CO3 SYSTEMS  
ASP.NET Incident Creation Application – Installation and Configuration Guide

Version 1.0

Installation and Configuration Guide

This is an example website to demonstrate how users can create new incidents using the Co3 Systems REST API and ASP.net.

# Prerequisites

The DLLs are compiled using .Net version 4.5. If your environment uses a different version of .Net, please recompile the project or contact us for assistance.

# Installation

You can extract all files in the WebSite folder into the root folder of an existing ASP.net enabled IIS website. Alternatively, you can create a new ASP.net website.  
Configuration

The Web.config file is renamed to Web.config-sample to avoid accidental overwrite of the existing Web.config file. You can rename it back to Web.config if you're creating a new website, or copy its contents to the existing Web.config.

CreateIncident.aspx is the only page in the web app. Before you can open this page with a browser, you need to supply your Co3 credentials in the Web.config file.

Web.Config

Below is the list of Web.config values used by CreateIncident.aspx.

|  |  |  |
| --- | --- | --- |
| **Setting** | **Required?** | **Description** |
| Co3ApiUrl | Yes | The Co3 REST API’s base URL. |
| Co3UserAccount | Yes | The Co3 user credential used to access the REST API. The user account must only belong to one org. |
| Co3UserPassword | Yes |
| Co3ProxyUser | No | If you have a HTTP proxy, specify the proxy credentials to allow the web application to access the REST API. Note that you’ll also need to add the following to your Web.config:  <system.net>  <defaultProxy>  <proxy usesystemdefault="true" proxyaddress="http://proxy\_url" />  </defaultProxy>  </system.net>  Be sure to replace proxy\_url with the correct value. |
| Co3ProxyPassword | No |
| Co3ProxyDomain | No |
| Co3CssRequired | No | If specified, this CSS class is added to fields to indicate that they are required. |
| Co3CssInvalid | No | If specified, this CSS class is added to fields with invalid data during postback. |

Customization

The CreateIncident page consists of the ASPX file and its CS corresponding Code-behind. CreateIncident.cs has most of the logic to connect to the Co3 REST API with the controls on the ASPX page. You can do most if not all of the customization by editing the ASPX file. Below are some of the actions you can take to adapt the page to your needs.

JavaScripts

The web application does not rely on JavaScripts to function properly. You can optionally add JavaScripts to the page to improve ease of use and page effects. CreateIncident.aspx has an example of how to add date or date time pickers to the controls using jQuery.

Master File

CreateIncident.aspx is a standalone ASPX page. If your site uses master files, you can convert CreateIncident.aspx to support master files by doing the following:

* Add the MasterPageFile attribute to the <%@Page> tag.
* Remove the <html> and <head> tags.
* Move the style and script links to the appropriate location of your website.
* Replace the <body> tag with the <asp:Content> tag.

Web Application

If your website was created as a web project, where the Code-behind files are compiled into a DLL, and you wish to incorporate this example, you can do so by editing CreateIncident.aspx and changing the CodeFile attribute in the <%@Page> element to CodeBehind (i.e. CodeBehind="CreateIncident.aspx.cs").

Incident Submission and Subsequent Redirection

The server side code at the bottom of the ASPX file controls the postback and redirection behaviors of the page. The submit button click event is registered in OnInit(). The click handler is implemented in Co3Submit\_Click(). If the incident is created, the user will be redirected to the url specified in the click handler.

Notice that you can specify preset values in the click handler. This allows you to save user specific information with the incident without needing the user to provide it.

Available fields

You can only use controls from the System.Web.UI.WebControls namespace. Each field must use a control that supports the field’s data type. The page will display an error if there is a data type mismatch.

Each field is identified uniquely by its designated ID as listed below. DropDownList, RadioButtonList and CheckBoxList controls will be automatically populated with data specific to the fields they represent.

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Allowable ASP.net Types** | **Description** |
| co3\_addr | TextBox | Physical location of the incident, if applicable |
| co3\_hard\_liability | TextBox | Assessed Liability |
| co3\_city | TextBox | City |
| co3\_country | DropDownList  RadioButtonList | Country |
| co3\_crimestatus\_id | DropDownList  RadioButtonList | Criminal Activity |
| co3\_data\_compromised | DropDownList  RadioButtonList | Whether sensitive or personal data was foreseeably exposed and/or compromised. A value of "Yes" or "Unknown" indicate that a breach response may be required. |
| co3\_data\_encrypted | DropDownList  RadioButtonList | Whether the data in question was encrypted |
| co3\_data\_format | DropDownList  RadioButtonList | Specify the format of the personal information involved |
| co3\_discovered\_date | TextBox | Date the incident was discovered/reported - this is the date upon which most reporting/action timelines are based, so it is important to ensure accuracy for this field |
| co3\_start\_date | TextBox | Date the incident occurred |
| co3\_exposure\_dept\_id | DropDownList  RadioButtonList | Department |
| co3\_description | TextBox | A free form text description of the incident |
| co3\_employee\_involved | DropDownList  RadioButtonList | Employee Involved |
| co3\_data\_contained | DropDownList  RadioButtonList | Whether the exposure has been addressed and rectified |
| co3\_exposure\_type\_id | DropDownList  RadioButtonList | Origin source of the exposure |
| co3\_harmstatus\_id | DropDownList  RadioButtonList | Harm Foreseeable |
| co3\_confirmed | DropDownList  RadioButtonList | Tag an issue as an unconfirmed (event) vs a confirmed incident |
| co3\_incident\_type\_ids | CheckBoxList | The type of incident |
| co3\_exposure\_individual\_name | TextBox | Individual Name |
| co3\_jurisdiction\_name | TextBox | Jurisdiction |
| co3\_name | TextBox | A unique name to identify this particular incident |
| co3\_negative\_pr\_likely | DropDownList  RadioButtonList | If it is foreseeable that the incident might generate any negative public image or publicity for your company or organization. |
| co3\_nist\_attack\_vectors | CheckBoxList | NIST Attack Vectors the incident falls under |
| co3\_phase\_id | DropDownList  RadioButtonList | The phase of the incident |
| co3\_postal\_code | TextBox | Postal Code |
| co3\_province | DropDownList  RadioButtonList | Canadian provinces |
| co3\_reporter | TextBox | Name of person who reported the event, such as a device owner or his/her manager |
| co3\_resolution\_id | DropDownList  RadioButtonList | Select an option that accurately describes the reason for closing this incident. |
| co3\_resolution\_summary | TextBox | Enter a summary which describes how this incident was resolved. |
| co3\_severity\_code | DropDownList  RadioButtonList | Your impression of the events relative severity vs. other events that may be entered into the system |
| co3\_inc\_training | DropDownList  RadioButtonList | Whether the incident is a simulation or a regular incident. This field is read-only. |
| co3\_data\_source\_ids | CheckBoxList | Original source of the data, such as the name of the database |
| co3\_state | DropDownList  RadioButtonList | United States states |
| co3\_plan\_status | DropDownList  RadioButtonList | Status |
| co3\_exposure\_vendor\_id | DropDownList  RadioButtonList | Vendor |
| co3\_zip | TextBox | Zip (or postal) code of the location of the incident |

Custom Fields

Custom fields can be added using their corresponding System.Web.UI.WebControl. The control’s ID is the API Access Name prefixed with “co3\_”. For example, if the API Access Name is “employee\_id”, then the control’s ID is “co3\_employee\_id”.

|  |  |
| --- | --- |
| **Custom Field Type** | **Allowable ASP.net Types** |
| Date Picker | TextBox |
| Date Time Picker | TextBox |
| Text | TextBox |
| Number | TextBox |
| Text Area | TextBox |
| Select | DropDownList |
| RadioButtonList | Boolean |
| DropDownList | RadioButtonList |
| Multiselect | CheckBoxList |

Troubleshooting

If an error occurs or an exception is thrown, the Code-behind will invoke the error handling function with the following signature:

protected override void OnHandleException(Exception ex, string message)

You can implement this method to display the error message. The ASPX file implements this method to provide a generic error message. You can expand this method to provide better error handling, or remove it.