Excel lightweight reporting tool (Eliweirt)

Content

Excel lightweight reporting tool (Eliweirt)

1 Overview

Excel lightweight reporting tool GWT based application, which provides report geneneration in Excel format. On the other hand Eliweirt demonstrates the capabilities of the set of libraries (henceforth referred as framework, for details see Databuffer, Dataservice, UI) on which it is based. Since reporting in most business applications is not the main purpose, then, for example, GWT application for a variety of business tasks can be obtained from Eliweirt by adding the required business functionality. The opposite issue (embedding reporting functionality into existing GWT application) can be solved by adding the Dataservice library in your project.

2 Main Features

2.1 JAAS based authentication

User authentication process is based on JAAS and doesn't depend on Web application, therefore authentication process implementation can vary depending on the security requirements without affecting the Web application itself. Eliweirt user authentication implementation is following: users and their roles that define access rights to the Web application resources are stored in the database. For Web server (further it is understood Tomcat) JAAS configuration file is defined as:

```
eliweirt {
        org.homedns.mkh.eliweirt.server.AppSecurityService required
        jdbc_login_resource_name="jdbc/login"
        jdbc_db_resource_name="jdbc/db"
        login_db="acr_user_login"
        access_rights_db="user_access_right"
        sys_params_db="sms_slot_floor"
        checkup_service_class="org.homedns.mkh.eliweirt.server.AppCheckupServcie"
        attempt_threshold="3";
};
```

In JAAS configuration references to login datasource (jdbc_login_resource_name="jdbc/login") and db datasource (jdbc_login_resource_name="jdbc/db") are specified, which in turn are defined in the context of the Web server:

```
        <!-- Specify a JDBC datasource for login-->
        <Resource name="jdbc/login" auth="ContainerLogin"
        type="javax.sql.DataSource" username="gms_login_user" password="l251957"
        driverClassName="org.postgresql.Driver"
        url="jdbc:postgresql://localhost:5432/gms"
        maxActive="20" maxIdle="3" maxWait="10000" />

        <!-- Specify a JDBC datasource for application-->
        <Resource name="jdbc/db" auth="ContainerDB"
        type="javax.sql.DataSource" username="gms_user" password="251957"
        driverClassName="org.postgresql.Driver"
        url="jdbc:postgresql://localhost:5432/gms"
        maxActive="50" maxIdle="5" maxWait="10000" />
```

The database user (database role in terms of Postgresql DBMS) for login datasource has minimally sufficient access rights, i.e. read only from the users table. Further into JAAS configuration file are specified:

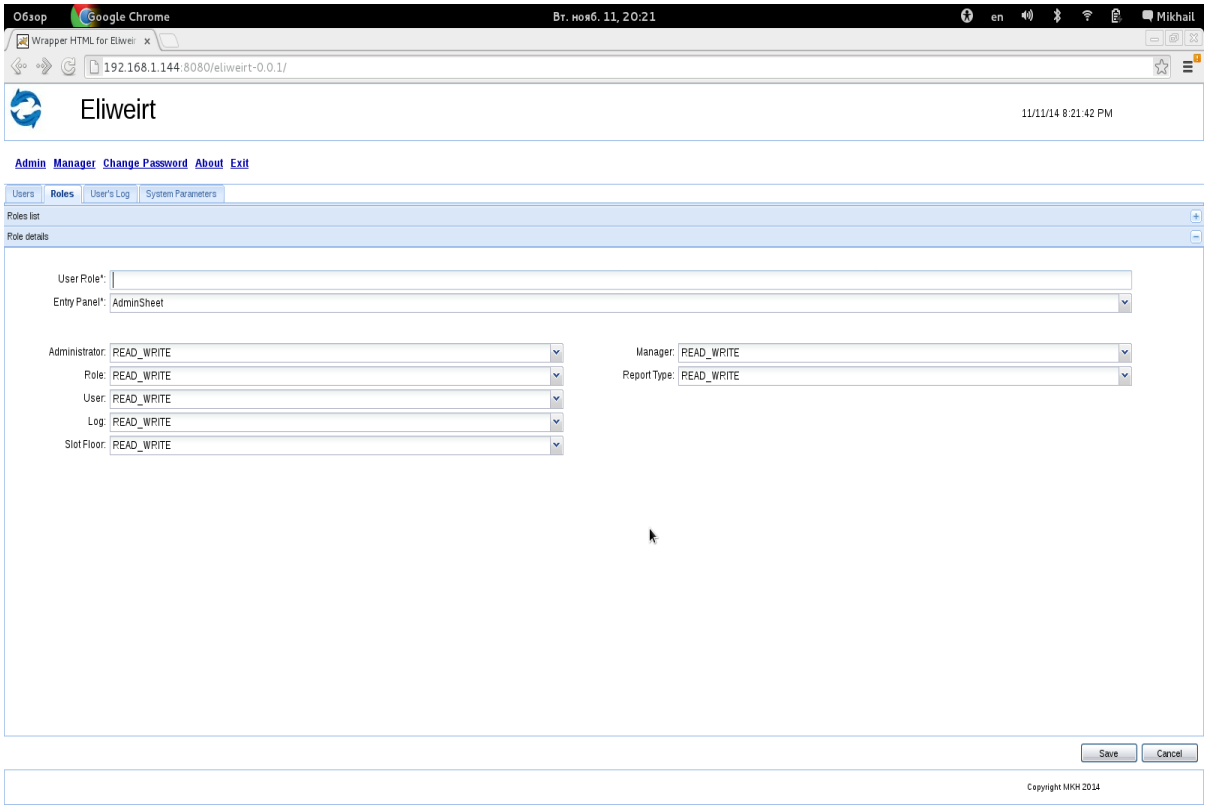
- Databuffer name where to data from users table will be retrieved during authentication process,
- Databuffer name where to roles and their associated access rights will be retrieved, if the authentication process succeeds
- limit of unsuccessful authentication attempts,
- SecurityService implementation
- CheckupService implementation.

Authentication is performed as follows:

1. SecurityService initializes CheckupService when login / password pair is entered (for details see Dataservice API).
2. CheckupService checks that the counter of failed login attempts from that IP address does not exceed setting limit, if so, the authentication process is completed with failure, otherwise continue
3. CheckupService sends a request to the database is this user valid, ie, a user with that login and password exists, and his status is "active". If response "yes", then the authentication process succeeds, otherwise, the process is not successful and failed attempts counter is incremented.

2.2 Granting access rights depending on user role

After successful login SecurityService close connection via login datasource and connect to the database via db datasource, retrieves roles and their associated access rights. This information is used to determine the user's access rights to the application resources. There are following available access rights: no access, read only, read write. Usually the system administrator or root have the right to modify the access rights for a particular role.



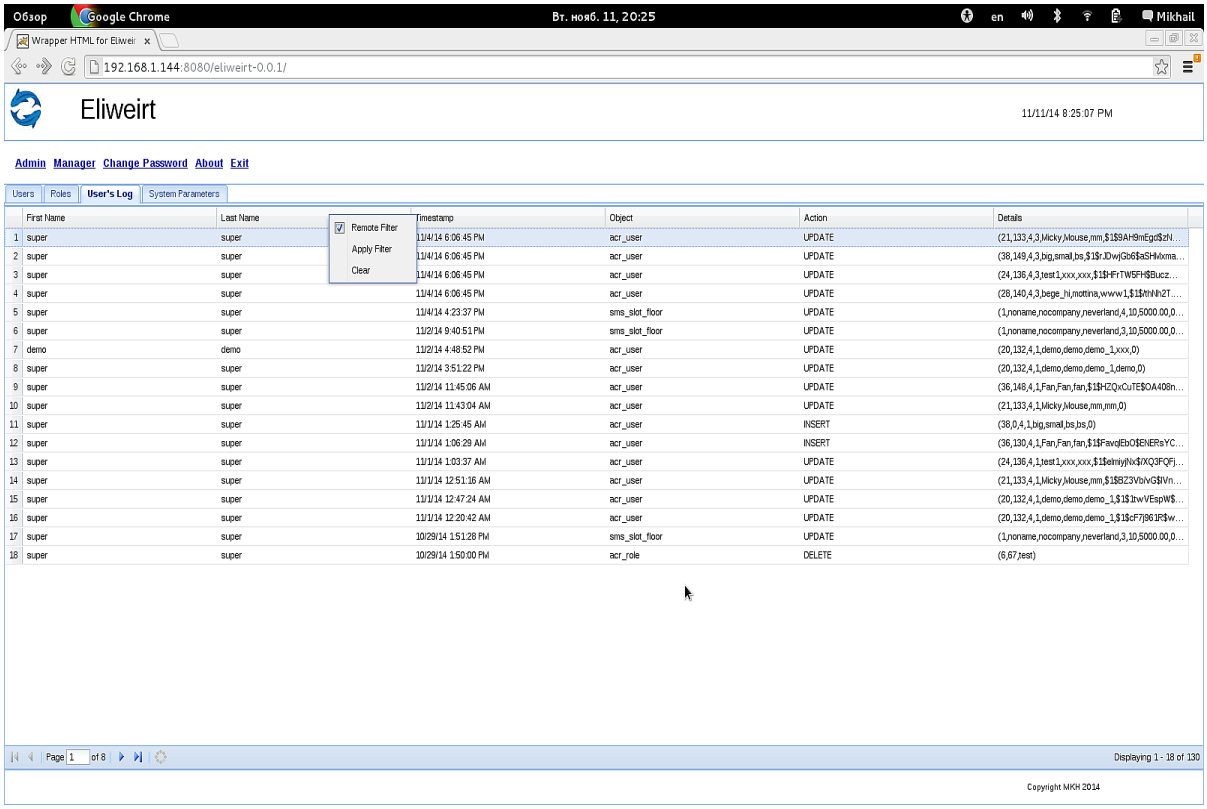
Editing access rights for role

2.3 Client IP address blocking after N incorrect login attempts

In the JAAS configuration file parameter "attempt_threshold" should be defined, it specifies the number of failed authentication attempts, 0 means that the parameter is turned off. During logon process CheckupService compares the current count of failed attempts with the value of this parameter. If the counter exceeds the value of the parameter, the authentication process is aborted with failure, the counter value is written to the database, and all requests to log in from that IP address is blocked. If the counter does not exceed the value of the parameter, the counter is incremented by one, and next attempt to logon is available. Upon successful logon counter value in the database is reset.

2.4 User modifying actions logging

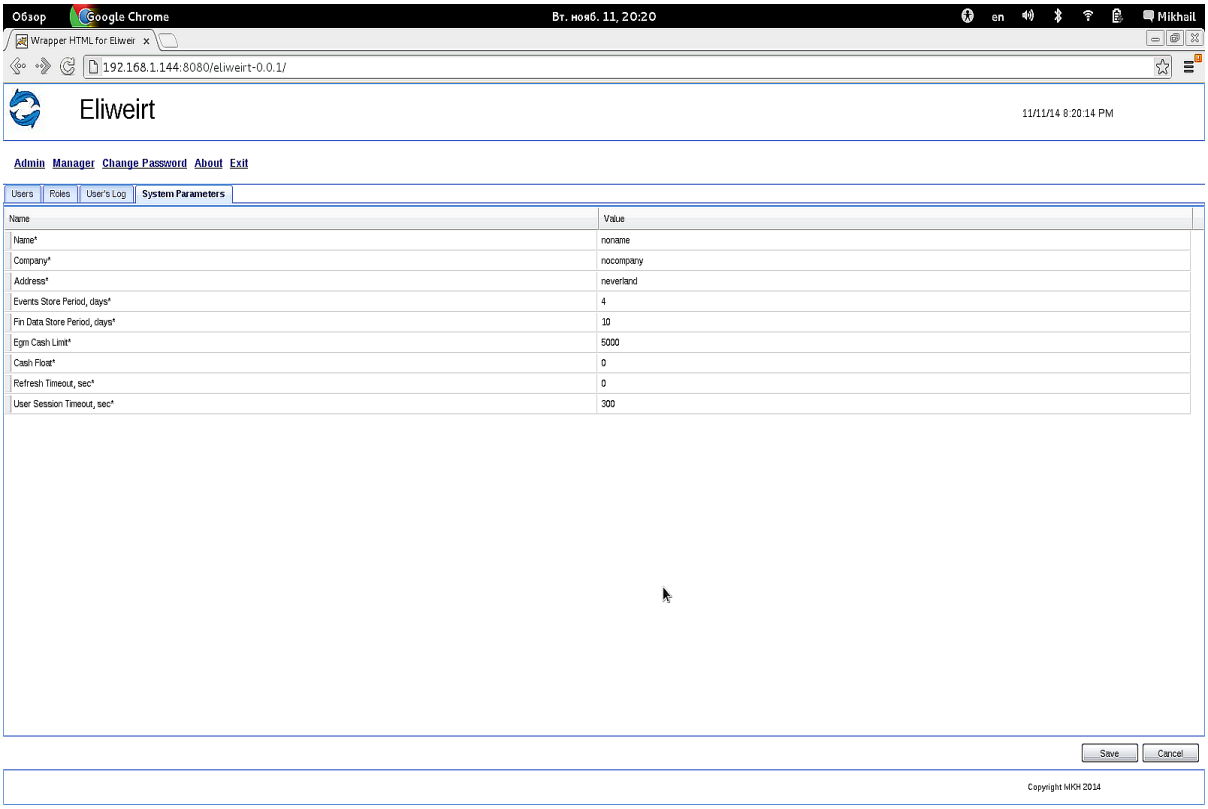
All user actions to modify data are logged. So easy to find out when and who made certain changes in the data.



User's actions log

2.5 User logoff on timeout

Automatically logoff if there is no user activity (keyboard or mouse events) for the specified number of seconds is activated by setting the value of the parameter "user session timeout" greater than 0. 0 means parameter is turned off.



System parameters editing

2.6 Multilanguage support

To localize application for different locales on client side Static String Internationalization is used, on the server side ResourceBundle is used.

2.7 Generation custom defined excel style reports

All report types used in the application and their parameters are described in the database. Under the description here means for report type: the data buffer name, which will retrieve the data from the database, for the report parameter: name, data type, serial number (see chap. 4). For each report type Excel template should be created (for details see example). Description of data buffer contains information where data and report parameters values to be inserted into the template (for details see example). At the time of the report generation, report data will be inserted into the template, excel file will be generated and downloaded to the client workstation. The user can open the file in an application that understands the xls format and / or save the file for later viewing / editing.

2.8 View and edit records

For easy viewing grid content, it is divided into pages, the number of records per page is set in the databuffer description. To navigate through the pages paging toolbar at the bottom of page is used. The grid provides records sorting, columns hide / show, setting filter criteria, filtering records by specified criteria, both locally on the client and remotely on the server. By double clicking on selected record edit form is activated, then user can edit the selected record or create a new one. After editing data they can be saved or changes are canceled.

3 Servlets Definition

For web application based on the framework following servlets must be defined in web.xml. Eliweirt example see here.

4 Database Description

For demo purposes it is sufficient to use database dump. For detailed analysis database model is described here. It contents database logical model description and specifies which parts of the model are responsible for

supporting one or another application functionality.

5 Deploy application

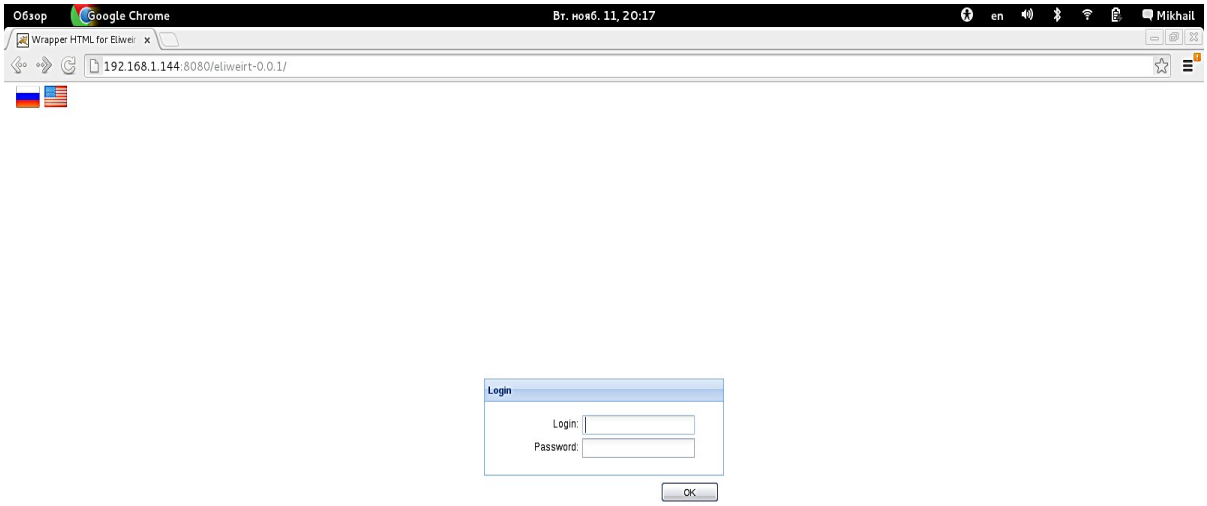
In Eliweirt project directory run build war:

$ mvn package

then put eliweirt-1.0.1.war to the web server webapp directory. Make sure that

- JAAS configuration file is defined as describe in 2.1 and is in $CATALINA_HOME/conf/,
- in $CATALINA_HOME/bin/catalina.sh must be added the following JAVA_OPTS="$JAVA_OPTS "-Djava.security.auth.login.config=="$CATALINA_HOME/conf/jaas.config",
- The data sources are defined as describe in 2.1 and added to the META-INF/context.xml

In browser address bar input https://<server name or server IP>:<port number, default for https 8443>/eleweirt-1.0.1/Eleweirt.html or http://<server name or server IP>:<port number, default for http 8080>/eleweirt-1.0.1/Eleweirt.html and see application login panel.



Login panel

Because of security reasons preferable is HTTPS, how to enable SSL for Web server see SSL Configuration HOW-TO

6 License

LICENSE, NOTICE