车站客流分时多条件查询失败，结果为空");

} else {

logger.debug("车站客流分时多条件查询成功，数据量为:({})", list.size());

}\*/

// List<String> list = stationShareFlowRepository.getData(ids.get(0), section);

Date startTime = null;

Date endTime = null;

if (time != null) {

SimpleDateFormat format1 = new SimpleDateFormat("yyyy-MM-dd 00:00:00");

String strStart = format1.format(time);

SimpleDateFormat format2 = new SimpleDateFormat("yyyy-MM-dd 23:59:59");

String strEnd = format2.format(time);

logger.debug("起始时间为：{}，结束时间为：{}", strStart, strEnd);

SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

try {

startTime = format.parse(strStart);

endTime = format.parse(strEnd);

} catch (Exception e) {

logger.error(e.getMessage());

}

}

logger.debug("ids数据量为:({})", ids.size());

ViewData<ObjStation> viewData = new ViewData<ObjStation>();

Map<Integer,List<Integer>> flowMap = new HashMap<Integer,List<Integer>>();

Map<Integer,ObjStation> objMap = new HashMap<Integer,ObjStation>();

List<Date> dateList = null;

//取范围较大的时间轴列表

Integer tempCount = 0;

int a = 0;

for(int i = 0;i<ids.size();i++) {

List<Integer> temp = new ArrayList<Integer>();

// List<StationShareFlow> list = stationShareFlowRepository.getData(ids.get(i), section);

List<StationShareFlow> list = stationShareFlowRepository.getData(ids.get(i), section,startTime,endTime);

logger.debug("数据量为:({},{})", list.size(),ids.get(i));

Integer key = null;

List<Date> tempDateList = new ArrayList<Date>();

for (StationShareFlow s:list){

temp.add(s.getFlowCount());

tempDateList.add(s.getFlowTime());

}

if(list.size() > tempCount) {

tempCount = list.size();

dateList = tempDateList;

}

a++;

logger.debug("aaa:({})", a);

flowMap.put(ids.get(i),temp);

}

logger.debug("flowMap数据量为:({})", flowMap.size());

// List<ObjStation> objList = stationShareFlowRepository.getObjData();

List<ObjStation> objList = objStationRepository.findBystationIdIn(ids);

logger.debug("ObjStation数据量为:({})", objList.size());

for (ObjStation obj:objList){

Integer s\_id = obj.getStationId();

if(s\_id == null){

logger.debug("s\_id为null!");

}

objMap.put(s\_id,obj);

}

viewData.setFlowCountMap(flowMap);

viewData.setObjMap(objMap);

viewData.setDateList(dateList);

builder.setResultEntity(viewData,ResponseCode.RETRIEVE\_SUCCEED);

} catch (Exception e) {

logger.error(e.getMessage());

builder.setErrorCode(ResponseCode.RETRIEVE\_FAILED);

}

return builder.getResponseEntity();

}

// Dynamic Query Utils

public Specification<StationShareFlow> byConditions(List<Integer> ids, Integer direct, Integer section, Date time) {

return new Specification<StationShareFlow>() {

public Predicate toPredicate(Root<StationShareFlow> root, CriteriaQuery<?> query, CriteriaBuilder builder) {

Predicate predicate = builder.conjunction();

logger.debug("stationShareByConditions请求的参数ids值为:{}", ids);

if (ids != null) {

// predicate.getExpressions().add(builder.equal(root.get(LineTimeShareFlow\_.id), ids));

// predicate.getExpressions().add(builder.in(root.get(LineTimeShareFlow\_.lineId)).in(ids));

predicate.getExpressions().add(root.get(StationShareFlow\_.stationId).in(ids));

}

//

logger.debug("stationShareByConditions请求的参数direct值为:{}", direct);

if (direct != null) {

predicate.getExpressions().add(builder.equal(root.get(StationShareFlow\_.direction), direct));

}

logger.debug("stationShareByConditions请求的参数section值为:{}", section);

if (section != null) {

predicate.getExpressions().add(builder.equal(root.get(StationShareFlow\_.section), section));

}

//

logger.debug("stationShareByConditions请求的参数time值为:{}", time);

if (time != null) {

SimpleDateFormat format1 = new SimpleDateFormat("yyyy-MM-dd 00:00:00");

String strStart = format1.format(time);

SimpleDateFormat format2 = new SimpleDateFormat("yyyy-MM-dd 23:59:59");

String strEnd = format2.format(time);

logger.debug("起始时间为：{}，结束时间为：{}",strStart,strEnd);

SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

Date startTime = null;

Date endTime = null;

try {

startTime = format.parse(strStart);

endTime = format.parse(strEnd);

}catch (Exception e){

logger.error(e.getMessage());

}

// predicate.getExpressions().add(builder.equal(root.get(StationShareFlow\_.flowTime), time));

predicate.getExpressions().add(builder.between(root.get(StationShareFlow\_.flowTime),startTime,endTime));

}

return predicate;

}

};

}

// Dynamic End

}

package com.avp.cdai.web.controller;

import com.avp.cdai.web.entity.TicketCumulativeFlow;

import com.avp.cdai.web.entity.TicketCumulativeFlow\_;

import com.avp.cdai.web.repository.TicketCumulativeFlowRepository;

import com.avp.cdai.web.rest.ResponseBuilder;

import com.avp.cdai.web.rest.ResponseCode;

import com.avp.cdai.web.rest.RestBody;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.data.jpa.domain.Specification;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.RequestParam;

import org.springframework.web.bind.annotation.RestController;

import javax.persistence.criteria.CriteriaBuilder;

import javax.persistence.criteria.CriteriaQuery;

import javax.persistence.criteria.Predicate;

import javax.persistence.criteria.Root;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.List;

import static org.springframework.data.jpa.domain.Specifications.where;

/\*\*

\* Created by guo on 2017/8/17.

\*/

@RestController

public class TicketCumulativeFlowController {

@Autowired

TicketCumulativeFlowRepository ticketCumulativeFlowRepository;

private final Logger logger = LoggerFactory.getLogger(this.getClass());

@RequestMapping(value = "ticketSumFlow",method = RequestMethod.GET)

ResponseEntity<RestBody<TicketCumulativeFlow>> findAll(){

ResponseBuilder builder = ResponseBuilder.createBuilder();

try {

List<TicketCumulativeFlow> list = ticketCumulativeFlowRepository.findAll();

logger.debug("票卡客流累计记录查询成功，记录数量为{}",list.size());

builder.setResultEntity(list,ResponseCode.RETRIEVE\_SUCCEED);

}catch (Exception e){

logger.error(e.getMessage());

builder.setErrorCode(ResponseCode.RETRIEVE\_FAILED);

}

return builder.getResponseEntity();

}

@RequestMapping(value = "ticketSumByConditions",method = RequestMethod.GET)

ResponseEntity<RestBody<TicketCumulativeFlow>> findByConditions(@RequestParam("ids") List<Integer> ids, @RequestParam(value = "lineIds",required = false) List<Integer> lineIds, @RequestParam(value = "stationIds",required = false) List<Integer> stationIds, @RequestParam(value = "direct",required = false) Integer direct, Integer section, Date time){

ResponseBuilder builder = ResponseBuilder.createBuilder();

try {

List<TicketCumulativeFlow> list = ticketCumulativeFlowRepository.findAll(where(byConditions(ids,lineIds,stationIds, direct,section, time)));

if (list.size() <= 0) {

logger.debug("票卡客流累计多条件查询失败，结果为空");

} else {

logger.debug("票卡客流累计多条件查询成功，数据量为:({})", list.size());

}

builder.setResultEntity(list, ResponseCode.RETRIEVE\_SUCCEED);

} catch (Exception e) {

logger.error(e.getMessage());

builder.setErrorCode(ResponseCode.RETRIEVE\_FAILED);

}

return builder.getResponseEntity();

}

// Dynamic Query Utils

public Specification<TicketCumulativeFlow> byConditions(List<Integer> ids, List<Integer> lineIds, List<Integer> stationIds, Integer direct, Integer section, Date time) {

return new Specification<TicketCumulativeFlow>() {

public Predicate toPredicate(Root<TicketCumulativeFlow> root, CriteriaQuery<?> query, CriteriaBuilder builder) {

Predicate predicate = builder.conjunction();

logger.debug("ticketSumByConditions请求的参数ids值为:{}", ids);

if (ids != null) {

// predicate.getExpressions().add(builder.equal(root.get(LineTimeShareFlow\_.id), ids));

// predicate.getExpressions().add(builder.in(root.get(LineTimeShareFlow\_.lineId)).in(ids));

predicate.getExpressions().add(root.<Integer>get(TicketCumulativeFlow\_.ticketId).in(ids));

}

logger.debug("ticketSumByConditions请求的参数lineIds值为:{}", lineIds);

if (lineIds != null) {

// predicate.getExpressions().add(builder.equal(root.get(LineTimeShareFlow\_.id), ids));

// predicate.getExpressions().add(builder.in(root.get(LineTimeShareFlow\_.lineId)).in(ids));

predicate.getExpressions().add(root.<Integer>get(TicketCumulativeFlow\_.lineId).in(lineIds));

}

logger.debug("ticketSumByConditions请求的参数stationIds值为:{}", stationIds);

if (stationIds != null) {

// predicate.getExpressions().add(builder.equal(root.get(LineTimeShareFlow\_.id), ids));

// predicate.getExpressions().add(builder.in(root.get(LineTimeShareFlow\_.lineId)).in(ids));

predicate.getExpressions().add(root.<Integer>get(TicketCumulativeFlow\_.stationId).in(stationIds));

}

//

logger.debug("ticketSumByConditions请求的参数direct值为:{}", direct);

if (direct != null) {

predicate.getExpressions().add(builder.equal(root.get(TicketCumulativeFlow\_.direction), direct));

}

logger.debug("ticketSumByConditions请求的参数section值为:{}", section);

if (section != null) {

predicate.getExpressions().add(builder.equal(root.get(TicketCumulativeFlow\_.section), section));

}

//

logger.debug("ticketSumByConditions请求的参数time值为:{}", time);

if (time != null) {

SimpleDateFormat format1 = new SimpleDateFormat("yyyy-MM-dd 00:00:00");

String strStart = format1.format(time);

SimpleDateFormat format2 = new SimpleDateFormat("yyyy-MM-dd 23:59:59");

String strEnd = format2.format(time);

logger.debug("起始时间为：{}，结束时间为：{}",strStart,strEnd);

SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

Date startTime = null;

Date endTime = null;

try {

startTime = format.parse(strStart);

endTime = format.parse(strEnd);

}catch (Exception e){

logger.error(e.getMessage());

}

// predicate.getExpressions().add(builder.equal(root.get(LineCumulativeFlow\_.flowTime), time));

predicate.getExpressions().add(builder.between(root.get(TicketCumulativeFlow\_.timestamp),startTime,endTime));

}

return predicate;

}

};

}

// Dynamic End

}

package com.avp.cdai.web.controller;

import com.avp.cdai.web.entity.TicketShareFlow;

import com.avp.cdai.web.entity.TicketShareFlow\_;

import com.avp.cdai.web.repository.TicketShareFlowRepository;

import com.avp.cdai.web.rest.ResponseBuilder;

import com.avp.cdai.web.rest.ResponseCode;

import com.avp.cdai.web.rest.RestBody;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.data.jpa.domain.Specification;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.RequestParam;

import org.springframework.web.bind.annotation.RestController;

import javax.persistence.criteria.CriteriaBuilder;

import javax.persistence.criteria.CriteriaQuery;

import javax.persistence.criteria.Predicate;

import javax.persistence.criteria.Root;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.List;

import static org.springframework.data.jpa.domain.Specifications.where;

/\*\*

\* Created by guo on 2017/8/16.

\*/

@RestController

public class TicketShareFlowController {

@Autowired

TicketShareFlowRepository ticketShareFlowRepository;

private final Logger logger = LoggerFactory.getLogger(this.getClass());

@RequestMapping(value = "ticketShareFlow",method = RequestMethod.GET)

ResponseEntity<RestBody<TicketShareFlow>> findAll(){

ResponseBuilder builder = ResponseBuilder.createBuilder();

try {

List<TicketShareFlow> list = ticketShareFlowRepository.findAll();

logger.debug("票卡客流分时记录查询数量为：{}",list.size());

builder.setResultEntity(list,ResponseCode.RETRIEVE\_SUCCEED);

}catch (Exception e){

logger.error(e.getMessage());

builder.setErrorCode(ResponseCode.RETRIEVE\_FAILED);

}

return builder.getResponseEntity();

}

@RequestMapping(value = "ticketShareByConditions",method = RequestMethod.GET)

ResponseEntity<RestBody<TicketShareFlow>> findByConditions(@RequestParam("ids") List<Integer> ids,@RequestParam(value = "lineIds",required = false) List<Integer> lineIds,@RequestParam(value = "stationIds",required = false) List<Integer> stationIds, @RequestParam(value = "direct",required = false) Integer direct, Integer section, Date time){

ResponseBuilder builder = ResponseBuilder.createBuilder();

try {

List<TicketShareFlow> list = ticketShareFlowRepository.findAll(where(byConditions(ids,lineIds,stationIds, direct,section, time)));

if (list.size() <= 0) {

logger.debug("票卡客流分时多条件查询失败，结果为空");

} else {

logger.debug("票卡客流分时多条件查询成功，数据量为:({})", list.size());

}

builder.setResultEntity(list, ResponseCode.RETRIEVE\_SUCCEED);

} catch (Exception e) {

logger.error(e.getMessage());

builder.setErrorCode(ResponseCode.RETRIEVE\_FAILED);

}

return builder.getResponseEntity();

}

// Dynamic Query Utils

public Specification<TicketShareFlow> byConditions(List<Integer> ids,List<Integer> lineIds, List<Integer> stationIds,Integer direct, Integer section, Date time) {

return new Specification<TicketShareFlow>() {

public Predicate toPredicate(Root<TicketShareFlow> root, CriteriaQuery<?> query, CriteriaBuilder builder) {

Predicate predicate = builder.conjunction();

logger.debug("ticketShareByConditions请求的参数ids值为:{}", ids);

if (ids != null) {

// predicate.getExpressions().add(builder.equal(root.get(LineTimeShareFlow\_.id), ids));

// predicate.getExpressions().add(builder.in(root.get(LineTimeShareFlow\_.lineId)).in(ids));

predicate.getExpressions().add(root.<Integer>get(TicketShareFlow\_.ticketId).in(ids));

}

logger.debug("ticketShareByConditions请求的参数lineIds值为:{}", lineIds);

if (lineIds != null) {

// predicate.getExpressions().add(builder.equal(root.get(LineTimeShareFlow\_.id), ids));

// predicate.getExpressions().add(builder.in(root.get(LineTimeShareFlow\_.lineId)).in(ids));

predicate.getExpressions().add(root.<Integer>get(TicketShareFlow\_.lineId).in(lineIds));

}

logger.debug("ticketShareByConditions请求的参数stationIds值为:{}", stationIds);

if (stationIds != null) {

// predicate.getExpressions().add(builder.equal(root.get(LineTimeShareFlow\_.id), ids));

// predicate.getExpressions().add(builder.in(root.get(LineTimeShareFlow\_.lineId)).in(ids));

predicate.getExpressions().add(root.<Integer>get(TicketShareFlow\_.stationId).in(stationIds));

}

//

logger.debug("ticketShareByConditions请求的参数direct值为:{}", direct);

if (direct != null) {

predicate.getExpressions().add(builder.equal(root.get(TicketShareFlow\_.direction), direct));

}

logger.debug("ticketShareByConditions请求的参数section值为:{}", section);

if (section != null) {

predicate.getExpressions().add(builder.equal(root.get(TicketShareFlow\_.section), section));

}

//

logger.debug("ticketShareByConditions请求的参数time值为:{}", time);

if (time != null) {

SimpleDateFormat format1 = new SimpleDateFormat("yyyy-MM-dd 00:00:00");

String strStart = format1.format(time);

SimpleDateFormat format2 = new SimpleDateFormat("yyyy-MM-dd 23:59:59");

String strEnd = format2.format(time);

logger.debug("起始时间为：{}，结束时间为：{}",strStart,strEnd);

SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

Date startTime = null;

Date endTime = null;

try {

startTime = format.parse(strStart);

endTime = format.parse(strEnd);

}catch (Exception e){

logger.error(e.getMessage());

}

// predicate.getExpressions().add(builder.equal(root.get(LineCumulativeFlow\_.flowTime), time));

predicate.getExpressions().add(builder.between(root.get(TicketShareFlow\_.timestamp),startTime,endTime));

}

return predicate;

}

};

}

// Dynamic End

}

package com.avp.cdai.web.controller;

import com.avp.cdai.web.entity.TicketType;

import com.avp.cdai.web.repository.TicketTypeRepository;

import com.avp.cdai.web.rest.ResponseBuilder;

import com.avp.cdai.web.rest.ResponseCode;

import com.avp.cdai.web.rest.RestBody;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.RestController;

import java.util.List;

/\*\*

\* Created by guo on 2017/8/16.

\*/

@RestController

public class TicketTypeController {

@Autowired

TicketTypeRepository ticketTypeRepository;

private final Logger logger = LoggerFactory.getLogger(this.getClass());

@RequestMapping(value = "tickets",method = RequestMethod.GET)

ResponseEntity<RestBody<TicketType>> findAll(){

ResponseBuilder builder = ResponseBuilder.createBuilder();

try {

List<TicketType> list = ticketTypeRepository.findAll();

logger.debug("票卡类型记录，数量为：{}",list.size());

builder.setResultEntity(list,ResponseCode.RETRIEVE\_SUCCEED);

}catch (Exception e){

logger.error(e.getMessage());

builder.setErrorCode(ResponseCode.RETRIEVE\_FAILED);

}

return builder.getResponseEntity();

}

}

package com.avp.cdai.web.rest;

import com.fasterxml.jackson.annotation.JsonIgnore;

import com.fasterxml.jackson.annotation.JsonProperty;

import org.springframework.data.domain.Page;

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

/\*\*

\* Created by boris feng on 2017/5/11.

\*

\* ResponseBuilder<T> builer = ResponseBuilder.createBuilder(request);

\* builder.setResultEntity(entity, ResponseCode.OK);

\* return builder.getResponseEntity();

\*

\*/

public class ResponseBuilder<T> {

private ResponseEntity<RestBody<T>> response

= new ResponseEntity<RestBody<T>>(new RestBody<T>(ResponseCode.FAILED), HttpStatus.INTERNAL\_SERVER\_ERROR);

class EntityBody<T> extends RestBody<T> {

private T result;

EntityBody(ResponseCode code) {

super(code);

}

EntityBody(T result, ResponseCode code) {

super(code);

this.result = result;

}

@JsonProperty("result")

public T getResult() {

return this.result;

}

}

class ListBody<T> extends EntityBody<T> {

private List<T> results = new ArrayList<T>();

ListBody(ResponseCode code) {

super(code);

}

ListBody(List<T> results, ResponseCode code) {

super(code);

this.results.addAll(results);

}

@Override

@JsonIgnore

public T getResult() {

return null;

}

@JsonProperty(value = "results")

public List<T> getResults() {

return this.results;

}

}

class PageBody<T> extends ListBody<T> {

private Page<T> page;

PageBody(Page<T> page, ResponseCode code) {

super(code);

this.page = page;

}

@Override

@JsonProperty(value = "results")

public List<T> getResults() {

return this.page.getContent();

}

@JsonProperty(value = "page")

public Map<String, Object> getPage() {

Map<String, Object> page = new HashMap<>();

page.put("size", this.page.getSize());

page.put("totalElements", this.page.getTotalElements());

page.put("totalPages", this.page.getTotalPages());

page.put("number", this.page.getNumber());

page.put("hasNext", this.page.hasNext());

page.put("hasPrevious", this.page.hasPrevious());

return page;

}

}

public static <T> ResponseBuilder createBuilder() {

return new ResponseBuilder<T>();

}

public ResponseEntity<RestBody<T>> getResponseEntity() {

return this.response;

}

public void setErrorCode(ResponseCode code) {

this.response = new ResponseEntity<>(new RestBody(code), getHttpStatus(code));

}

public void setErrorCode(ResponseCode code, String message) {

this.setErrorCode(code);

this.response.getBody().setMessage(message);

}

public <T> ResponseBuilder setResultEntity(T entity) {

return this.setResultEntity(entity, ResponseCode.OK);

}

public <T> ResponseBuilder setResultEntity(T entity, ResponseCode code) {

this.response = entity == null ?

new ResponseEntity<>(HttpStatus.NOT\_FOUND)

: new ResponseEntity(new EntityBody<T>(entity, code), getHttpStatus(code));

return this;

}

public <T> ResponseBuilder setResultEntity(List<T> entities) {

return this.setResultEntity(entities, ResponseCode.OK);

}

public <T> ResponseBuilder setResultEntity(List<T> entities, ResponseCode code) {

this.response = new ResponseEntity(new ListBody<T>(entities, code), getHttpStatus(code));

return this;

}

public <T> ResponseBuilder setResultEntity(Page<T> page, ResponseCode code) {

this.response = new ResponseEntity(new PageBody<>( page, code), getHttpStatus(code));

return this;

}

public <T> ResponseBuilder setResultEntity(Page<T> page) {

return this.setResultEntity(page, ResponseCode.OK);

}

private HttpStatus getHttpStatus(ResponseCode code) {

if (!\_router.containsKey(code))

return HttpStatus.INTERNAL\_SERVER\_ERROR;

return \_router.get(code);

}

static private Map<ResponseCode, HttpStatus> \_router = new HashMap<ResponseCode, HttpStatus>() {

{

put(ResponseCode.OK, HttpStatus.OK);

put(ResponseCode.FAILED, HttpStatus.INTERNAL\_SERVER\_ERROR);

put(ResponseCode.CREATE\_SUCCEED, HttpStatus.OK);

put(ResponseCode.CREATE\_FAILED, HttpStatus.INTERNAL\_SERVER\_ERROR);

put(ResponseCode.DELETE\_SUCCEED, HttpStatus.OK);

put(ResponseCode.DELETE\_FAILED, HttpStatus.INTERNAL\_SERVER\_ERROR);

put(ResponseCode.UPDATE\_SUCCEED, HttpStatus.OK);

put(ResponseCode.UPDATE\_FAILED, HttpStatus.INTERNAL\_SERVER\_ERROR);

put(ResponseCode.RETRIEVE\_SUCCEED, HttpStatus.OK);

put(ResponseCode.RETRIEVE\_FAILED, HttpStatus.INTERNAL\_SERVER\_ERROR);

put(ResponseCode.ALREADY\_EXIST, HttpStatus.BAD\_REQUEST);

put(ResponseCode.NOT\_EXIST, HttpStatus.BAD\_REQUEST);

put(ResponseCode.BAD\_REQUEST, HttpStatus.BAD\_REQUEST);

put(ResponseCode.PARAM\_MISSING, HttpStatus.BAD\_REQUEST);

}

};

}

package com.avp.cdai.web.rest;

import java.util.Date;

/\*\*

\* Created by boris feng on 2017/5/27.

\*/

public class RestBody<T> {

private Integer code = -1;

private String message = "unknown";

private Long timestamp = new Date().getTime();

RestBody(ResponseCode code) {

this.code = code.value();

this.message = code.phrase();

}

public String getMessage() {

return message;

}

public void setMessage(String message) {

this.message = message;

}

public Integer getCode() {

return this.code;

}

public Long getTimestamp() {

return this.timestamp;

}

}

package com.avp.cdai.web.rest;

/\*\*

\* Created by boris feng on 2017/5/23.

\*/

public enum ResponseCode {

OK (0, "成功"),

FAILED (-1, "失败"),

CREATE\_SUCCEED(10000, "新建成功"),

CREATE\_FAILED(-10000, "新建失败"),

DELETE\_SUCCEED(20000, "删除成功"),

DELETE\_FAILED(-20000, "删除失败"),

UPDATE\_SUCCEED(30000, "修改成功"),

UPDATE\_FAILED(-30000, "修改失败"),

RETRIEVE\_SUCCEED(40000, "查询成功"),

RETRIEVE\_FAILED(-40000, "查询失败"),

ALREADY\_EXIST(-50000, "数据已存在"),

NOT\_EXIST(-50001, "数据不存在"),

BAD\_REQUEST(-60000, "错误请求"),

PARAM\_MISSING(-70000, "参数缺失"),

NOT\_IMPLEMENTED(-80000, "未实现");

private final Integer value;

private final String phrase;

private ResponseCode(int value, String phrase) {

this.value = value;

this.phrase = phrase;

}

public Integer value() {

return this.value;

}

public String phrase() {

return this.phrase;

}

}

package com.avp.cdai.web.web;

import org.springframework.core.Ordered;

import org.springframework.core.annotation.Order;

import org.springframework.stereotype.Component;

import javax.servlet.\*;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import java.io.IOException;

@Component

@Order(Ordered.HIGHEST\_PRECEDENCE)

public class CorsFilter implements Filter {

@Override

public void doFilter(ServletRequest req, ServletResponse res, FilterChain chain)

throws IOException, ServletException {

HttpServletResponse response = (HttpServletResponse) res;

HttpServletRequest request = (HttpServletRequest) req;

response.setHeader("Access-Control-Allow-Origin", "\*");

response.setHeader("Access-Control-Allow-Methods", "POST, PUT, GET, OPTIONS, DELETE");

response.setHeader("Access-Control-Allow-Headers", "Access-Control-Allow-Headers, Origin,Accept, X-Requested-With, Content-Type, Access-Control-Request-Method, Access-Control-Request-Headers,authorization");

response.setHeader("Access-Control-Max-Age", "3600");

if (!request.getMethod().equalsIgnoreCase("OPTIONS")) {

chain.doFilter(req, res);

} else {

}

}

@Override

public void init(FilterConfig fc) throws ServletException {

}

@Override

public void destroy() {

}

}

<!DOCTYPE html>

<!--

This is a starter template page. Use this page to start your new project from

scratch. This page gets rid of all links and provides the needed markup only.

-->

<html>

<head>

<meta charset="utf-8"/>

<meta http-equiv="X-UA-Compatible" content="IE=edge"/>

<title>AdminLTE 2 | Starter</title>

<!-- Tell the browser to be responsive to screen width -->

<meta content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no" name="viewport"/>

<!-- Bootstrap 3.3.6 -->

<link rel="stylesheet" href="bootstrap/css/bootstrap.min.css"/>

<!-- Font Awesome -->

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.5.0/css/font-awesome.min.css"/>

<!-- Ionicons -->

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/ionicons/2.0.1/css/ionicons.min.css"/>

<!-- Theme style -->

<link rel="stylesheet" href="dist/css/AdminLTE.min.css"/>

<!-- AdminLTE Skins. We have chosen the skin-blue for this starter

page. However, you can choose any other skin. Make sure you

apply the skin class to the body tag so the changes take effect.

-->

<link rel="stylesheet" href="dist/css/skins/skin-blue.min.css"/>

<!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media queries -->

<!-- WARNING: Respond.js doesn't work if you view the page via file:// -->

<!--[if lt IE 9]>

<script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>

<script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>

<![endif]-->

</head>

<body class="hold-transition skin-blue sidebar-mini">

<div class="wrapper">

<!-- Main Header -->

<header class="main-header">

<!-- Logo -->

<a href="index2.html" class="logo">

<!-- mini logo for sidebar mini 50x50 pixels -->

<span class="logo-mini"><b>A</b>LT</span>

<!-- logo for regular state and mobile devices -->

<span class="logo-lg"><b>Admin</b>LTE</span>

</a>

<!-- Header Navbar -->

<nav class="navbar navbar-static-top" role="navigation">

<!-- Sidebar toggle button-->

<a href="#" class="sidebar-toggle" data-toggle="offcanvas" role="button">

<span class="sr-only">Toggle navigation</span>

</a>

<!-- Navbar Right Menu -->

<div class="navbar-custom-menu">

<ul class="nav navbar-nav">

<!-- Messages: style can be found in dropdown.less-->

<li class="dropdown messages-menu">

<!-- Menu toggle button -->

<a href="#" class="dropdown-toggle" data-toggle="dropdown">

<i class="fa fa-envelope-o"></i>

<span class="label label-success">4</span>

</a>

<ul class="dropdown-menu">

<li class="header">You have 4 messages</li>

<li>

<!-- inner menu: contains the messages -->

<ul class="menu">

<li><!-- start message -->

<a href="#">

<div class="pull-left">

<!-- User Image -->

<img src="dist/img/user2-160x160.jpg" class="img-circle" alt="User Image"/>

</div>

<!-- Message title and timestamp -->

<h4>

Support Team

<small><i class="fa fa-clock-o"></i> 5 mins</small>

</h4>

<!-- The message -->

<p>Why not buy a new awesome theme?</p>

</a>

</li>

<!-- end message -->

</ul>

<!-- /.menu -->

</li>

<li class="footer"><a href="#">See All Messages</a></li>

</ul>

</li>

<!-- /.messages-menu -->

<!-- Notifications Menu -->

<li class="dropdown notifications-menu">

<!-- Menu toggle button -->

<a href="#" class="dropdown-toggle" data-toggle="dropdown">

<i class="fa fa-bell-o"></i>

<span class="label label-warning">10</span>

</a>

<ul class="dropdown-menu">

<li class="header">You have 10 notifications</li>

<li>

<!-- Inner Menu: contains the notifications -->

<ul class="menu">

<li><!-- start notification -->

<a href="#">

<i class="fa fa-users text-aqua"></i> 5 new members joined today

</a>

</li>

<!-- end notification -->

</ul>

</li>

<li class="footer"><a href="#">View all</a></li>

</ul>

</li>

<!-- Tasks Menu -->

<li class="dropdown tasks-menu">

<!-- Menu Toggle Button -->

<a href="#" class="dropdown-toggle" data-toggle="dropdown">

<i class="fa fa-flag-o"></i>

<span class="label label-danger">9</span>

</a>

<ul class="dropdown-menu">

<li class="header">You have 9 tasks</li>

<li>

<!-- Inner menu: contains the tasks -->

<ul class="menu">

<li><!-- Task item -->

<a href="#">

<!-- Task title and progress text -->

<h3>

Design some buttons

<small class="pull-right">20%</small>

</h3>

<!-- The progress bar -->

<div class="progress xs">

<!-- Change the css width attribute to simulate progress -->

<div class="progress-bar progress-bar-aqua" style="width: 20%" role="progressbar" aria-valuenow="20" aria-valuemin="0" aria-valuemax="100">

<span class="sr-only">20% Complete</span>

</div>

</div>

</a>

</li>

<!-- end task item -->

</ul>

</li>

<li class="footer">

<a href="#">View all tasks</a>

</li>

</ul>

</li>

<!-- User Account Menu -->

<li class="dropdown user user-menu">

<!-- Menu Toggle Button -->

<a href="#" class="dropdown-toggle" data-toggle="dropdown">

<!-- The user image in the navbar-->

<img src="dist/img/user2-160x160.jpg" class="user-image" alt="User Image"/>

<!-- hidden-xs hides the username on small devices so only the image appears. -->

<span class="hidden-xs">Alexander Pierce</span>

</a>

<ul class="dropdown-menu">

<!-- The user image in the menu -->

<li class="user-header">

<img src="dist/img/user2-160x160.jpg" class="img-circle" alt="User Image"/>

<p>

Alexander Pierce - Web Developer

<small>Member since Nov. 2012</small>

</p>

</li>

<!-- Menu Body -->

<li class="user-body">

<div class="row">

<div class="col-xs-4 text-center">

<a href="#">Followers</a>

</div>

<div class="col-xs-4 text-center">

<a href="#">Sales</a>

</div>

<div class="col-xs-4 text-center">

<a href="#">Friends</a>

</div>

</div>

<!-- /.row -->

</li>

<!-- Menu Footer-->

<li class="user-footer">

<div class="pull-left">

<a href="#" class="btn btn-default btn-flat">Profile</a>

</div>

<div class="pull-right">

<a href="#" class="btn btn-default btn-flat">Sign out</a>

</div>

</li>

</ul>

</li>

<!-- Control Sidebar Toggle Button -->

<li>

<a href="#" data-toggle="control-sidebar"><i class="fa fa-gears"></i></a>

</li>

</ul>

</div>

</nav>

</header>

<!-- Left side column. contains the logo and sidebar -->

<aside class="main-sidebar">

<!-- sidebar: style can be found in sidebar.less -->

<section class="sidebar">

<!-- Sidebar user panel (optional) -->

<div class="user-panel">

<div class="pull-left image">

<img src="dist/img/user2-160x160.jpg" class="img-circle" alt="User Image"/>

</div>

<div class="pull-left info">

<p>Alexander Pierce</p>

<!-- Status -->

<a href="#"><i class="fa fa-circle text-success"></i> Online</a>

</div>

</div>

<!-- search form (Optional) -->

<form action="#" method="get" class="sidebar-form">

<div class="input-group">

<input type="text" name="q" class="form-control" placeholder="Search..."/>

<span class="input-group-btn">

<button type="submit" name="search" id="search-btn" class="btn btn-flat"><i class="fa fa-search"></i>

</button>

</span>

</div>

</form>

<!-- /.search form -->

<!-- Sidebar Menu -->

<ul class="sidebar-menu">

<li class="header">HEADER</li>

<!-- Optionally, you can add icons to the links -->

<li class="active"><a href="#"><i class="fa fa-link"></i> <span>Link</span></a></li>

<li><a href="#"><i class="fa fa-link"></i> <span>Another Link</span></a></li>

<li class="treeview">

<a href="#"><i class="fa fa-link"></i> <span>Multilevel</span>

<span class="pull-right-container">

<i class="fa fa-angle-left pull-right"></i>

</span>

</a>

<ul class="treeview-menu">

<li><a href="#">Link in level 2</a></li>

<li><a href="#">Link in level 2</a></li>

</ul>

</li>

</ul>

<!-- /.sidebar-menu -->

</section>

<!-- /.sidebar -->

</aside>

<!-- Content Wrapper. Contains page content -->

<div class="content-wrapper">

<!-- Content Header (Page header) -->

<section class="content-header">

<h1>

Page Header

<small>Optional description</small>

</h1>

<ol class="breadcrumb">

<li><a href="#"><i class="fa fa-dashboard"></i> Level</a></li>

<li class="active">Here</li>

</ol>

</section>

<!-- Main content -->

<section class="content">

<!-- Your Page Content Here -->

</section>

<!-- /.content -->

</div>

<!-- /.content-wrapper -->

<!-- Main Footer -->

<footer class="main-footer">

<!-- To the right -->

<div class="pull-right hidden-xs">

Anything you want

</div>

<!-- Default to the left -->

<strong>Copyright &copy; 2016 <a href="#">Company</a>.</strong> All rights reserved.

</footer>

<!-- Control Sidebar -->

<aside class="control-sidebar control-sidebar-dark">

<!-- Create the tabs -->

<ul class="nav nav-tabs nav-justified control-sidebar-tabs">

<li class="active"><a href="#control-sidebar-home-tab" data-toggle="tab"><i class="fa fa-home"></i></a></li>

<li><a href="#control-sidebar-settings-tab" data-toggle="tab"><i class="fa fa-gears"></i></a></li>

</ul>

<!-- Tab panes -->

<div class="tab-content">

<!-- Home tab content -->

<div class="tab-pane active" id="control-sidebar-home-tab">

<h3 class="control-sidebar-heading">Recent Activity</h3>

<ul class="control-sidebar-menu">

<li>

<a href="javascript:;">

<i class="menu-icon fa fa-birthday-cake bg-red"></i>

<div class="menu-info">

<h4 class="control-sidebar-subheading">Langdon's Birthday</h4>

<p>Will be 23 on April 24th</p>

</div>

</a>

</li>

</ul>

<!-- /.control-sidebar-menu -->

<h3 class="control-sidebar-heading">Tasks Progress</h3>

<ul class="control-sidebar-menu">

<li>

<a href="javascript:;">

<h4 class="control-sidebar-subheading">

Custom Template Design

<span class="pull-right-container">

<span class="label label-danger pull-right">70%</span>

</span>

</h4>

<div class="progress progress-xxs">

<div class="progress-bar progress-bar-danger" style="width: 70%"></div>

</div>

</a>

</li>

</ul>

<!-- /.control-sidebar-menu -->

</div>

<!-- /.tab-pane -->

<!-- Stats tab content -->

<div class="tab-pane" id="control-sidebar-stats-tab">Stats Tab Content</div>

<!-- /.tab-pane -->

<!-- Settings tab content -->

<div class="tab-pane" id="control-sidebar-settings-tab">

<form method="post">

<h3 class="control-sidebar-heading">General Settings</h3>

<div class="form-group">

<label class="control-sidebar-subheading">

Report panel usage

<input type="checkbox" class="pull-right" checked="checked"/>

</label>

<p>

Some information about this general settings option

</p>

</div>

<!-- /.form-group -->

</form>

</div>

<!-- /.tab-pane -->

</div>

</aside>

<!-- /.control-sidebar -->

<!-- Add the sidebar's background. This div must be placed

immediately after the control sidebar -->

<div class="control-sidebar-bg"></div>

</div>

<!-- ./wrapper -->

<!-- REQUIRED JS SCRIPTS -->

<!-- jQuery 2.2.3 -->

<script src="plugins/jQuery/jquery-2.2.3.min.js"></script>

<!-- Bootstrap 3.3.6 -->

<script src="bootstrap/js/bootstrap.min.js"></script>

<!-- AdminLTE App -->

<script src="dist/js/app.min.js"></script>

<!-- Optionally, you can add Slimscroll and FastClick plugins.

Both of these plugins are recommended to enhance the

user experience. Slimscroll is required when using the

fixed layout. -->

</body>

</html>

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<title>AdminLTE 2 | Dashboard</title>

<!-- Tell the browser to be responsive to screen width -->

<meta content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no" name="viewport">

<!-- Bootstrap 3.3.6 -->

<link rel="stylesheet" href="bootstrap/css/bootstrap.min.css">

<!-- Font Awesome -->

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.5.0/css/font-awesome.min.css">

<!-- Ionicons -->

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/ionicons/2.0.1/css/ionicons.min.css">

<!-- Theme style -->

<link rel="stylesheet" href="dist/css/AdminLTE.min.css">

<!-- AdminLTE Skins. Choose a skin from the css/skins

folder instead of downloading all of them to reduce the load. -->

<link rel="stylesheet" href="dist/css/skins/\_all-skins.min.css">

<!-- iCheck -->

<link rel="stylesheet" href="plugins/iCheck/flat/blue.css">

<!-- Morris chart -->

<link rel="stylesheet" href="plugins/morris/morris.css">

<!-- jvectormap -->

<link rel="stylesheet" href="plugins/jvectormap/jquery-jvectormap-1.2.2.css">

<!-- Date Picker -->

<link rel="stylesheet" href="plugins/datepicker/datepicker3.css">

<!-- Daterange picker -->

<link rel="stylesheet" href="plugins/daterangepicker/daterangepicker.css">

<!-- bootstrap wysihtml5 - text editor -->

<link rel="stylesheet" href="plugins/bootstrap-wysihtml5/bootstrap3-wysihtml5.min.css">

<!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media queries -->

<!-- WARNING: Respond.js doesn't work if you view the page via file:// -->

<!--[if lt IE 9]>

<script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>

<script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>

<![endif]-->

</head>

<body class="hold-transition skin-blue sidebar-mini">

<div class="wrapper">

<header class="main-header">

<!-- Logo -->

<a href="index2.html" class="logo">

<!-- mini logo for sidebar mini 50x50 pixels -->

<span class="logo-mini"><b>A</b>LT</span>

<!-- logo for regular state and mobile devices -->

<span class="logo-lg"><b>Admin</b>LTE</span>

</a>

<!-- Header Navbar: style can be found in header.less -->

<nav class="navbar navbar-static-top">

<!-- Sidebar toggle button-->

<a href="#" class="sidebar-toggle" data-toggle="offcanvas" role="button">

<span class="sr-only">Toggle navigation</span>

</a>

<div class="navbar-custom-menu">

<ul class="nav navbar-nav">

<!-- Messages: style can be found in dropdown.less-->

<li class="dropdown messages-menu">

<a href="#" class="dropdown-toggle" data-toggle="dropdown">

<i class="fa fa-envelope-o"></i>

<span class="label label-success">4</span>

</a>

<ul class="dropdown-menu">

<li class="header">You have 4 messages</li>

<li>

<!-- inner menu: contains the actual data -->

<ul class="menu">

<li><!-- start message -->

<a href="#">

<div class="pull-left">

<img src="dist/img/user2-160x160.jpg" class="img-circle" alt="User Image">

</div>

<h4>

Support Team

<small><i class="fa fa-clock-o"></i> 5 mins</small>

</h4>

<p>Why not buy a new awesome theme?</p>

</a>

</li>

<!-- end message -->

<li>

<a href="#">

<div class="pull-left">

<img src="dist/img/user3-128x128.jpg" class="img-circle" alt="User Image">

</div>

<h4>

AdminLTE Design Team

<small><i class="fa fa-clock-o"></i> 2 hours</small>

</h4>

<p>Why not buy a new awesome theme?</p>

</a>

</li>

<li>

<a href="#">

<div class="pull-left">

<img src="dist/img/user4-128x128.jpg" class="img-circle" alt="User Image">

</div>

<h4>

Developers

<small><i class="fa fa-clock-o"></i> Today</small>

</h4>

<p>Why not buy a new awesome theme?</p>

</a>

</li>

<li>

<a href="#">

<div class="pull-left">

<img src="dist/img/user3-128x128.jpg" class="img-circle" alt="User Image">

</div>

<h4>

Sales Department

<small><i class="fa fa-clock-o"></i> Yesterday</small>

</h4>

<p>Why not buy a new awesome theme?</p>

</a>

</li>

<li>

<a href="#">

<div class="pull-left">

<img src="dist/img/user4-128x128.jpg" class="img-circle" alt="User Image">

</div>

<h4>

Reviewers

<small><i class="fa fa-clock-o"></i> 2 days</small>

</h4>

<p>Why not buy a new awesome theme?</p>

</a>

</li>

</ul>

</li>

<li class="footer"><a href="#">See All Messages</a></li>

</ul>

</li>

<!-- Notifications: style can be found in dropdown.less -->

<li class="dropdown notifications-menu">

<a href="#" class="dropdown-toggle" data-toggle="dropdown">

<i class="fa fa-bell-o"></i>

<span class="label label-warning">10</span>

</a>

<ul class="dropdown-menu">

<li class="header">You have 10 notifications</li>

<li>

<!-- inner menu: contains the actual data -->

<ul class="menu">

<li>

<a href="#">

<i class="fa fa-users text-aqua"></i> 5 new members joined today

</a>

</li>

<li>

<a href="#">

<i class="fa fa-warning text-yellow"></i> Very long description here that may not fit into the

page and may cause design problems

</a>

</li>

<li>

<a href="#">

<i class="fa fa-users text-red"></i> 5 new members joined

</a>

</li>

<li>

<a href="#">

<i class="fa fa-shopping-cart text-green"></i> 25 sales made

</a>

</li>

<li>

<a href="#">

<i class="fa fa-user text-red"></i> You changed your username

</a>

</li>

</ul>

</li>

<li class="footer"><a href="#">View all</a></li>

</ul>

</li>

<!-- Tasks: style can be found in dropdown.less -->

<li class="dropdown tasks-menu">

<a href="#" class="dropdown-toggle" data-toggle="dropdown">

<i class="fa fa-flag-o"></i>

<span class="label label-danger">9</span>

</a>

<ul class="dropdown-menu">

<li class="header">You have 9 tasks</li>

<li>

<!-- inner menu: contains the actual data -->

<ul class="menu">

<li><!-- Task item -->

<a href="#">

<h3>

Design some buttons

<small class="pull-right">20%</small>

</h3>

<div class="progress xs">

<div class="progress-bar progress-bar-aqua" style="width: 20%" role="progressbar" aria-valuenow="20" aria-valuemin="0" aria-valuemax="100">

<span class="sr-only">20% Complete</span>

</div>

</div>

</a>

</li>

<!-- end task item -->

<li><!-- Task item -->

<a href="#">

<h3>

Create a nice theme

<small class="pull-right">40%</small>

</h3>

<div class="progress xs">

<div class="progress-bar progress-bar-green" style="width: 40%" role="progressbar" aria-valuenow="20" aria-valuemin="0" aria-valuemax="100">

<span class="sr-only">40% Complete</span>

</div>

</div>

</a>

</li>

<!-- end task item -->

<li><!-- Task item -->

<a href="#">

<h3>

Some task I need to do

<small class="pull-right">60%</small>

</h3>

<div class="progress xs">

<div class="progress-bar progress-bar-red" style="width: 60%" role="progressbar" aria-valuenow="20" aria-valuemin="0" aria-valuemax="100">

<span class="sr-only">60% Complete</span>

</div>

</div>

</a>

</li>

<!-- end task item -->

<li><!-- Task item -->

<a href="#">

<h3>

Make beautiful transitions

<small class="pull-right">80%</small>

</h3>

<div class="progress xs">

<div class="progress-bar progress-bar-yellow" style="width: 80%" role="progressbar" aria-valuenow="20" aria-valuemin="0" aria-valuemax="100">

<span class="sr-only">80% Complete</span>

</div>

</div>

</a>

</li>

<!-- end task item -->

</ul>

</li>

<li class="footer">

<a href="#">View all tasks</a>

</li>

</ul>

</li>

<!-- User Account: style can be found in dropdown.less -->

<li class="dropdown user user-menu">

<a href="#" class="dropdown-toggle" data-toggle="dropdown">

<img src="dist/img/user2-160x160.jpg" class="user-image" alt="User Image">

<span class="hidden-xs">Alexander Pierce</span>

</a>

<ul class="dropdown-menu">

<!-- User image -->

<li class="user-header">

<img src="dist/img/user2-160x160.jpg" class="img-circle" alt="User Image">

<p>

Alexander Pierce - Web Developer

<small>Member since Nov. 2012</small>

</p>

</li>

<!-- Menu Body -->

<li class="user-body">

<div class="row">

<div class="col-xs-4 text-center">

<a href="#">Followers</a>

</div>

<div class="col-xs-4 text-center">

<a href="#">Sales</a>

</div>

<div class="col-xs-4 text-center">

<a href="#">Friends</a>

</div>

</div>

<!-- /.row -->

</li>

<!-- Menu Footer-->

<li class="user-footer">

<div class="pull-left">

<a href="#" class="btn btn-default btn-flat">Profile</a>

</div>

<div class="pull-right">

<a href="#" class="btn btn-default btn-flat">Sign out</a>

</div>

</li>

</ul>

</li>

<!-- Control Sidebar Toggle Button -->

<li>

<a href="#" data-toggle="control-sidebar"><i class="fa fa-gears"></i></a>

</li>

</ul>

</div>

</nav>

</header>

<!-- Left side column. contains the logo and sidebar -->

<aside class="main-sidebar">

<!-- sidebar: style can be found in sidebar.less -->

<section class="sidebar">

<!-- Sidebar user panel -->

<div class="user-panel">

<div class="pull-left image">

<img src="dist/img/user2-160x160.jpg" class="img-circle" alt="User Image">

</div>

<div class="pull-left info">

<p>Alexander Pierce</p>

<a href="#"><i class="fa fa-circle text-success"></i> Online</a>

</div>

</div>

<!-- search form -->

<form action="#" method="get" class="sidebar-form">

<div class="input-group">

<input type="text" name="q" class="form-control" placeholder="Search...">

<span class="input-group-btn">

<button type="submit" name="search" id="search-btn" class="btn btn-flat"><i class="fa fa-search"></i>

</button>

</span>

</div>

</form>

<!-- /.search form -->

<!-- sidebar menu: : style can be found in sidebar.less -->

<ul class="sidebar-menu">

<li class="header">MAIN NAVIGATION</li>

<li class="active treeview">

<a href="#">

<i class="fa fa-dashboard"></i> <span>Dashboard</span>

<span class="pull-right-container">

<i class="fa fa-angle-left pull-right"></i>

</span>

</a>

<ul class="treeview-menu">

<li class="active"><a href="index.html"><i class="fa fa-circle-o"></i> Dashboard v1</a></li>

<li><a href="index2.html"><i class="fa fa-circle-o"></i> Dashboard v2</a></li>

</ul>

</li>

<li class="treeview">

<a href="#">

<i class="fa fa-files-o"></i>

<span>Layout Options</span>

<span class="pull-right-container">

<span class="label label-primary pull-right">4</span>

</span>

</a>

<ul class="treeview-menu">

<li><a href="pages/layout/top-nav.html"><i class="fa fa-circle-o"></i> Top Navigation</a></li>

<li><a href="pages/layout/boxed.html"><i class="fa fa-circle-o"></i> Boxed</a></li>

<li><a href="pages/layout/fixed.html"><i class="fa fa-circle-o"></i> Fixed</a></li>

<li><a href="pages/layout/collapsed-sidebar.html"><i class="fa fa-circle-o"></i> Collapsed Sidebar</a></li>

</ul>

</li>

<li>

<a href="pages/widgets.html">

<i class="fa fa-th"></i> <span>Widgets</span>

<span class="pull-right-container">

<small class="label pull-right bg-green">new</small>

</span>

</a>

</li>

<li class="treeview">

<a href="#">

<i class="fa fa-pie-chart"></i>

<span>Charts</span>

<span class="pull-right-container">

<i class="fa fa-angle-left pull-right"></i>

</span>

</a>

<ul class="treeview-menu">

<li><a href="pages/charts/chartjs.html"><i class="fa fa-circle-o"></i> ChartJS</a></li>

<li><a href="pages/charts/morris.html"><i class="fa fa-circle-o"></i> Morris</a></li>

<li><a href="pages/charts/flot.html"><i class="fa fa-circle-o"></i> Flot</a></li>

<li><a href="pages/charts/inline.html"><i class="fa fa-circle-o"></i> Inline charts</a></li>

</ul>

</li>

<li class="treeview">

<a href="#">

<i class="fa fa-laptop"></i>

<span>UI Elements</span>

<span class="pull-right-container">

<i class="fa fa-angle-left pull-right"></i>

</span>

</a>

<ul class="treeview-menu">

<li><a href="pages/UI/general.html"><i class="fa fa-circle-o"></i> General</a></li>

<li><a href="pages/UI/icons.html"><i class="fa fa-circle-o"></i> Icons</a></li>

<li><a href="pages/UI/buttons.html"><i class="fa fa-circle-o"></i> Buttons</a></li>

<li><a href="pages/UI/sliders.html"><i class="fa fa-circle-o"></i> Sliders</a></li>

<li><a href="pages/UI/timeline.html"><i class="fa fa-circle-o"></i> Timeline</a></li>

<li><a href="pages/UI/modals.html"><i class="fa fa-circle-o"></i> Modals</a></li>

</ul>

</li>

<li class="treeview">

<a href="#">

<i class="fa fa-edit"></i> <span>Forms</span>

<span class="pull-right-container">

<i class="fa fa-angle-left pull-right"></i>

</span>

</a>

<ul class="treeview-menu">

<li><a href="pages/forms/general.html"><i class="fa fa-circle-o"></i> General Elements</a></li>

<li><a href="pages/forms/advanced.html"><i class="fa fa-circle-o"></i> Advanced Elements</a></li>

<li><a href="pages/forms/editors.html"><i class="fa fa-circle-o"></i> Editors</a></li>

</ul>

</li>

<li class="treeview">

<a href="#">

<i class="fa fa-table"></i> <span>Tables</span>

<span class="pull-right-container">

<i class="fa fa-angle-left pull-right"></i>

</span>

</a>

<ul class="treeview-menu">

<li><a href="pages/tables/simple.html"><i class="fa fa-circle-o"></i> Simple tables</a></li>

<li><a href="pages/tables/data.html"><i class="fa fa-circle-o"></i> Data tables</a></li>

</ul>

</li>

<li>

<a href="pages/calendar.html">

<i class="fa fa-calendar"></i> <span>Calendar</span>

<span class="pull-right-container">

<small class="label pull-right bg-red">3</small>

<small class="label pull-right bg-blue">17</small>

</span>

</a>

</li>

<li>

<a href="pages/mailbox/mailbox.html">

<i class="fa fa-envelope"></i> <span>Mailbox</span>

<span class="pull-right-container">

<small class="label pull-right bg-yellow">12</small>

<small class="label pull-right bg-green">16</small>

<small class="label pull-right bg-red">5</small>

</span>

</a>

</li>

<li class="treeview">

<a href="#">

<i class="fa fa-folder"></i> <span>Examples</span>

<span class="pull-right-container">

<i class="fa fa-angle-left pull-right"></i>

</span>

</a>

<ul class="treeview-menu">

<li><a href="pages/examples/invoice.html"><i class="fa fa-circle-o"></i> Invoice</a></li>

<li><a href="pages/examples/profile.html"><i class="fa fa-circle-o"></i> Profile</a></li>

<li><a href="pages/examples/login.html"><i class="fa fa-circle-o"></i> Login</a></li>

<li><a href="pages/examples/register.html"><i class="fa fa-circle-o"></i> Register</a></li>

<li><a href="pages/examples/lockscreen.html"><i class="fa fa-circle-o"></i> Lockscreen</a></li>

<li><a href="pages/examples/404.html"><i class="fa fa-circle-o"></i> 404 Error</a></li>

<li><a href="pages/examples/500.html"><i class="fa fa-circle-o"></i> 500 Error</a></li>

<li><a href="pages/examples/blank.html"><i class="fa fa-circle-o"></i> Blank Page</a></li>

<li><a href="pages/examples/pace.html"><i class="fa fa-circle-o"></i> Pace Page</a></li>

</ul>

</li>

<li class="treeview">

<a href="#">

<i class="fa fa-share"></i> <span>Multilevel</span>

<span class="pull-right-container">

<i class="fa fa-angle-left pull-right"></i>

</span>

</a>

<ul class="treeview-menu">

<li><a href="#"><i class="fa fa-circle-o"></i> Level One</a></li>

<li>

<a href="#"><i class="fa fa-circle-o"></i> Level One

<span class="pull-right-container">

<i class="fa fa-angle-left pull-right"></i>

</span>

</a>

<ul class="treeview-menu">

<li><a href="#"><i class="fa fa-circle-o"></i> Level Two</a></li>

<li>

<a href="#"><i class="fa fa-circle-o"></i> Level Two

<span class="pull-right-container">

<i class="fa fa-angle-left pull-right"></i>

</span>

</a>

<ul class="treeview-menu">

<li><a href="#"><i class="fa fa-circle-o"></i> Level Three</a></li>

<li><a href="#"><i class="fa fa-circle-o"></i> Level Three</a></li>

</ul>

</li>

</ul>

</li>

<li><a href="#"><i class="fa fa-circle-o"></i> Level One</a></li>

</ul>

</li>

<li><a href="documentation/index.html"><i class="fa fa-book"></i> <span>Documentation</span></a></li>

<li class="header">LABELS</li>

<li><a href="#"><i class="fa fa-circle-o text-red"></i> <span>Important</span></a></li>

<li><a href="#"><i class="fa fa-circle-o text-yellow"></i> <span>Warning</span></a></li>

<li><a href="#"><i class="fa fa-circle-o text-aqua"></i> <span>Information</span></a></li>

</ul>

</section>

<!-- /.sidebar -->

</aside>

<!-- Content Wrapper. Contains page content -->

<div class="content-wrapper">

<!-- Content Header (Page header) -->

<section class="content-header">

<h1>

Dashboard

<small>Control panel</small>

</h1>

<ol class="breadcrumb">

<li><a href="#"><i class="fa fa-dashboard"></i> Home</a></li>

<li class="active">Dashboard</li>

</ol>

</section>

<!-- Main content -->

<section class="content">

<!-- Small boxes (Stat box) -->

<div class="row">

<div class="col-lg-3 col-xs-6">

<!-- small box -->

<div class="small-box bg-aqua">

<div class="inner">

<h3>150</h3>

<p>New Orders</p>

</div>

<div class="icon">

<i class="ion ion-bag"></i>

</div>

<a href="#" class="small-box-footer">More info <i class="fa fa-arrow-circle-right"></i></a>

</div>

</div>

<!-- ./col -->

<div class="col-lg-3 col-xs-6">

<!-- small box -->

<div class="small-box bg-green">

<div class="inner">

<h3>53<sup style="font-size: 20px">%</sup></h3>

<p>Bounce Rate</p>

</div>

<div class="icon">

<i class="ion ion-stats-bars"></i>

</div>

<a href="#" class="small-box-footer">More info <i class="fa fa-arrow-circle-right"></i></a>

</div>

</div>

<!-- ./col -->

<div class="col-lg-3 col-xs-6">

<!-- small box -->

<div class="small-box bg-yellow">

<div class="inner">

<h3>44</h3>

<p>User Registrations</p>

</div>

<div class="icon">

<i class="ion ion-person-add"></i>

</div>

<a href="#" class="small-box-footer">More info <i class="fa fa-arrow-circle-right"></i></a>

</div>

</div>

<!-- ./col -->

<div class="col-lg-3 col-xs-6">

<!-- small box -->

<div class="small-box bg-red">

<div class="inner">

<h3>65</h3>

<p>Unique Visitors</p>

</div>

<div class="icon">

<i class="ion ion-pie-graph"></i>

</div>

<a href="#" class="small-box-footer">More info <i class="fa fa-arrow-circle-right"></i></a>

</div>

</div>

<!-- ./col -->

</div>

<!-- /.row -->

<!-- Main row -->

<div class="row">

<!-- Left col -->

<section class="col-lg-7 connectedSortable">

<!-- Custom tabs (Charts with tabs)-->

<div class="nav-tabs-custom">

<!-- Tabs within a box -->

<ul class="nav nav-tabs pull-right">

<li class="active"><a href="#revenue-chart" data-toggle="tab">Area</a></li>

<li><a href="#sales-chart" data-toggle="tab">Donut</a></li>

<li class="pull-left header"><i class="fa fa-inbox"></i> Sales</li>

</ul>

<div class="tab-content no-padding">

<!-- Morris chart - Sales -->

<div class="chart tab-pane active" id="revenue-chart" style="position: relative; height: 300px;"></div>

<div class="chart tab-pane" id="sales-chart" style="position: relative; height: 300px;"></div>

</div>

</div>

<!-- /.nav-tabs-custom -->

<!-- Chat box -->

<div class="box box-success">

<div class="box-header">

<i class="fa fa-comments-o"></i>

<h3 class="box-title">Chat</h3>

<div class="box-tools pull-right" data-toggle="tooltip" title="Status">

<div class="btn-group" data-toggle="btn-toggle">

<button type="button" class="btn btn-default btn-sm active"><i class="fa fa-square text-green"></i>

</button>

<button type="button" class="btn btn-default btn-sm"><i class="fa fa-square text-red"></i></button>

</div>

</div>

</div>

<div class="box-body chat" id="chat-box">

<!-- chat item -->

<div class="item">

<img src="dist/img/user4-128x128.jpg" alt="user image" class="online">

<p class="message">

<a href="#" class="name">

<small class="text-muted pull-right"><i class="fa fa-clock-o"></i> 2:15</small>

Mike Doe

</a>

I would like to meet you to discuss the latest news about

the arrival of the new theme. They say it is going to be one the

best themes on the market

</p>

<div class="attachment">

<h4>Attachments:</h4>

<p class="filename">

Theme-thumbnail-image.jpg

</p>

<div class="pull-right">

<button type="button" class="btn btn-primary btn-sm btn-flat">Open</button>

</div>

</div>

<!-- /.attachment -->

</div>

<!-- /.item -->

<!-- chat item -->

<div class="item">

<img src="dist/img/user3-128x128.jpg" alt="user image" class="offline">

<p class="message">

<a href="#" class="name">

<small class="text-muted pull-right"><i class="fa fa-clock-o"></i> 5:15</small>

Alexander Pierce

</a>

I would like to meet you to discuss the latest news about

the arrival of the new theme. They say it is going to be one the

best themes on the market

</p>

</div>

<!-- /.item -->

<!-- chat item -->

<div class="item">

<img src="dist/img/user2-160x160.jpg" alt="user image" class="offline">

<p class="message">

<a href="#" class="name">

<small class="text-muted pull-right"><i class="fa fa-clock-o"></i> 5:30</small>

Susan Doe

</a>

I would like to meet you to discuss the latest news about

the arrival of the new theme. They say it is going to be one the

best themes on the market

</p>

</div>

<!-- /.item -->

</div>

<!-- /.chat -->

<div class="box-footer">

<div class="input-group">

<input class="form-control" placeholder="Type message...">

<div class="input-group-btn">

<button type="button" class="btn btn-success"><i class="fa fa-plus"></i></button>

</div>

</div>

</div>

</div>

<!-- /.box (chat box) -->

<!-- TO DO List -->

<div class="box box-primary">

<div class="box-header">

<i class="ion ion-clipboard"></i>

<h3 class="box-title">To Do List</h3>

<div class="box-tools pull-right">

<ul class="pagination pagination-sm inline">

<li><a href="#">&laquo;</a></li>

<li><a href="#">1</a></li>

<li><a href="#">2</a></li>

<li><a href="#">3</a></li>

<li><a href="#">&raquo;</a></li>

</ul>

</div>

</div>

<!-- /.box-header -->

<div class="box-body">

<ul class="todo-list">

<li>

<!-- drag handle -->

<span class="handle">

<i class="fa fa-ellipsis-v"></i>

<i class="fa fa-ellipsis-v"></i>

</span>

<!-- checkbox -->

<input type="checkbox" value="">

<!-- todo text -->

<span class="text">Design a nice theme</span>

<!-- Emphasis label -->

<small class="label label-danger"><i class="fa fa-clock-o"></i> 2 mins</small>

<!-- General tools such as edit or delete-->

<div class="tools">

<i class="fa fa-edit"></i>

<i class="fa fa-trash-o"></i>

</div>

</li>

<li>

<span class="handle">

<i class="fa fa-ellipsis-v"></i>

<i class="fa fa-ellipsis-v"></i>

</span>

<input type="checkbox" value="">

<span class="text">Make the theme responsive</span>

<small class="label label-info"><i class="fa fa-clock-o"></i> 4 hours</small>

<div class="tools">

<i class="fa fa-edit"></i>

<i class="fa fa-trash-o"></i>

</div>

</li>

<li>

<span class="handle">

<i class="fa fa-ellipsis-v"></i>

<i class="fa fa-ellipsis-v"></i>

</span>

<input type="checkbox" value="">

<span class="text">Let theme shine like a star</span>

<small class="label label-warning"><i class="fa fa-clock-o"></i> 1 day</small>

<div class="tools">

<i class="fa fa-edit"></i>

<i class="fa fa-trash-o"></i>

</div>

</li>

<li>

<span class="handle">

<i class="fa fa-ellipsis-v"></i>

<i class="fa fa-ellipsis-v"></i>

</span>

<input type="checkbox" value="">

<span class="text">Let theme shine like a star</span>

<small class="label label-success"><i class="fa fa-clock-o"></i> 3 days</small>

<div class="tools">

<i class="fa fa-edit"></i>

<i class="fa fa-trash-o"></i>

</div>

</li>

<li>

<span class="handle">

<i class="fa fa-ellipsis-v"></i>

<i class="fa fa-ellipsis-v"></i>

</span>

<input type="checkbox" value="">

<span class="text">Check your messages and notifications</span>

<small class="label label-primary"><i class="fa fa-clock-o"></i> 1 week</small>

<div class="tools">

<i class="fa fa-edit"></i>

<i class="fa fa-trash-o"></i>

</div>

</li>

<li>

<span class="handle">

<i class="fa fa-ellipsis-v"></i>

<i class="fa fa-ellipsis-v"></i>

</span>

<input type="checkbox" value="">

<span class="text">Let theme shine like a star</span>

<small class="label label-default"><i class="fa fa-clock-o"></i> 1 month</small>

<div class="tools">

<i class="fa fa-edit"></i>

<i class="fa fa-trash-o"></i>

</div>

</li>

</ul>

</div>

<!-- /.box-body -->

<div class="box-footer clearfix no-border">

<button type="button" class="btn btn-default pull-right"><i class="fa fa-plus"></i> Add item</button>

</div>

</div>

<!-- /.box -->

<!-- quick email widget -->

<div class="box box-info">

<div class="box-header">

<i class="fa fa-envelope"></i>

<h3 class="box-title">Quick Email</h3>

<!-- tools box -->

<div class="pull-right box-tools">

<button type="button" class="btn btn-info btn-sm" data-widget="remove" data-toggle="tooltip" title="Remove">

<i class="fa fa-times"></i></button>

</div>

<!-- /. tools -->

</div>

<div class="box-body">

<form action="#" method="post">

<div class="form-group">

<input type="email" class="form-control" name="emailto" placeholder="Email to:">

</div>

<div class="form-group">

<input type="text" class="form-control" name="subject" placeholder="Subject">

</div>

<div>

<textarea class="textarea" placeholder="Message" style="width: 100%; height: 125px; font-size: 14px; line-height: 18px; border: 1px solid #dddddd; padding: 10px;"></textarea>

</div>

</form>

</div>

<div class="box-footer clearfix">

<button type="button" class="pull-right btn btn-default" id="sendEmail">Send

<i class="fa fa-arrow-circle-right"></i></button>

</div>

</div>

</section>

<!-- /.Left col -->

<!-- right col (We are only adding the ID to make the widgets sortable)-->

<section class="col-lg-5 connectedSortable">

<!-- Map box -->

<div class="box box-solid bg-light-blue-gradient">

<div class="box-header">

<!-- tools box -->

<div class="pull-right box-tools">

<button type="button" class="btn btn-primary btn-sm daterange pull-right" data-toggle="tooltip" title="Date range">

<i class="fa fa-calendar"></i></button>

<button type="button" class="btn btn-primary btn-sm pull-right" data-widget="collapse" data-toggle="tooltip" title="Collapse" style="margin-right: 5px;">

<i class="fa fa-minus"></i></button>

</div>

<!-- /. tools -->

<i class="fa fa-map-marker"></i>

<h3 class="box-title">

Visitors

</h3>

</div>

<div class="box-body">

<div id="world-map" style="height: 250px; width: 100%;"></div>

</div>

<!-- /.box-body-->

<div class="box-footer no-border">

<div class="row">

<div class="col-xs-4 text-center" style="border-right: 1px solid #f4f4f4">

<div id="sparkline-1"></div>

<div class="knob-label">Visitors</div>

</div>

<!-- ./col -->

<div class="col-xs-4 text-center" style="border-right: 1px solid #f4f4f4">

<div id="sparkline-2"></div>

<div class="knob-label">Online</div>

</div>

<!-- ./col -->

<div class="col-xs-4 text-center">

<div id="sparkline-3"></div>

<div class="knob-label">Exists</div>

</div>

<!-- ./col -->

</div>

<!-- /.row -->

</div>

</div>

<!-- /.box -->

<!-- solid sales graph -->

<div class="box box-solid bg-teal-gradient">

<div class="box-header">

<i class="fa fa-th"></i>

<h3 class="box-title">Sales Graph</h3>

<div class="box-tools pull-right">

<button type="button" class="btn bg-teal btn-sm" data-widget="collapse"><i class="fa fa-minus"></i>

</button>

<button type="button" class="btn bg-teal btn-sm" data-widget="remove"><i class="fa fa-times"></i>

</button>

</div>

</div>

<div class="box-body border-radius-none">

<div class="chart" id="line-chart" style="height: 250px;"></div>

</div>

<!-- /.box-body -->

<div class="box-footer no-border">

<div class="row">

<div class="col-xs-4 text-center" style="border-right: 1px solid #f4f4f4">

<input type="text" class="knob" data-readonly="true" value="20" data-width="60" data-height="60" data-fgColor="#39CCCC">

<div class="knob-label">Mail-Orders</div>

</div>

<!-- ./col -->

<div class="col-xs-4 text-center" style="border-right: 1px solid #f4f4f4">

<input type="text" class="knob" data-readonly="true" value="50" data-width="60" data-height="60" data-fgColor="#39CCCC">

<div class="knob-label">Online</div>

</div>

<!-- ./col -->

<div class="col-xs-4 text-center">

<input type="text" class="knob" data-readonly="true" value="30" data-width="60" data-height="60" data-fgColor="#39CCCC">

<div class="knob-label">In-Store</div>

</div>

<!-- ./col -->

</div>

<!-- /.row -->

</div>

<!-- /.box-footer -->

</div>

<!-- /.box -->

<!-- Calendar -->

<div class="box box-solid bg-green-gradient">

<div class="box-header">

<i class="fa fa-calendar"></i>

<h3 class="box-title">Calendar</h3>

<!-- tools box -->

<div class="pull-right box-tools">

<!-- button with a dropdown -->

<div class="btn-group">

<button type="button" class="btn btn-success btn-sm dropdown-toggle" data-toggle="dropdown">

<i class="fa fa-bars"></i></button>

<ul class="dropdown-menu pull-right" role="menu">

<li><a href="#">Add new event</a></li>

<li><a href="#">Clear events</a></li>

<li class="divider"></li>

<li><a href="#">View calendar</a></li>

</ul>

</div>

<button type="button" class="btn btn-success btn-sm" data-widget="collapse"><i class="fa fa-minus"></i>

</button>

<button type="button" class="btn btn-success btn-sm" data-widget="remove"><i class="fa fa-times"></i>

</button>

</div>

<!-- /. tools -->

</div>

<!-- /.box-header -->

<div class="box-body no-padding">

<!--The calendar -->

<div id="calendar" style="width: 100%"></div>

</div>

<!-- /.box-body -->

<div class="box-footer text-black">

<div class="row">

<div class="col-sm-6">

<!-- Progress bars -->

<div class="clearfix">

<span class="pull-left">Task #1</span>

<small class="pull-right">90%</small>

</div>

<div class="progress xs">

<div class="progress-bar progress-bar-green" style="width: 90%;"></div>

</div>

<div class="clearfix">

<span class="pull-left">Task #2</span>

<small class="pull-right">70%</small>

</div>

<div class="progress xs">

<div class="progress-bar progress-bar-green" style="width: 70%;"></div>

</div>

</div>

<!-- /.col -->

<div class="col-sm-6">

<div class="clearfix">

<span class="pull-left">Task #3</span>

<small class="pull-right">60%</small>

</div>

<div class="progress xs">

<div class="progress-bar progress-bar-green" style="width: 60%;"></div>

</div>

<div class="clearfix">

<span class="pull-left">Task #4</span>

<small class="pull-right">40%</small>

</div>

<div class="progress xs">

<div class="progress-bar progress-bar-green" style="width: 40%;"></div>

</div>

</div>

<!-- /.col -->

</div>

<!-- /.row -->

</div>

</div>

<!-- /.box -->

</section>

<!-- right col -->

</div>

<!-- /.row (main row) -->

</section>

<!-- /.content -->

</div>

<!-- /.content-wrapper -->

<footer class="main-footer">

<div class="pull-right hidden-xs">

<b>Version</b> 2.3.8

</div>

<strong>Copyright &copy; 2014-2016 <a href="http://almsaeedstudio.com">Almsaeed Studio</a>.</strong> All rights

reserved.

</footer>

<!-- Control Sidebar -->

<aside class="control-sidebar control-sidebar-dark">

<!-- Create the tabs -->

<ul class="nav nav-tabs nav-justified control-sidebar-tabs">

<li><a href="#control-sidebar-home-tab" data-toggle="tab"><i class="fa fa-home"></i></a></li>

<li><a href="#control-sidebar-settings-tab" data-toggle="tab"><i class="fa fa-gears"></i></a></li>

</ul>

<!-- Tab panes -->

<div class="tab-content">

<!-- Home tab content -->

<div class="tab-pane" id="control-sidebar-home-tab">

<h3 class="control-sidebar-heading">Recent Activity</h3>

<ul class="control-sidebar-menu">

<li>

<a href="javascript:void(0)">

<i class="menu-icon fa fa-birthday-cake bg-red"></i>

<div class="menu-info">

<h4 class="control-sidebar-subheading">Langdon's Birthday</h4>

<p>Will be 23 on April 24th</p>

</div>

</a>

</li>

<li>

<a href="javascript:void(0)">

<i class="menu-icon fa fa-user bg-yellow"></i>

<div class="menu-info">

<h4 class="control-sidebar-subheading">Frodo Updated His Profile</h4>

<p>New phone +1(800)555-1234</p>

</div>

</a>

</li>

<li>

<a href="javascript:void(0)">

<i class="menu-icon fa fa-envelope-o bg-light-blue"></i>

<div class="menu-info">

<h4 class="control-sidebar-subheading">Nora Joined Mailing List</h4>

<p>nora@example.com</p>

</div>

</a>

</li>

<li>

<a href="javascript:void(0)">

<i class="menu-icon fa fa-file-code-o bg-green"></i>

<div class="menu-info">

<h4 class="control-sidebar-subheading">Cron Job 254 Executed</h4>

<p>Execution time 5 seconds</p>

</div>

</a>

</li>

</ul>

<!-- /.control-sidebar-menu -->

<h3 class="control-sidebar-heading">Tasks Progress</h3>

<ul class="control-sidebar-menu">

<li>

<a href="javascript:void(0)">

<h4 class="control-sidebar-subheading">

Custom Template Design

<span class="label label-danger pull-right">70%</span>

</h4>

<div class="progress progress-xxs">

<div class="progress-bar progress-bar-danger" style="width: 70%"></div>

</div>

</a>

</li>

<li>

<a href="javascript:void(0)">

<h4 class="control-sidebar-subheading">

Update Resume

<span class="label label-success pull-right">95%</span>

</h4>

<div class="progress progress-xxs">

<div class="progress-bar progress-bar-success" style="width: 95%"></div>

</div>

</a>

</li>

<li>

<a href="javascript:void(0)">

<h4 class="control-sidebar-subheading">

Laravel Integration

<span class="label label-warning pull-right">50%</span>

</h4>

<div class="progress progress-xxs">

<div class="progress-bar progress-bar-warning" style="width: 50%"></div>

</div>

</a>

</li>

<li>

<a href="javascript:void(0)">

<h4 class="control-sidebar-subheading">

Back End Framework

<span class="label label-primary pull-right">68%</span>

</h4>

<div class="progress progress-xxs">

<div class="progress-bar progress-bar-primary" style="width: 68%"></div>

</div>

</a>

</li>

</ul>

<!-- /.control-sidebar-menu -->

</div>

<!-- /.tab-pane -->

<!-- Stats tab content -->

<div class="tab-pane" id="control-sidebar-stats-tab">Stats Tab Content</div>

<!-- /.tab-pane -->

<!-- Settings tab content -->

<div class="tab-pane" id="control-sidebar-settings-tab">

<form method="post">

<h3 class="control-sidebar-heading">General Settings</h3>

<div class="form-group">

<label class="control-sidebar-subheading">

Report panel usage

<input type="checkbox" class="pull-right" checked>

</label>

<p>

Some information about this general settings option

</p>

</div>

<!-- /.form-group -->

<div class="form-group">

<label class="control-sidebar-subheading">

Allow mail redirect

<input type="checkbox" class="pull-right" checked>

</label>

<p>

Other sets of options are available

</p>

</div>

<!-- /.form-group -->

<div class="form-group">

<label class="control-sidebar-subheading">

Expose author name in posts

<input type="checkbox" class="pull-right" checked>

</label>

<p>

Allow the user to show his name in blog posts

</p>

</div>

<!-- /.form-group -->

<h3 class="control-sidebar-heading">Chat Settings</h3>

<div class="form-group">

<label class="control-sidebar-subheading">

Show me as online

<input type="checkbox" class="pull-right" checked>

</label>

</div>

<!-- /.form-group -->

<div class="form-group">

<label class="control-sidebar-subheading">

Turn off notifications

<input type="checkbox" class="pull-right">

</label>

</div>

<!-- /.form-group -->

<div class="form-group">

<label class="control-sidebar-subheading">

Delete chat history

<a href="javascript:void(0)" class="text-red pull-right"><i class="fa fa-trash-o"></i></a>

</label>

</div>

<!-- /.form-group -->

</form>

</div>

<!-- /.tab-pane -->

</div>

</aside>

<!-- /.control-sidebar -->

<!-- Add the sidebar's background. This div must be placed

immediately after the control sidebar -->

<div class="control-sidebar-bg"></div>

</div>

<!-- ./wrapper -->

<!-- jQuery 2.2.3 -->

<script src="plugins/jQuery/jquery-2.2.3.min.js"></script>

<!-- jQuery UI 1.11.4 -->

<script src="https://code.jquery.com/ui/1.11.4/jquery-ui.min.js"></script>

<!-- Resolve conflict in jQuery UI tooltip with Bootstrap tooltip -->

<script>

$.widget.bridge('uibutton', $.ui.button);

</script>

<!-- Bootstrap 3.3.6 -->

<script src="bootstrap/js/bootstrap.min.js"></script>

<!-- Morris.js charts -->

<script src="https://cdnjs.cloudflare.com/ajax/libs/raphael/2.1.0/raphael-min.js"></script>

<script src="plugins/morris/morris.min.js"></script>

<!-- Sparkline -->

<script src="plugins/sparkline/jquery.sparkline.min.js"></script>

<!-- jvectormap -->

<script src="plugins/jvectormap/jquery-jvectormap-1.2.2.min.js"></script>

<script src="plugins/jvectormap/jquery-jvectormap-world-mill-en.js"></script>

<!-- jQuery Knob Chart -->

<script src="plugins/knob/jquery.knob.js"></script>

<!-- daterangepicker -->

<script src="https://cdnjs.cloudflare.com/ajax/libs/moment.js/2.11.2/moment.min.js"></script>

<script src="plugins/daterangepicker/daterangepicker.js"></script>

<!-- datepicker -->

<script src="plugins/datepicker/bootstrap-datepicker.js"></script>

<!-- Bootstrap WYSIHTML5 -->

<script src="plugins/bootstrap-wysihtml5/bootstrap3-wysihtml5.all.min.js"></script>

<!-- Slimscroll -->

<script src="plugins/slimScroll/jquery.slimscroll.min.js"></script>

<!-- FastClick -->

<script src="plugins/fastclick/fastclick.js"></script>

<!-- AdminLTE App -->

<script src="dist/js/app.min.js"></script>

<!-- AdminLTE dashboard demo (This is only for demo purposes) -->

<script src="dist/js/pages/dashboard.js"></script>

<!-- AdminLTE for demo purposes -->

<script src="dist/js/demo.js"></script>

</body>

</html>