2013

Finoux Solutions Pvt. Ltd.

Sagar Raut
Naveen Kumar

[ADMIN PANEL GUIDE]

A Complete Guide For Admin Panel System.

1 OVERVIEW 3

2 CONCEPT 3

3 API 3

3.1 REST API'S 3

3.1.1 Authentication 3

3.1.1.1 Authenticate 3

3.1.1.2 Authorize 4

3.1.1.3 Page Event Authorization 4

3.1.1.4 Get User List 5

4 WEB SITE 7

4.1 WEB.CONFIG 7

5 APPENDIX 9

1 Overview

Admin Panel is a standalone Admin System which provides facility line Authentication, Authorization, User Management, Role Management under one window. Admin Panel Exposes API for other products/ Projects to Utilize these API's and integrate their Product/ Module with Admin Panel Seamlessly.

The Consumer of the Admin Panel can concentrate on their Products/ Projects business requirements whereas the other mentioned requirements are handled by the Admin Panel.

This document is a brief regarding the approach, architecture and API's that will be available in the Admin Panel.

2 Concept

Admin Panel is an ASP.net Web site developed with C# language. The Panel integrated functionality of Login, Logout, Change Password, Forgot Password, User Management, Role Management. Products or Projects can use this panel for developing there Admin Site. The consumer will have to develop a Header/ Footer/ Menu less site which calls Admin Panel Authentication and Authorization API's and opens the project/ products admin module pages in IFRAME.

The integration is seamless and carried out using set of API's exposed. The Product/ Project can create there module wise menu by creating a XML file with details explained further in the document. Super admin will have the rights to manage modules, create users, create roles and map roles to users which make it a complete authentication and authorization system.

Product/ Project will have to deploy there admin modules under admin panel as a virtual directory.

3 API

3.1 REST API's

Admin Panel exposes API's for integration of products/ projects with admin panel. These are Rest based API's with XML output. Consumer needs to pass collection of cookies to Admin Panel Rest API along with API specific parameters.

3.1.1 Authentication

3.1.1.1 Authenticate

- Summary
  - This API provides authentication functionality to the project/ product which will use the admin panel. Consumer will have to call the Authenticate API by passing the cookie collection. The API internally identifies the required cookie and does the necessary authentication. If no or invalid cookie is found then the API returns an error.
- URI
  - `Authenticate`
- Method
  - GET
- URL Request Parameters
    - NA

- POST Request Body Stream
  - NA
- Output:
  - XML

```
<response success="true">
      <id>User ID</id>
      <name><![CDATA[User Name]]></name>
</response>
```

- Exception:
  - Refer Appendix I

3.1.1.2 Authorize

- Summary
  - This helps the consumer to Authorize the user for particular page or action.
- URI
  - `Authorize`
- Method
  - POST
- URL Request Parameters
  - NA
- POST Request Body Stream

```
<authorize>
      <type> 1: URL, 2: Activity Code</type>
      <value><![CDATA[URL or Activity Code value as mentioned in the Module
            Configuration file ]]></value>
</authorize>
```

- Output:
  - XML

```
<response success="true">
      <id>User ID</id>
      <name><![CDATA[User Name]]></name>
</response>
```

- Exception:
  - Refer Appendix I

3.1.1.3 Page Event Authorization

- Summary

- This helps the consumer to get all the Page Event Availability status.

- URI
  - `page/events`
- Method
  - POST
- URL Request Parameters
  - NA
- POST Request Body Stream

```
<authorize>
      <type> 1: URL, 2: Activity Code</type>
      <value><![CDATA[URL or Activity Code value as mentioned in the Module
             Configuration file ]]></value>
</authorize>
```

- Output:
  - XML

```
<response success="true">
      <activities>
            <activity>
                  <code></code>
                  <status>true/false</status>
            </activity>
            <activity>
                  <code></code>
                  <status>true/false</status>
            </activity>
            <activity>
                   …
            </activity>
</response>
```

- Exception:
  - Refer Appendix I

3.1.1.4 Get User List

- Summary
  - This API provides list of admin users. If a particular admin user details are required then we need to pass the user id of the admin user or else pass 0 for all the admin user list.
- URI
  - `list/user/{userid}`
- Method

- POST

- URL Request Parameters
  - userid
    - Admin User ID
    - Integer
    - Pass “0” for all the users
    - Pass specific User ID if need specific user details
- POST Request Body Stream
    - NA
- Output:
  - XML

```
<response success="true">
      <users>
            <user>
                  <id>User ID</id>
                  <name><![CDATA[User Name]]></name>
                  <email>user Email ID</email>
                  <company><![CDATA[User Company Details]]></company>
                  <mobile>Mobile Number</mobile>
                  <roleid>Role ID</roleid>
                  <rolename><![CDATA[Name of the Role]]></rolename>
            </user>
            <user>
                  … … …
            </user>
      </users>
</response>
```

- Exception:
  - Refer Appendix I

4 Web Site

Admin panel web site is a solution for building an admin panel for any project/ product. To successfully integrate the project/ product with admin panel, consumer needs to follow certain steps and instructions. We have already detailed down the API's required for the Authentication and Authorization functionality.

This section describes the deployment and configurations required to setup the Admin Panel and using it with your project/ product.

4.1 Web.config

Web.config setting for admin panel

```
<appSettings>
        <!-- Default Page Size for Admin Listing Pages -->
        <add key="PageSize" value="10"/>
        <!-- Authentication Key -->
        <add key="AUTHKEY" value="A1X2I3S4"/>
        <!-- Auth Cookie Session Timeout-->
        <add key="sessionTimeOutMinutes" value="14400"/>
        <!-- Auth Cookie Name-->
        <add key="UserContextCookieName" value="UserContext"/>
        <!-- Super Admin Default Page-->
        <add key="SAHomePage" value="/User/ManageUser.aspx"/>

        <!-- Administrative Site Base URI -->
        <add key="AdminBaseURL" value="http://admin.xyz.com"/>
        <!-- Login Page URL-->
        <add key="LoginURL" value="http://admin.xyz.com/login.aspx"></add>
        <!-- Unauthorized Page URL-->
        <add key="UnauthorizedURL"
value="http://admin.xyz.com/landingpages/Unauthorized.aspx"></add>

        <!-- Common Page Title Prefix-->
        <add key="pageTitlePrefix" value="CLEO Content Management System"/>
        <!-- Error Mail Settings-->
        <add key="SendErrorMails" value="true"/>
        <!-- Module XML file Common Folder Path-->
        <add key="ModuleFolderPath" value="E:\Cleo-Admin\settings\Config"/>
        <!-- Setting path Email and other configs-->
        <add key="SettingPath" value="E:\ Cleo-Admin\settings\"/>
</appSettings>

<MailSettings>
        <!-- SMTP/ Exchange Server IP or Host Name-->
        <add key="DefaultServer" value="192.168.32.134" />
        <!-- Port Number-->
        <add key="DefaultPort" value="25" />
        <!-- DeliveryMethod = 'iis' for smtp and 'network' for exchange server-->
        <add key="DefaultDeliveryMethod" value="iis" />
        <!-- true if credentials are required-->
        <add key="UseDefaultCredentials" value="false" />
        <!-- User name in case credentials are required-->
        <add key="SMTPUser" value=" " />
        <!-- Password in case credentials are required-->
        <add key="SMTPPassword" value=" " />
        <!-- if Alternate server not available then provide the default settings for
Alternate Server as well-->
        <!-- SMTP/ Exchange Server IP or Host Name-->
        <add key="AlternateServer" value="" />
```

```
        <!-- Port Number-->
        <add key="AlternatePort" value="25" />
        <!-- DeliveryMethod = 'iis' for smtp and 'network' for exchange server-->
        <add key="AlternateDeliveryMethod" value="iis" />
        <!-- true if credentials are required-->
        <add key="AlternateUseDefaultCredentials" value="false" />
        <!-- User name in case credentials are required-->
        <add key="AlternateSMTPUser" value=" " />
        <!-- Password in case credentials are required-->
        <add key="AlternateSMTPPassword" value=" " />
        <!--Email Content file [this is not required]-->
        <add key="ContentFile" value=" c:\EmailBody.xml " />
        <!-- Enable Ssl-->
        <add key="EnableSsl" value="true" />
</MailSettings>
<connectionStrings>
        <add name="ConnectionString" connectionString="Data Source=<IP/ DNS>;Initial
        Catalog=<DB Name>;User ID=<UserID>;Password=<Password>;"
        providerName="SQLServer"/>
</connectionStrings>
```

5 Appendix

- **Appendix I**
  - Exception messages are return in following format
    - XML

```
<response success="false">
      <code>Error Code</code>
      <message><![CDATA[Error Message]]></message>
      <redirecturl><![CDATA[Action URL]]></redirecturl>
</response>
```

  - Error Codes

Error Code	Description
-1	You are not logged In.
-2	You are not authorized.
-3	Error while retrieving user list

- **Appendix II**
  - Module configuration file format

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Codes for each should be unique and should follow the following naming conventions-->
<!-- Module: Research-->
<!-- Modile-> Name-> RES -->
  <!-- Page: add.aspx -->
  <!-- Page-> Name-> RES-ADD-->
    <!-- Activity: publish now -->
    <!-- Activity-> Name-> RES-ADD-PUBLISHNOW-->

<!--------------------------------------------
Data Limit:
  code: Varchar[100]
  name: Varchar[100]
  desc: Varchar[500]
  ismenu: Bit
  url: Varchar[500]
-------------------------------------------->
<modules>
  <module>
    <code><![CDATA[]]></code>
    <name><![CDATA[]]></name>
    <desc><![CDATA[]]></desc>
    <!-- ismenu 0: not in menu/ 1: in menu -->
    <ismenu></ismenu>
    <!-- isdelete 0: initial setting while adding a module/ 1: to delete a module with all pages and activities -->
    <isdelete></isdelete>
    <pages>
      <page>
        <code><![CDATA[]]></code>
        <name><![CDATA[]]></name>
        <desc><![CDATA[]]></desc>
        <!-- url of the page that needs to be added to the menu -->
        <url></url>
        <!-- ismenu 0: not in menu/ 1: in menu -->
        <!-- 0: if the page is not part of the menu navigation -->
        <!-- this setting will be ignored if the module is marked as not in menu [0] -->
        <ismenu></ismenu>
        <!-- this is required if control level authorization is required in the page -->
        <!-- isdelete 0: initial setting while adding a page/ 1: to delete a page with all activities -->
        <isdelete></isdelete>

        <activities>
          <adctivity>
            <code><![CDATA[]]></code>
```

```
            <name><![CDATA[]]></name>
            <desc><![CDATA[]]></desc>
```

```
          <!-- this is required if control level authorization is required in the page -->
          <!-- isdelete 0: initial setting while adding a activity/ 1: to delete the activity -->
         </adctivity>
        </activities>
      </page>
    </pages>
  </module>
</modules>
```