Source code of Fly Away-(An airline booking portal)

-Jyothi N.M

Git Link

<https://github.com/jyothiarunkr/phase2-project2-flyaway-jyothinm.git>

|  |
| --- |
| package in.co.air.line.ticket.controller; |
|  |  |
|  | import java.io.IOException; |
|  |  |
|  |  |
|  | import javax.servlet.ServletException; |
|  | import javax.servlet.annotation.WebServlet; |
|  | import javax.servlet.http.HttpServlet; |
|  | import javax.servlet.http.HttpServletRequest; |
|  | import javax.servlet.http.HttpServletResponse; |
|  |  |
|  | import org.apache.log4j.Logger; |
|  | import org.apache.log4j.net.SyslogAppender; |
|  |  |
|  | import in.co.air.line.ticket.util.ServletUtility; |
|  |  |
|  |  |
|  | /\*\* |
|  | \* |
|  | \* |
|  | \* Servlet implementation class WelcomeCtl |
|  | \* |
|  | \* |
|  | \* |
|  | \* \* @author Navigable Set |
|  | \* @version 1.0 |
|  | \* @Copyright (c) Navigable Set |
|  | \*/ |
|  |  |
|  |  |
|  | @WebServlet(name = "WelcomeCtl", urlPatterns = { "/WelcomeCtl" }) |
|  | public class WelcomeCtl extends BaseCtl { |
|  |  |
|  | private static final long serialVersionUID = 1L; |
|  |  |
|  |  |
|  | /\*\* |
|  | \* @see HttpServlet#HttpServlet() |
|  | \*/ |
|  | /\*public WelcomeCtl() { |
|  | super(); |
|  | // TODO Auto-generated constructor stub |
|  | }\*/ |
|  |  |
|  | /\*\* |
|  | \* Contains display logic |
|  | \*/ |
|  | /\*\* |
|  | \* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response) |
|  | \*/ |
|  | protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException { |
|  | ServletUtility.forward(ATBView.WELCOME\_VIEW, request, response); |
|  |  |
|  | } |
|  | /\*\* |
|  | \* Returns the VIEW page of this Controller |
|  | \* |
|  | \* @return |
|  | \*/ |
|  | @Override |
|  | protected String getView() { |
|  | // TODO Auto-generated method stub |
|  | return ATBView.WELCOME\_VIEW; |
|  | } |
|  |  |
|  | } |

|  |
| --- |
| \*\* |
|  | \* Servlet implementation class UserRegistrationCtl |
|  | \*/ |
|  |  |
|  | /\*\* |
|  | \* UserRegistration functionality Controller. Performs operation for Validate and add a User |
|  | \* As Student Role |
|  | \*/ |
|  |  |
|  | @WebServlet(name = "UserRegistrationCtl", urlPatterns = { "/UserRegistrationCtl" }) |
|  | public class UserRegistrationCtl extends BaseCtl { |
|  | public static final String OP\_SIGN\_UP = "SignUp"; |
|  |  |
|  | private static Logger log = Logger.getLogger(UserRegistrationCtl.class); |
|  | /\*\* |
|  | \* Validate input Data Entered By User |
|  | \* |
|  | \* @param request |
|  | \* @return |
|  | \*/ |
|  | @Override |
|  | protected boolean validate(HttpServletRequest request) { |
|  | log.debug("UserRegistrationCtl Method validate Started"); |
|  |  |
|  | boolean pass = true; |
|  |  |
|  | String login = request.getParameter("login"); |
|  |  |
|  |  |
|  | if (DataValidator.isNull(request.getParameter("login"))) { |
|  | request.setAttribute("login", |
|  | PropertyReader.getValue("error.email", "Login")); |
|  | pass = false; |
|  | } |
|  |  |
|  |  |
|  | if (DataValidator.isNull(request.getParameter("confirmPassword"))) { |
|  | request.setAttribute("confirmPassword", PropertyReader.getValue( |
|  | "error.require", "Confirm Password")); |
|  | pass = false; |
|  | } |
|  |  |
|  |  |
|  | if (DataValidator.isNull(request.getParameter("password"))) { |
|  | request.setAttribute("password", |
|  | PropertyReader.getValue("error.require", "Password")); |
|  | pass = false; |
|  |  |
|  | } else if (!DataValidator.isPassword(request.getParameter("password"))) { |
|  | request.setAttribute("password", |
|  | PropertyReader.getValue("error.password", "Password")); |
|  | return false; |
|  | }else if (!DataValidator.isPassword(request.getParameter("password"))) { |
|  | request.setAttribute("password", |
|  | PropertyReader.getValue("error.password", "Password")); |
|  | return false; |
|  | } |
|  |  |
|  | if (!request.getParameter("password").equals( |
|  | request.getParameter("confirmPassword")) |
|  | && !"".equals(request.getParameter("confirmPassword"))) { |
|  |  |
|  | request.setAttribute("confirmPassword", PropertyReader.getValue("error.confirmPassword","Confirm Password")); |
|  | pass = false; |
|  | } |
|  |  |
|  |  |
|  |  |
|  | if (DataValidator.isNull(request.getParameter("firstName"))) { |
|  |  |
|  | request.setAttribute("firstName", PropertyReader.getValue("error.require", "First Name")); |
|  | pass = false; |
|  | } |
|  |  |
|  | if (DataValidator.isNull(request.getParameter("lastName"))) { |
|  | request.setAttribute("lastName", PropertyReader.getValue("error.require", "Last Name")); |
|  | pass = false; |
|  | } |
|  |  |
|  |  |
|  |  |
|  | log.debug("UserRegistrationCtl Method validate Ended"); |
|  | return pass; |
|  | } |
|  |  |
|  | /\*\* |
|  | \* Populates bean object from request parameters |
|  | \* |
|  | \* @param request |
|  | \* @return |
|  | \*/ |
|  | @Override |
|  | protected BaseBean populateBean(HttpServletRequest request) { |
|  | log.debug("UserRegistrationCtl Method populatebean Started"); |
|  |  |
|  | UserBean bean = new UserBean(); |
|  |  |
|  | bean.setId(DataUtility.getLong(request.getParameter("id"))); |
|  |  |
|  | bean.setRoleId(2L); |
|  |  |
|  |  |
|  | bean.setLogin(DataUtility.getString(request.getParameter("login"))); |
|  |  |
|  | bean.setPassword(DataUtility.getString(request.getParameter("password"))); |
|  |  |
|  | bean.setConfirmPassword(DataUtility.getString(request |
|  | .getParameter("confirmPassword"))); |
|  | bean.setFirstName(DataUtility.getString(request.getParameter("firstName"))); |
|  | bean.setLastName(DataUtility.getString(request.getParameter("lastName"))); |
|  |  |
|  | populateDTO(bean, request); |
|  |  |
|  | log.debug("UserRegistrationCtl Method populatebean Ended"); |
|  |  |
|  | return bean; |
|  | } |
|  |  |
|  | private static final long serialVersionUID = 1L; |
|  |  |
|  | /\*\* |
|  | \* @see HttpServlet#HttpServlet() |
|  | \*/ |
|  | public UserRegistrationCtl() { |
|  | super(); |
|  | // TODO Auto-generated constructor stub |
|  | } |
|  | /\*\* |
|  | \* Contains display logic |
|  | \*/ |
|  | /\*\* |
|  | \* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response) |
|  | \*/ |
|  | protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException { |
|  | log.debug("UserRegistrationCtl Method doGet Started"); |
|  | ServletUtility.forward(getView(), request, response); |
|  |  |
|  | } |
|  | /\*\* |
|  | \* Contains submit logic |
|  | \*/ |
|  | /\*\* |
|  | \* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response) |
|  | \*/ |
|  | protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException { |
|  | System.out.println("in post method"); |
|  | log.debug("UserRegistrationCtl Method doPost Started"); |
|  |  |
|  | String op = DataUtility.getString(request.getParameter("operation")); |
|  | // get model |
|  | UserModel model = new UserModel(); |
|  |  |
|  | long id = DataUtility.getLong(request.getParameter("id")); |
|  |  |
|  |  |
|  | if (OP\_SIGN\_UP.equalsIgnoreCase(op)) { |
|  |  |
|  | UserBean bean = (UserBean) populateBean(request); |
|  | try { |
|  | // System.out.println("in try sign up"); |
|  | long pk = model.registerUser(bean); |
|  | //System.out.println("register"); |
|  | bean.setId(pk); |
|  |  |
|  | request.getSession().setAttribute("UserBean", bean); |
|  | ServletUtility.setSuccessMessage("User Successfully Registered", request); |
|  | Long bid=(Long)request.getSession().getAttribute("boId"); |
|  | if(bid>0) { |
|  | UserBean uBean=model.findByPK(pk); |
|  | request.getSession().setAttribute("user",uBean); |
|  | ServletUtility.redirect(ATBView.BOOK\_CTL, request, response); |
|  | return; |
|  | } |
|  | ServletUtility.forward(ATBView.USER\_REGISTRATION\_VIEW, request, response); |
|  | return; |
|  | } catch (DuplicateRecordException e) { |
|  | log.error(e); |
|  | ServletUtility.setBean(bean, request); |
|  | ServletUtility.setErrorMessage("Login id already exists", |
|  | request); |
|  | ServletUtility.forward(getView(), request, response); |
|  | } catch (ApplicationException e) { |
|  | ServletUtility.handleException(e, request, response); |
|  | e.printStackTrace(); |
|  | return; |
|  | } |
|  | }else if (OP\_RESET.equalsIgnoreCase(op)) { |
|  | ServletUtility.redirect(ATBView.USER\_REGISTRATION\_CTL, request, response); |
|  | return; |
|  | } |
|  | log.debug("UserRegistrationCtl Method doPost Ended"); |
|  | } |
|  | /\*\* |
|  | \* Returns the VIEW page of this Controller |
|  | \* |
|  | \* @return |
|  | \*/ |
|  | @Override |
|  | protected String getView() { |
|  | return ATBView.USER\_REGISTRATION\_VIEW; |
|  | } |
|  |  |
|  | } |

|  |
| --- |
| /\*\* |
|  | \* Servlet implementation class LoginCtl |
|  | \*/ |
|  | /\*\* |
|  | \* Login functionality Controller. Performs operation for Authentication, and |
|  | \* Session Creation and Give permission to access all Functionality regarding |
|  | \* Role |
|  | \* |
|  | \* |
|  | \* @author Navigable Set |
|  | \* @version 1.0 |
|  | \* @Copyright (c) Navigable Set |
|  | \* |
|  | \*/ |
|  | @WebServlet(name = "LoginCtl", urlPatterns = { "/LoginCtl" }) |
|  | public class LoginCtl extends BaseCtl { |
|  |  |
|  | private static final long serialVersionUID = 1L; |
|  |  |
|  | public static final String OP\_REGISTER = "Register"; |
|  | public static final String OP\_SIGN\_IN = "SignIn"; |
|  | public static final String OP\_SIGN\_UP = "SignUp"; |
|  | public static final String OP\_LOG\_OUT = "logout"; |
|  | public static String HIT\_URI = null; |
|  |  |
|  | private static Logger log = Logger.getLogger(LoginCtl.class); |
|  |  |
|  | /\*\* |
|  | \* @see HttpServlet#HttpServlet() |
|  | \*/ |
|  | public LoginCtl() { |
|  | super(); |
|  | // TODO Auto-generated constructor stub |
|  | } |
|  |  |
|  | /\*\* |
|  | \* Validate input Data Entered By User |
|  | \* |
|  | \* @param request |
|  | \* @return |
|  | \*/ |
|  | @Override |
|  | protected boolean validate(HttpServletRequest request) { |
|  |  |
|  | log.debug("LoginCtl Method validate Started"); |
|  |  |
|  | boolean pass = true; |
|  |  |
|  | String op = request.getParameter("operation"); |
|  |  |
|  | if (OP\_SIGN\_UP.equals(op) || OP\_LOG\_OUT.equals(op)) { |
|  | return pass; |
|  | } |
|  |  |
|  |  |
|  | String login = request.getParameter("login"); |
|  |  |
|  | if (DataValidator.isNull(login)) { |
|  | request.setAttribute("login", PropertyReader.getValue("error.require", "Login Id")); |
|  | pass = false; |
|  |  |
|  | } |
|  | if (DataValidator.isNull(request.getParameter("password"))) { |
|  | request.setAttribute("password", PropertyReader.getValue("error.require", "Password")); |
|  | pass = false; |
|  | } |
|  | log.debug("LoginCtl Method validate Ended"); |
|  | return pass; |
|  | } |
|  |  |
|  | /\*\* |
|  | \* Populates bean object from request parameters |
|  | \* |
|  | \* @param request |
|  | \* @return |
|  | \*/ |
|  | @Override |
|  | protected BaseBean populateBean(HttpServletRequest request) { |
|  |  |
|  | log.debug("LoginCtl Method populateBean Started"); |
|  |  |
|  | UserBean bean = new UserBean(); |
|  |  |
|  | bean.setId(DataUtility.getLong(request.getParameter("id"))); |
|  |  |
|  | bean.setLogin(DataUtility.getString(request.getParameter("login"))); |
|  |  |
|  | bean.setPassword(DataUtility.getString(request.getParameter("password"))); |
|  |  |
|  | log.debug("LOginCtl Method PopulatedBean End"); |
|  |  |
|  | return bean; |
|  | } |
|  |  |
|  | /\*\* |
|  | \* Display Login form |
|  | \* |
|  | \*/ |
|  | /\*\* |
|  | \* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse |
|  | \* response) |
|  | \*/ |
|  | protected void doGet(HttpServletRequest request, HttpServletResponse response) |
|  | throws ServletException, IOException { |
|  |  |
|  | log.debug("Method doGet Started"); |
|  |  |
|  |  |
|  | HttpSession session = request.getSession(true); |
|  | String op = DataUtility.getString(request.getParameter("operation")); |
|  |  |
|  | UserModel model = new UserModel(); |
|  |  |
|  | long id = DataUtility.getLong(request.getParameter("id")); |
|  |  |
|  | long Mid=DataUtility.getLong(request.getParameter("bId")); |
|  | session.setAttribute("boId",Mid); |
|  |  |
|  | if (id > 0) { |
|  | UserBean userBean; |
|  | try { |
|  | userBean = model.findByPK(id); |
|  | ServletUtility.setBean(userBean, request); |
|  |  |
|  | } catch (Exception e) { |
|  | log.error(e); |
|  | ServletUtility.handleException(e, request, response); |
|  | return; |
|  | } |
|  | } else if (OP\_LOG\_OUT.equals(op)) { |
|  | session = request.getSession(false); |
|  | session.invalidate(); |
|  | ServletUtility.setSuccessMessage("You have been logged out successfully", request); |
|  |  |
|  | ServletUtility.forward(ATBView.LOGIN\_VIEW, request, response); |
|  | return; |
|  | } |
|  | if (session.getAttribute("user") != null) { |
|  | ServletUtility.redirect(ATBView.WELCOME\_CTL, request, response); |
|  | return; |
|  | } |
|  | ServletUtility.forward(getView(), request, response); |
|  | log.debug("Method doGet end"); |
|  | } |
|  |  |
|  | /\*\* |
|  | \* Submit Logic |
|  | \*/ |
|  |  |
|  | /\*\* |
|  | \* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse |
|  | \* response) |
|  | \*/ |
|  | protected void doPost(HttpServletRequest request, HttpServletResponse response) |
|  | throws ServletException, IOException { |
|  |  |
|  |  |
|  |  |
|  | log.debug(" LoginCtl Method doPost Started"); |
|  |  |
|  | HttpSession session = request.getSession(true); |
|  |  |
|  | String op = DataUtility.getString(request.getParameter("operation")); |
|  | // get Model |
|  | UserModel model = new UserModel(); |
|  |  |
|  | long id = DataUtility.getLong(request.getParameter("id")); |
|  |  |
|  |  |
|  | if (OP\_SIGN\_IN.equalsIgnoreCase(op)) { |
|  | UserBean bean = (UserBean) populateBean(request); |
|  | try { |
|  | bean = model.authenticate(bean.getLogin(), bean.getPassword()); |
|  |  |
|  | if (bean != null) { |
|  | session.setAttribute("user", bean); |
|  | session.setMaxInactiveInterval(10 \* 6000); |
|  |  |
|  |  |
|  | // save state of session remember URL |
|  | String uri = request.getParameter("uri"); |
|  |  |
|  | Long bid=(Long)request.getSession().getAttribute("boId"); |
|  | if(bid>0) { |
|  | ServletUtility.redirect(ATBView.BOOK\_CTL, request, response); |
|  | return; |
|  | } |
|  |  |
|  | if (uri == null || "null".equalsIgnoreCase(uri)) { |
|  | ServletUtility.redirect(ATBView.WELCOME\_CTL, request, response); |
|  | return; |
|  | } else { |
|  | ServletUtility.redirect(uri, request, response); |
|  | } |
|  | return; |
|  | } else { |
|  | bean = (UserBean) populateBean(request); |
|  | ServletUtility.setBean(bean, request); |
|  | ServletUtility.setErrorMessage("Invalid LoginId And Password", request); |
|  | } |
|  |  |
|  | } catch (ApplicationException e) { |
|  | log.error(e); |
|  | ServletUtility.handleException(e, request, response); |
|  |  |
|  | return; |
|  | } |
|  | } else if (OP\_SIGN\_UP.equalsIgnoreCase(op)) { |
|  | ServletUtility.redirect(ATBView.USER\_REGISTRATION\_CTL, request, response); |
|  | return; |
|  | } |
|  |  |
|  |  |
|  | ServletUtility.forward(getView(), request, response); |
|  | log.debug("UserCtl Method doPost Ended"); |
|  | } |
|  |  |
|  | /\*\* |
|  | \* Returns the VIEW page of this Controller |
|  | \* |
|  | \* @return |
|  | \*/ |
|  | @Override |
|  | protected String getView() { |
|  | return ATBView.LOGIN\_VIEW; |
|  | } |
|  |  |
|  | } |

|  |
| --- |
| /\*\* |
|  | \* Servlet implementation class FlightListCtl |
|  | \*/ |
|  | @WebServlet(name = "FlightListCtl", urlPatterns = { "/ctl/FlightListCtl" }) |
|  | public class FlightListCtl extends BaseCtl { |
|  | private static final long serialVersionUID = 1L; |
|  |  |
|  | private static Logger log = Logger.getLogger(FlightListCtl.class); |
|  |  |
|  | /\*\* |
|  | \* Populates bean object from request parameters |
|  | \* |
|  | \* @param request |
|  | \* @return |
|  | \*/ |
|  | @Override |
|  | protected BaseBean populateBean(HttpServletRequest request) { |
|  | log.debug("FlightListCtl populateBean method start"); |
|  | FlightBean bean = new FlightBean(); |
|  | bean.setFightName(DataUtility.getString(request.getParameter("name"))); |
|  | bean.setFlightNo(DataUtility.getString(request.getParameter("flightNo"))); |
|  | log.debug("FlightListCtl populateBean method end"); |
|  | return bean; |
|  | } |
|  |  |
|  | /\*\* |
|  | \* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response) |
|  | \*/ |
|  | protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException { |
|  | log.debug("FlightListCtl doGet method start"); |
|  | List list = null; |
|  | int pageNo = 1; |
|  | int pageSize = DataUtility.getInt(PropertyReader.getValue("page.size")); |
|  |  |
|  | FlightModel model = new FlightModel(); |
|  | FlightBean bean = (FlightBean) populateBean(request); |
|  | try { |
|  | list = model.search(bean, pageNo, pageSize); |
|  | if (list == null || list.size() == 0) { |
|  | ServletUtility.setErrorMessage("No Record Found", request); |
|  | } |
|  | ServletUtility.setList(list, request); |
|  | ServletUtility.setPageNo(pageNo, request); |
|  | ServletUtility.setPageSize(pageSize, request); |
|  | ServletUtility.forward(getView(), request, response); |
|  |  |
|  | } catch (ApplicationException e) { |
|  | ServletUtility.handleException(e, request, response); |
|  | e.printStackTrace(); |
|  | return; |
|  | } |
|  | log.debug("FlightListCtl doGet method end"); |
|  | } |
|  |  |
|  | /\*\* |
|  | \* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response) |
|  | \*/ |
|  | protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException { |
|  | log.debug("FlightListCtl doPost method start"); |
|  | List list = null; |
|  |  |
|  | int pageNo = DataUtility.getInt(request.getParameter("pageNo")); |
|  |  |
|  | int pageSize = DataUtility.getInt(request.getParameter("pageSize")); |
|  |  |
|  | pageNo = (pageNo == 0) ? 1 : pageNo; |
|  |  |
|  | pageSize = (pageSize == 0) ? DataUtility.getInt(PropertyReader.getValue("page.size")) : pageSize; |
|  |  |
|  | FlightBean bean = (FlightBean) populateBean(request); |
|  |  |
|  | FlightModel model = new FlightModel(); |
|  | String[] ids = request.getParameterValues("ids"); |
|  | String op = DataUtility.getString(request.getParameter("operation")); |
|  |  |
|  | if (OP\_SEARCH.equalsIgnoreCase(op) || OP\_NEXT.equalsIgnoreCase(op) || OP\_PREVIOUS.equalsIgnoreCase(op)) { |
|  |  |
|  | if (OP\_SEARCH.equalsIgnoreCase(op)) { |
|  |  |
|  | pageNo = 1; |
|  |  |
|  | } else if (OP\_NEXT.equalsIgnoreCase(op)) { |
|  |  |
|  | pageNo++; |
|  | } else if (OP\_PREVIOUS.equalsIgnoreCase(op) && pageNo > 1) { |
|  |  |
|  | pageNo--; |
|  | } |
|  | } else if (OP\_NEW.equalsIgnoreCase(op)) { |
|  | ServletUtility.redirect(ATBView.FLIGHT\_CTL, request, response); |
|  | return; |
|  | } else if (OP\_DELETE.equalsIgnoreCase(op)) { |
|  | pageNo = 1; |
|  | if (ids != null && ids.length > 0) { |
|  | FlightBean deletebean = new FlightBean(); |
|  | for (String id : ids) { |
|  | deletebean.setId(DataUtility.getInt(id)); |
|  | try { |
|  | model.delete(deletebean); |
|  | } catch (ApplicationException e) { |
|  | ServletUtility.handleException(e, request, response); |
|  | e.printStackTrace(); |
|  | return; |
|  | } |
|  | } |
|  | ServletUtility.setSuccessMessage("Data Deleted Successfully", request); |
|  | } else { |
|  | ServletUtility.setErrorMessage("Select at least one record", request); |
|  | } |
|  | } else if (OP\_RESET.equalsIgnoreCase(op)) { |
|  | ServletUtility.redirect(ATBView.FLIGHT\_LIST\_CTL, request, response); |
|  | return; |
|  |  |
|  | } |
|  |  |
|  | try { |
|  |  |
|  | list = model.search(bean, pageNo, pageSize); |
|  | if (list == null || list.size() == 0) { |
|  | ServletUtility.setErrorMessage("NO Record Found", request); |
|  | } |
|  | ServletUtility.setList(list, request); |
|  | ServletUtility.setPageNo(pageNo, request); |
|  | ServletUtility.setPageSize(pageSize, request); |
|  | ServletUtility.forward(getView(), request, response); |
|  | } catch (ApplicationException e) { |
|  | ServletUtility.handleException(e, request, response); |
|  | e.printStackTrace(); |
|  | return; |
|  | } |
|  |  |
|  | log.debug("FlightListCtl doPost method end"); |
|  | } |
|  |  |
|  | @Override |
|  | protected String getView() { |
|  | // TODO Auto-generated method stub |
|  | return ATBView.FLIGHT\_LIST\_VIEW; |
|  | } |
|  |  |
|  | } |