

Ivanov Ivan

Java Developer

Project

HRMS-INFO – control system for HR-specialists, automates the process of selection, training of human resources for the implementation of IT projects.

Development environment



Server environment

APACHE
HTTP SERVER



Data transfer object

```
public class RoleDto implements Serializable {  
  
    private static final long serialVersionUID = -5448036812725394372L;  
  
    private Long roleId;  
    private String name;  
    private Long numberOfRoles;  
  
    public RoleDto() {  
    }  
  
    public Long getRoleId() { return roleId; }  
  
    public void setRoleId(Long roleId) { this.roleId = roleId; }  
  
    public String getName() { return name; }  
  
    public void setName(String name) { this.name = name; }  
  
    public Long getNumberOfRoles() { return numberOfRoles; }  
  
    public void setNumberOfRoles(Long numberOfRoles) { this.numberOfRoles = numberOfRoles; }  
}
```

Data access object

```
@Override
public List<RoleDto> getUserRoleCount() throws ServiceException {
    List<RoleDto> userRoles = new ArrayList<>();
    try {
        String hql = queries.getProperty("role.user.count");
        Query query = userRoleDao.getQuery(hql);
        query.setParameter("cust", getCustVO().getId());
        List<Object[]> list = query.list();
        RoleDto roleDto;
        for (Object[] userRole : list) {
            roleDto = new RoleDto();
            roleDto.setName((String) userRole[0]);
            roleDto.setNumberOfRoles((Long) userRole[1]);
            userRoles.add(roleDto);
        }
    } catch (DaoException e) {
        throw new ServiceException(e, ServiceErrorCode.RMS_SRVC_300);
    }
    return userRoles;
}
```

```
@Override
public RoleDto getUsersCountWithoutRole() throws ServiceException {
    RoleDto userRole = null;
    try {
        String hql = queries.getProperty("role.user.without");
        Query query = userRoleDao.getQuery(hql);
        query.setParameter("cust", getCustVO().getId());
        List<Object> list = query.list();
        if (list != null && list.size() > 0) {
            userRole = new RoleDto();
            if (list.get(0) != null) {
                userRole.setNumberOfRoles((Long) list.get(0));
            }
            userRole.setName("Without roles");
        }
    } catch (DaoException e) {
        throw new ServiceException(e, ServiceErrorCode.RMS_SRVC_301);
    }
    return userRole;
}
```

Service

```
public interface UserRoleFacade {  
  
    List<RoleDto> getUserRoleCount() throws ServiceException;  
  
    RoleDto getUsersCountWithoutRole() throws ServiceException;  
}
```

```
public class UserRoleFacadeImpl implements UserRoleFacade {  
  
    UserRoleService userRoleService;  
  
    public void setUserUserRoleService(UserRoleService userRoleService)  
  
    @Override  
    public List<RoleDto> getUserRoleCount() throws ServiceException {  
        return userRoleService.getUserRoleCount();  
    }  
  
    @Override  
    public RoleDto getUsersCountWithoutRole() throws ServiceException {  
        return userRoleService.getUsersCountWithoutRole();  
    }  
}
```

Controller

```
@MethodInfo(application = Application.RMS_INFO, userAction = ActionCode.RMS_ACTION_CODE_301)
@RequestMapping(method = {RequestMethod.GET, RequestMethod.POST})
@PreAuthorize("hasAnyRole('Admin', 'Chief Methodist') or principal.hasAnyTask('Add new role', 'Edit role', 'Delete role')")
public String getRolesPage(ModelMap model) throws UIException {
    CustomUser user = (CustomUser) SecurityContextHolder.getContext().getAuthentication().getPrincipal();
    if (user.getCustomerVO().getId() == 1) {
        try {
            Map<String, RoleDto> roleDtoMap;
            List<RoleDto> sortRoles = null;
            List<RoleDto> roleDtoList = userRoleFacade.getUserRoleCount();
            if (roleDtoList != null && roleDtoList.size() > 0) {
                roleDtoMap = new HashMap<String, RoleDto>();
                for (RoleDto roleDto : roleDtoList) {
                    roleDtoMap.put(roleDto.getName(), roleDto);
                }
                sortRoles = new ArrayList<RoleDto>();
                for (String role : sortedRoles) {
                    if (roleDtoMap.containsKey(role)) {
                        sortRoles.add(roleDtoMap.get(role));
                    }
                }
            }
            RoleDto withoutRole = userRoleFacade.getUsersCountWithoutRole();
            model.put("roles", sortRoles);
            model.put("withoutRole", withoutRole);
        } catch (ServiceException e) {
            throw new UIException(e, UIErrorCode.RMS_UI_305);
        }
        return "showListRoles";
    } else {
        return "showRoles";
    }
}
```


Apache Tiles

```
<!-- Role views -->
```

```
<definition name="showRoles" extends="base.definition">  
  <put-attribute name="title" value="title.showRoles"/>  
  <put-attribute name="content" value="/WEB-INF/tiles/roles/listRoles.jsp"/>  
</definition>
```

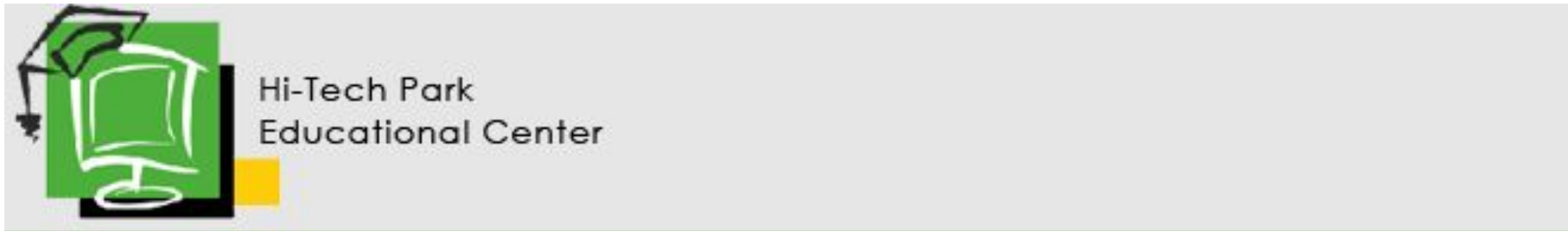
```
<definition name="showListRoles" extends="base.definition">  
  <put-attribute name="title" value="title.showRoles"/>  
  <put-attribute name="content" value="/WEB-INF/tiles/roles/listRolesReports.jsp"/>  
</definition>
```

Hibernate Query Language

```
role.user.count = select ur.roleName, count(uu)
                  from UserRoleVO ur
                  left join ur.userVOs uu
                  where ur.customer.id like :cust
                     and (uu is null or uu.customer.id like :cust)
                  group by ur.roleName
```

```
role.user.without = select count(uu)
                   from UserVO uu
                   left join uu.userRoleVOs ur
                   where uu.customer.id like :cust
                      and (ur is null or ur.customer.id not like :cust
                          or ur.customer.id is null)
                   group by uu.customer.id
```

Web Application



Dashboard
My account
▼ Security
Users
Roles
➤ Profiles

Role	Number of users
Admin	1
Chief Methodist	1
Methodist	1
HR-Manager	1
Trainer	1
Without role	0

Thank you for attention!