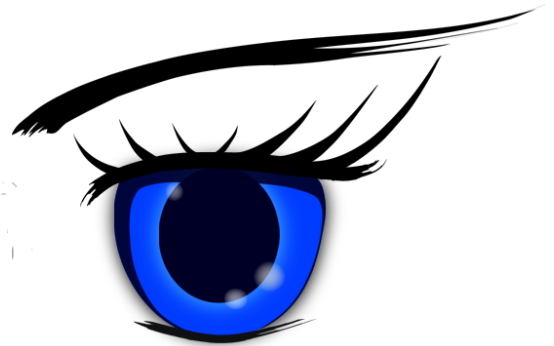


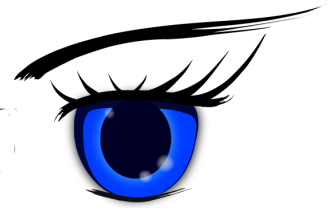
EYECATCH



My First Umbraco Datatype

Until now I have been using Doc2Form as a comment/contact form on my blog. Recently however, I have started to get some spam comments (Probably because of the increased visitor rate after my "My First Umbraco Usercontrol" article). I'm not very fond of spam, so I decided to implement a reCAPTCHA control for use with Doc2Form.

Ove Andersen
12.05.2010

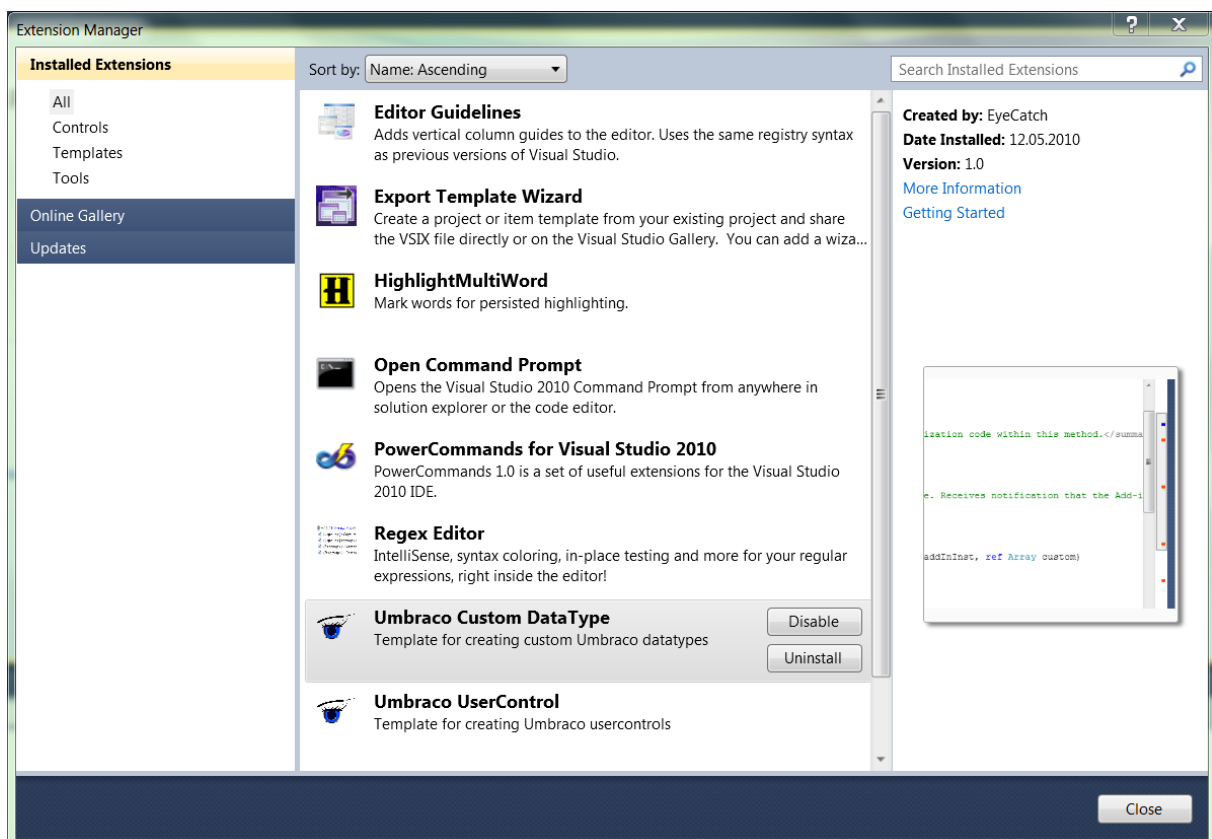


Part 1a: Setting up Visual Studio 2010

First, we need to create a new Web Application. To ease the process of setting up the datatype in Visual Studio 2010, I created a template which you can download [here](#).

First, install the template you just downloaded by double-clicking on it. If the file has a .zip extension, rename it to .vsix.

Then create a new Web Application by clicking “File” -> “New Project” -> “Visual C#” -> “Umbraco Abstract Datatype”. You can name the datatype whatever you like. My project is called “EyeCatchReCaptcha”.

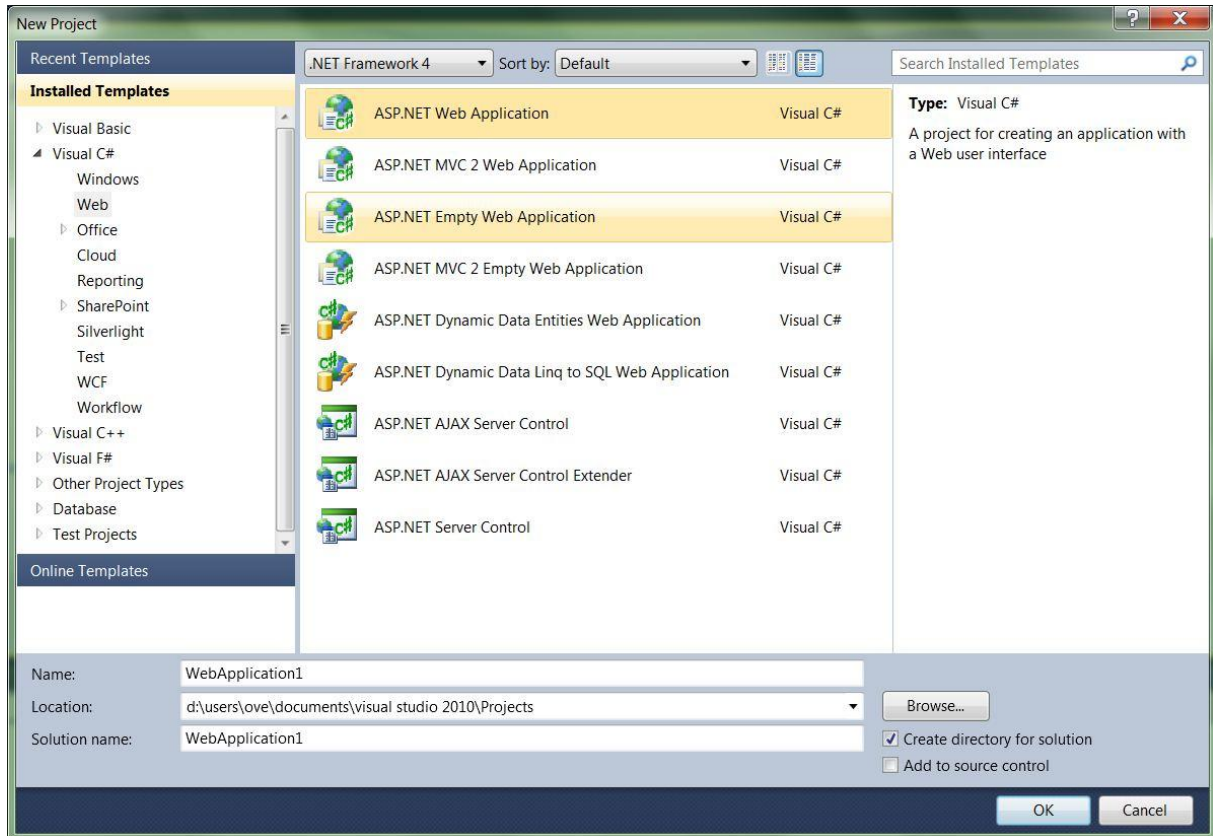


2

Now you are ready to create your first custom Umbraco datatype!

Part 1b: Setting up Visual Studio 2008/2010

First, create a new Web Application (You can find it by clicking “New Project” -> “ASP.NET Web Application”). You can call it whatever you like.



3

Then you need to copy some files from Umbraco to your “bin” folder. Not all of the following files are necessary, but I included them just in case. [You can find a zip with the files here.](#)

- umbraco.dll
- umbraco.editorControls.dll
- umbraco.DataLayer.dll
- cms.dll
- interfaces.dll
- businesslogic.dll

You also need a copy of reCAPTCHA, [which you can find here.](#)

You also need to add references to these files. You can do this by right- clicking the “References” folder and press “Add Reference”. Then you go to the “Browse” pane, and locate the .dll files you just copied.



Part 2: Rendering a reCAPTCHA control

The goal of this post is to render the datatype on a page using Doc2Form.

EyeCatchReCaptcha.cs

```
using System;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace EyeCatchReCaptchaDatatype
{
    /// <summary>
    /// This class is used for the actual datatype dataeditor, i.e. the control you will
    get in the content section of umbraco.
    /// </summary>
    public class EyeCatchReCaptcha : Panel
    {
        // DataType Controls
        private readonly Recaptcha.RecaptchaControl _recaptcha = new
Recaptcha.RecaptchaControl();

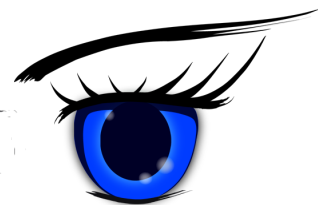
        // Datatype Configuration
        ///<summary>
        ///</summary>
        public string PrivateKey { get; set; }
        ///<summary>
        ///</summary>
        public string PublicKey { get; set; }
        ///<summary>
        ///</summary>
        public string Theme { get; set; }

        /// <summary>
        /// Raises the <see cref="E:System.Web.UI.Control.Init"/> event.
        /// </summary>
        /// <param name="e">An <see cref="T:System.EventArgs"/> object that contains the
event data.</param>
        protected override void OnInit(EventArgs e)
        {
            base.OnInit(e);

            _recaptcha.Theme = this.Theme;
            _recaptcha.PrivateKey = this.PrivateKey;
            _recaptcha.PublicKey = this.PublicKey;

            this.Controls.Add(_recaptcha);
            this.Controls.Add(new LiteralControl("&nbsp;"));
        }
    }
}
```

4



DataType.cs

```

using System;
using System.Collections;
using System.Collections.Generic;
using umbraco.cms.businesslogic.datatype;

namespace EyeCatchReCaptchaDatatype
{
    /// <summary>
    /// This is basicly where you give the datatype it's name and unique id.
    /// The name will show up in the rendercontrol dropdown of the edit datatype page.
    /// </summary>
    public class DataType : AbstractDataEditor
    {
        private PrevalueEditor _prevalueeditor;

        // Instance of the Datatype
        private EyeCatchReCaptcha control = new EyeCatchReCaptcha();

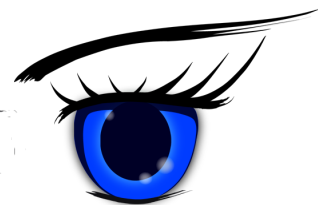
        /// <summary>
        /// Initializes a new instance of the <see cref="EyeCatchReCaptcha"/> class.
        /// </summary>
        public DataType()
        {
            base.RenderControl = control;
            control.Init += new EventHandler(ControlInit);
            base.DataEditorControl.OnSave += DataEditorControl_OnSave;
        }

        /// <summary>
        /// Handles the Init event of the control control.
        /// </summary>
        /// <param name="sender">The source of the event.</param>
        /// <param name="e">The <see cref="System.EventArgs"/> instance containing the
        event data.</param>
        private void ControlInit(object sender, EventArgs e)
        {
            // Get configuration
            SortedList s = ((PrevalueEditor) PrevalueEditor).StoredConfiguration;

            // Set configuration
            if (s.ContainsKey("privatekey"))
            {
                control.PrivateKey = s.GetByIndex(s.IndexOfKey("privatekey")).ToString();
            }

            if (s.ContainsKey("publickey"))
            {
                control.PublicKey = s.GetByIndex(s.IndexOfKey("publickey")).ToString();
            }
        }
    }
}

```



```
/// <summary>
/// </summary>
/// <param name="e">The <see cref="System.EventArgs"/> instance containing the
event data.</param>
private static void DataEditorControl_OnSave(EventArgs e)
{
    // We don't need to do anything here
}

/// <summary>
/// Gets the datatype unique id.
/// </summary>
/// <value>The id.</value>
public override Guid Id
{
    get
    {
        return new Guid("947ceafb-a19f-468a-8a57-bf9ef57e9a52");
    }
}

/// <summary>
/// Gets the name of the data type.
/// </summary>
/// <value>The name of the data type.</value>
public override string DataTypeName
{
    get
    {
        return "[EyeCatch] reCAPTCHA";
    }
}

/// <summary>
/// Gets the prevalue editor.
/// </summary>
/// <value>The prevalue editor.</value>
public override umbraco.interfaces.IDataPrevalue PrevalueEditor
{
    get
    {
        if (_prevalueeditor == null) {
            _prevalueeditor = new PrevalueEditor(this);
        }
        return _prevalueeditor;
    }
}
}
}
```



PrevalueEditor.cs

```
using System;
using System.Collections;
using System.Web.UI;
using System.Web.UI.WebControls;
using umbraco.BusinessLogic;
using umbraco.cms.businesslogic.datatype;
using umbraco.DataLayer;
using umbraco.interfaces;

namespace EyeCatchReCaptchaDatatype
{
    /// <summary>
    /// This class is used to setup the datatype settings.
    /// On save it will store these values (using the datalayer) in the database
    /// </summary>
    public class PrevalueEditor : Placeholder, IDataPrevalue
    {
        #region IDataPrevalue Members

        // Referenced datatype
        private static BaseDataType _datatype;

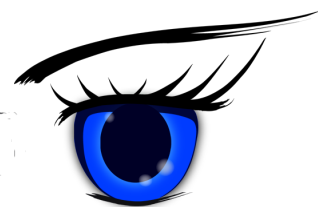
        // Datatype Controls
        private TextBox _privateKey;
        private TextBox _publicKey;

        /// <summary>
        /// Initializes a new instance of the <see cref="PrevalueEditor"/> class.
        /// </summary>
        /// <param name="dataType">Type of the data.</param>
        public PrevalueEditor(DataType dataType)
        {
            _datatype = dataType;
            SetupChildControls();
        }

        /// <summary>
        /// Sets up the child controls.
        /// Creates new instances of the Datatype Controls
        /// </summary>
        private void SetupChildControls()
        {
            _privateKey = new TextBox { ID = "txtPrivateKey", Text = "", Width = 250};
            _publicKey = new TextBox { ID = "txtPublicKey", Text = "", Width = 250 };

            Controls.Add(_privateKey);
            Controls.Add(_publicKey);
        }
    }
}
```

7



```
/// <summary>
/// Gets the editor.
/// </summary>
/// <value>The editor.</value>
public Control Editor
{
    get
    {
        return this;
    }
}

/// <summary>
/// Raises the <see cref="E:System.Web.UI.Control.Load"/> event.
/// </summary>
/// <param name="e">The <see cref="T:System.EventArgs"/> object that contains the
event data.</param>
protected override void OnLoad(EventArgs e)
{
    base.OnLoad(e);

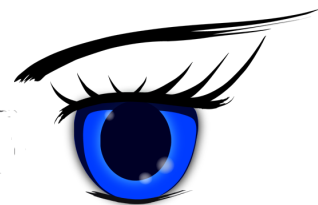
    // Sets the configuration properties from the Configuration function
    if (!Page.IsPostBack)
    {
        SortedList s = StoredConfiguration;

        if (s.ContainsKey("privatekey"))
        {
            _privateKey.Text = s.GetByIndex(s.IndexOfKey("privatekey")).ToString();
        }

        if (s.ContainsKey("publickey"))
        {
            _publicKey.Text = s.GetByIndex(s.IndexOfKey("publickey")).ToString();
        }
    }
}

/// <summary>
/// Saves the prevalues to the Umbraco database.
/// </summary>
public void Save()
{
    _datatype.DBType = (DBTypes)Enum.Parse(typeof(DBTypes),
DBTypes.Ntext.ToString(), true);

    // Deletes former values
    SqlHelper.ExecuteNonQuery("delete from cmsDataTypePreValues where
datatypeid = @dtdefid", SqlHelper.CreateParameter("@dtdefid",
_datatype.DataTypeDefinitionId));
}
```



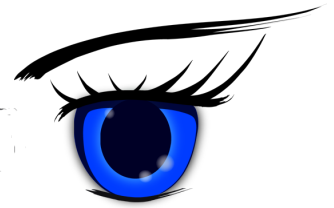
```
// Inserts the current values
    SqlHelper.ExecuteNonQuery("insert into cmsDataTypePreValues
(datatypenodeid,[value],sortorder,alias) values (@dtdefid,@value,0,'privatekey')",
SqlHelper.CreateParameter("@dtdefid", _datatype.DataTypeDefinitionId),
SqlHelper.CreateParameter("@value", _privateKey.Text));
    SqlHelper.ExecuteNonQuery("insert into cmsDataTypePreValues
(datatypenodeid,[value],sortorder,alias) values (@dtdefid,@value,0,'publickey')",
SqlHelper.CreateParameter("@dtdefid", _datatype.DataTypeDefinitionId),
SqlHelper.CreateParameter("@value", _publicKey.Text));
}

/// <summary>
/// Sends server control content to a provided <see
cref="T:System.Web.UI.HtmlTextWriter"/> object,
/// which writes the content to be rendered on the client.
/// </summary>
/// <param name="writer">The <see cref="T:System.Web.UI.HtmlTextWriter"/> object
that receives the server control content.</param>
protected override void Render(HtmlTextWriter writer)
{
    writer.WriteLine("<table>");

    writer.Write("<tr><th>Private API Key:</th><td>");
    _privateKey.RenderControl(writer);
    writer.Write("</td><td rowspan='2'><small>");
    writer.Write("<a href='https://www.google.com/recaptcha/admin/create'
target='_blank'>Get API keys (You need a Google account)</a>");
    writer.Write("</small></td></tr>");

    writer.Write("<tr><th>Public API Key:</th><td>");
    _publicKey.RenderControl(writer);
    writer.Write("</td></tr>");

    writer.Write("</table>");
}
}
```



```

/// <summary>
/// Gets the Prevalues from the database.
/// </summary>
/// <value>The configuration.</value>
public SortedList StoredConfiguration
{
    get
    {
        // Read values from the database
        IRecordsReader rr = SqlHelper.ExecuteReader("select alias, value from
cmsDataTypePreValues where datatypenodeid =
@datatypenodeid",SqlHelper.CreateParameter("@datatypenodeid",
_datatype.DataTypeDefinitionId));

        // Array for stored configuration values
        SortedList values = new SortedList();

        // Go through the values from the database and store them in the array.
        while (rr.Read())
        {
            values.Add(rr.GetString("alias"), rr.GetString("value"));
        }

        return values;
    }
}

#endregion

/// <summary>
/// Gets the Umbraco SQL helper.
/// </summary>
/// <value>The SQL helper.</value>
public static ISqlHelper SqlHelper
{
    get
    {
        return Application.SqlHelper;
    }
}
}
}

```

10

Build the project and upload EyeCatchReCaptcha.dll and Recaptcha.dll to your bin- folder.



In this tutorial, we are going to create a comment form. You also need to [download and install Doc2Form](#) if you haven't done this already.

Then add a new Datatype with the name "reCAPTCHA" in the Developer section and select "[EyeCatch] reCAPTCHA" as a render control. Set the Private Key and Public Key to your own keys. Click on the link to the right of the text boxes to get to the registration page.

| | |
|----------------|----------------------|
| Name | reCAPTCHA |
| Render control | [EyeCatch] reCAPTCHA |

Settings

| | | |
|------------------|--|--|
| Private API Key: | 6Ld_A7oSAAAAAGmMcLgU3WA9J5spbY10MsKpgilp | Get API keys (You need a Google account) |
| Public API Key: | 6Ld_A7oSAAAAAFZl34g37hQ09jJLqjsNxDPkDb4F | |

Create a new Document Type with the name "Comment" and add the following properties:

Tab: Content

| | | |
|---|---|---|
| Name (name), Type: Textstring | ▼ | ✕ |
| E-Mail (email), Type: Textstring | ▼ | ✕ |
| Website (website), Type: Textstring | ▼ | ✕ |
| Message (message), Type: Textbox multiple | ▼ | ✕ |
| CAPTCHA (CAPTCHA), Type: reCAPTCHA | ▼ | ✕ |

11

Now select your template and add a "Doc2Form – Form" macro. Select your document type in the first dropdown, and then select the name of the tab(s) where your comment properties are located.

You can choose where to store the comments with the "Save or Edit node location". I chose to save it under the current page by setting the value to "[#pageID]" after the macro was inserted. I also set "PublishOnSubmit" to true.

Remember to allow the new document type under the document type of the current page.

The template should look something like this:


```
<umbraco:Macro DocumentType="1130" Template="" TabName="content" PageTabs="0"
HideTabNames="1" ChooseWhereToStore="#"[#pageID]" EditMode="0" ShowTitle="0"
TitleName="" SaveMemberAlias=", " ShowDescriptions="1" RequiredText=""
SubmitButtonText="Submit" PreviousButtonText="" NextButtonText="" TextOnSubmit=""
PublishOnSubmit="1" RefreshToParent="0" RedirectToNode="#"[#pageID]"
PublishWithUserId="12" StorePropertiesInCookies="" UseAjax="0" DefaultValueNode=""
Alias="Doc2FormInput" runat="server"></umbraco:Macro>
```

12.05.2010

Org. No: NO 991 157 328 MVA
Phone: +47 99 01 66 10

Address: Straume Teknologisenter
v/ 3D Design Media
Trollhaugmyra 15
5353 Straume
Norway

Finally, the resulting web page should render [a reCAPTCHA control like here](#) :
(Click the "Leave Comment" link to see the form.)

 [Leave comment](#)

Name


E-Mail

Not visible


Website

Message

CAPTCHA



Type the two words:

 **reCAPTCHA™**
stop spam.
read books.



Part 3: Extending the plugin

Now we have a working comment form with reCAPTCHA validation. To make the datatype more flexible, we can add support for theme selection and custom translations. So we will make some changes the source files.

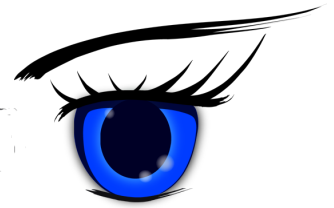
EyeCatchReCaptcha.cs

```
using System;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace EyeCatchReCaptchaDatatype
{
    /// <summary>
    /// This class is used for the actual datatype dataeditor, i.e. the control you will
    get in the content section of umbraco.
    /// </summary>
    public class EyeCatchReCaptcha : Panel
    {
        // DataType Controls
        private readonly Recaptcha.RecaptchaControl _recaptcha = new
Recaptcha.RecaptchaControl();

        // Datatype Configuration
        ///<summary>
        ///</summary>
        public string PrivateKey { get; set; }
        ///<summary>
        ///</summary>
        public string PublicKey { get; set; }
        ///<summary>
        ///</summary>
        public string Theme { get; set; }
        ///<summary>
        ///</summary>
        public string Language { get; set; }
        ///<summary>
        ///</summary>
        public string CustomThemeWidget { get; set; }
        ///<summary>
        ///</summary>
        public string CustomTranslations { get; set; }
    }
}
```

13



```
/// <summary>
/// Raises the <see cref="E:System.Web.UI.Control.Init"/> event.
/// </summary>
/// <param name="e">An <see cref="T:System.EventArgs"/> object that contains the
event data.</param>
protected override void OnInit(EventArgs e)
{
    base.OnInit(e);

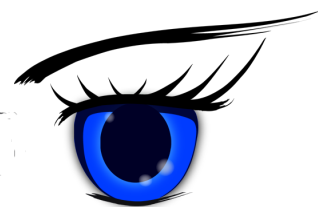
    _recaptcha.Theme = this.Theme;
    _recaptcha.PrivateKey = this.PrivateKey;
    _recaptcha.PublicKey = this.PublicKey;

    if (!Language.Equals(""))
    {
        _recaptcha.Language = this.Language;
    }

    if (!CustomTranslations.Equals(""))
    {
        _recaptcha.CustomTranslations = this.CustomTranslations;
    }

    if (!CustomThemeWidget.Equals(""))
    {
        _recaptcha.CustomThemeWidget = this.CustomThemeWidget;
    }

    this.Controls.Add(_recaptcha);
    this.Controls.Add(new LiteralControl("&nbsp;"));
}
}
```



DataType.cs

```

using System;
using System.Collections;
using System.Collections.Generic;
using umbraco.cms.businesslogic.datatype;

namespace EyeCatchReCaptchaDatatype
{
    /// <summary>
    /// This is basicly where you give the datatype it's name and unique id.
    /// The name will show up in the rendercontrol dropdown of the edit datatype page.
    /// </summary>
    public class DataType : AbstractDataEditor
    {
        private PrevalueEditor _prevalueeditor;

        // Instance of the Datatype
        private EyeCatchReCaptcha control = new EyeCatchReCaptcha();

        /// <summary>
        /// Initializes a new instance of the <see cref="EyeCatchReCaptcha"/> class.
        /// </summary>
        public DataType()
        {
            base.RenderControl = control;
            control.Init += new EventHandler(ControlInit);
            base.DataEditorControl.OnSave += DataEditorControl_OnSave;
        }

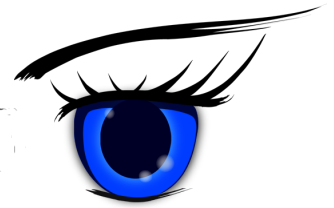
        /// <summary>
        /// Handles the Init event of the control control.
        /// </summary>
        /// <param name="sender">The source of the event.</param>
        /// <param name="e">The <see cref="System.EventArgs"/> instance containing the
        event data.</param>
        private void ControlInit(object sender, EventArgs e)
        {
            // Get configuration
            SortedList s = ((PrevalueEditor) PrevalueEditor).StoredConfiguration;

            // Set configuration
            if (s.ContainsKey("theme"))
            {
                control.Theme = s.GetByIndex(s.IndexOfKey("theme")).ToString();
            }

            if (s.ContainsKey("privatekey"))
            {
                control.PrivateKey = s.GetByIndex(s.IndexOfKey("privatekey")).ToString();
            }

            if (s.ContainsKey("privatekey"))
            {
                control.PublicKey = s.GetByIndex(s.IndexOfKey("publickey")).ToString();
            }
        }
    }
}

```



```
        if (s.ContainsKey("language"))
        {
            control.Language = s.GetByIndex(s.IndexOfKey("language")).ToString();
        }

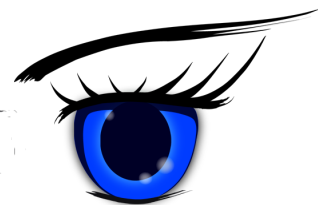
        if (s.ContainsKey("customthemewidget"))
        {
            control.CustomThemWidget =
s.GetByIndex(s.IndexOfKey("customthemewidget")).ToString();
        }

        if (s.ContainsKey("customtranslations"))
        {
            control.CustomTranslations =
s.GetByIndex(s.IndexOfKey("customtranslations")).ToString();
        }
    }

    /// <summary>
    /// </summary>
    /// <param name="e">The <see cref="System.EventArgs"/> instance containing the
event data.</param>
    private static void DataEditorControl_OnSave(EventArgs e)
    {
        // We don't need to do anything here
    }

    /// <summary>
    /// Gets the datatype unique id.
    /// </summary>
    /// <value>The id.</value>
    public override Guid Id
    {
        get
        {
            return new Guid("947ceafb-a19f-468a-8a57-bf9ef57e9a52");
        }
    }

    /// <summary>
    /// Gets the name of the data type.
    /// </summary>
    /// <value>The name of the data type.</value>
    public override string DataTypeName
    {
        get
        {
            return "[EyeCatch] reCAPTCHA";
        }
    }
}
```



```

/// <summary>
/// Gets the prevalue editor.
/// </summary>
/// <value>The prevalue editor.</value>
public override umbraco.interfaces.IDataPrevalue PrevalueEditor
{
    get
    {
        if (_prevalueeditor == null) {
            _prevalueeditor = new PrevalueEditor(this);
        }
        return _prevalueeditor;
    }
}
}
}

```

PrevalueEditor.cs

```

using System;
using System.Collections;
using System.Web.UI;
using System.Web.UI.WebControls;
using umbraco.BusinessLogic;
using umbraco.cms.businesslogic.datatype;
using umbraco.DataLayer;
using umbraco.interfaces;

```

```

namespace EyeCatchReCaptchaDatatype
{

```

```

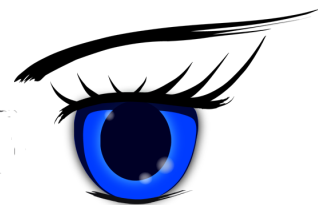
    /// <summary>
    /// This class is used to setup the datatype settings.
    /// On save it will store these values (using the datalayer) in the database
    /// </summary>
    public class PrevalueEditor : Placeholder, IDataPrevalue
    {
        #region IDataPrevalue Members

        // Referenced datatype
        private static BaseDataType _datatype;

        // Datatype Controls
        private DropDownList _themes;
        private TextBox _privateKey;
        private TextBox _publicKey;
        private TextBox _language;
        private TextBox _customThemWidget;
        private TextBox _customTranslations;

        // List of selectable themes.
        // Note: Custom theme doesn't work for now
        private readonly ListItem[] _themeList = { new ListItem("red"), new
        ListItem("white"), new ListItem("blackglass"), new ListItem("clean") };

```



```

/// <summary>
/// Initializes a new instance of the <see cref="PrevalueEditor"/> class.
/// </summary>
/// <param name="dataType">Type of the data.</param>
public PrevalueEditor(DataType dataType)
{
    _datatype = dataType;
    SetupChildControls();
}

/// <summary>
/// Sets up the child controls.
/// Creates new instances of the Datatype Controls
/// </summary>
private void SetupChildControls()
{
    _themes = new DropDownList { ID = "ddlThemes", SelectedValue = "red", Width =
250};
    _themes.Items.AddRange(_themeList);

    _privateKey = new TextBox { ID = "txtPrivateKey", Text = "", Width = 250};
    _publicKey = new TextBox { ID = "txtPublicKey", Text = "", Width = 250 };
    _language = new TextBox { ID = "txtLanguage", Text = "", Width = 250 };

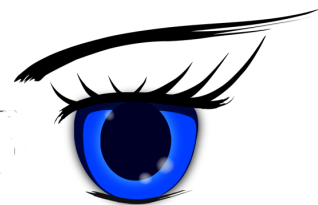
    _customThemWidget = new TextBox { ID = "txtCustomThemWidget", Text = "",
Width = 250 };

    _customTranslations = new TextBox { ID = "txtCustomTranslations", Text = "",
TextMode = TextBoxMode.MultiLine, Rows = 10, Width = 250 };

    Controls.Add(_themes);
    Controls.Add(_privateKey);
    Controls.Add(_publicKey);
    Controls.Add(_language);
    Controls.Add(_customThemWidget);
    Controls.Add(_customTranslations);
}

/// <summary>
/// Gets the editor.
/// </summary>
/// <value>The editor.</value>
public Control Editor
{
    get
    {
        return this;
    }
}

```



```
/// <summary>
/// Raises the <see cref="E:System.Web.UI.Control.Load"/> event.
/// </summary>
/// <param name="e">The <see cref="T:System.EventArgs"/> object that contains the
event data.</param>
protected override void OnLoad(EventArgs e)
{
    base.OnLoad(e);

    // Sets the configuration properties from the Configuration function
    if (!Page.IsPostBack)
    {
        SortedList s = StoredConfiguration;

        if (s.ContainsKey("theme"))
        {
            _themes.SelectedValue = s.GetByIndex(s.IndexOfKey("theme")).ToString();
        }

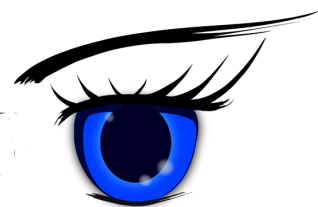
        if (s.ContainsKey("privatekey"))
        {
            _privateKey.Text = s.GetByIndex(s.IndexOfKey("privatekey")).ToString();
        }

        if (s.ContainsKey("publickey"))
        {
            _publicKey.Text = s.GetByIndex(s.IndexOfKey("publickey")).ToString();
        }

        if (s.ContainsKey("language"))
        {
            _language.Text = s.GetByIndex(s.IndexOfKey("language")).ToString();
        }

        if (s.ContainsKey("customthemewidget"))
        {
            _customThemeWidget.Text =
s.GetByIndex(s.IndexOfKey("customthemewidget")).ToString();
        }

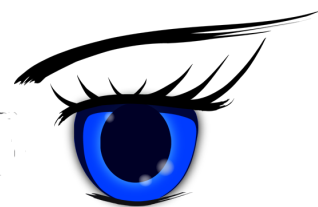
        if (s.ContainsKey("customtranslations"))
        {
            _customTranslations.Text =
s.GetByIndex(s.IndexOfKey("customtranslations")).ToString();
        }
    }
}
```



```
/// <summary>
/// Saves the prevalues to the Umbraco database.
/// </summary>
public void Save()
{
    _datatype.DBType = (DBTypes)Enum.Parse(typeof(DBTypes),
DBTypes.Ntext.ToString(), true);

    // Deletes former values
    SqlHelper.ExecuteNonQuery("delete from cmsDataTypePreValues where
datatypenodeid = @dtdefid", SqlHelper.CreateParameter("@dtdefid",
_datatype.DataTypeDefinitionId));

    // Inserts the current values
    SqlHelper.ExecuteNonQuery("insert into cmsDataTypePreValues
(datatypenodeid,[value],sortorder,alias) values (@dtdefid,@value,0,'theme')",
SqlHelper.CreateParameter("@dtdefid", _datatype.DataTypeDefinitionId),
SqlHelper.CreateParameter("@value", _themes.SelectedValue));
    SqlHelper.ExecuteNonQuery("insert into cmsDataTypePreValues
(datatypenodeid,[value],sortorder,alias) values (@dtdefid,@value,0,'privatekey')",
SqlHelper.CreateParameter("@dtdefid", _datatype.DataTypeDefinitionId),
SqlHelper.CreateParameter("@value", _privateKey.Text));
    SqlHelper.ExecuteNonQuery("insert into cmsDataTypePreValues
(datatypenodeid,[value],sortorder,alias) values (@dtdefid,@value,0,'publickey')",
SqlHelper.CreateParameter("@dtdefid", _datatype.DataTypeDefinitionId),
SqlHelper.CreateParameter("@value", _publicKey.Text));
    SqlHelper.ExecuteNonQuery("insert into cmsDataTypePreValues
(datatypenodeid,[value],sortorder,alias) values (@dtdefid,@value,0,'language')",
SqlHelper.CreateParameter("@dtdefid", _datatype.DataTypeDefinitionId),
SqlHelper.CreateParameter("@value", _language.Text));
    SqlHelper.ExecuteNonQuery("insert into cmsDataTypePreValues
(datatypenodeid,[value],sortorder,alias) values (@dtdefid,@value,0,'customthemewidget')",
SqlHelper.CreateParameter("@dtdefid", _datatype.DataTypeDefinitionId),
SqlHelper.CreateParameter("@value", _customThemeWidget.Text));
    SqlHelper.ExecuteNonQuery("insert into cmsDataTypePreValues
(datatypenodeid,[value],sortorder,alias) values (@dtdefid,@value,0,'customtranslations')",
SqlHelper.CreateParameter("@dtdefid", _datatype.DataTypeDefinitionId),
SqlHelper.CreateParameter("@value", _customTranslations.Text));
}
```



```

/// <summary>
/// Sends server control content to a provided <see
 cref="T:System.Web.UI.HtmlTextWriter"/> object,
/// which writes the content to be rendered on the client.
/// </summary>
/// <param name="writer">The <see cref="T:System.Web.UI.HtmlTextWriter"/> object
that receives the server control content.</param>
protected override void Render(HtmlTextWriter writer)
{
    writer.WriteLine("<table>");

    writer.Write("<tr><th>Theme:</th><td>");
    _themes.RenderControl(writer);
    writer.Write("</td></tr>");

    writer.Write("<tr><th>Private API Key:</th><td>");
    _privateKey.RenderControl(writer);
    writer.Write("</td><td rowspan='2'><small>");
    writer.Write("<a href='https://www.google.com/recaptcha/admin/create'
target='_blank'>Get API keys (You need a Google account)</a>");
    writer.Write("</small></td></tr>");

    writer.Write("<tr><th>Public API Key:</th><td>");
    _publicKey.RenderControl(writer);
    writer.Write("</td></tr>");

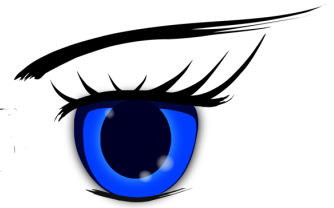
    writer.Write("<tr><th>Language:</th><td>");
    _language.RenderControl(writer);
    writer.Write("</td><td><small>");
    writer.Write("<a href='http://en.wikipedia.org/wiki/List_of_ISO_639-1_codes'
target='_blank'>List of possible ISO 639-1 codes</a>");
    writer.Write("</small><br /><small>");
    writer.Write("<a href='http://recaptcha.net/apidocs/captcha/client.html#look-n-
feel' target='_blank'>List of languages with built-in support (Languages that don't need
custom translation)</a>");
    writer.Write("</small></td></tr>");

    writer.Write("<tr><th>Custom Theme Widget:</th><td>");
    _customThemeWidget.RenderControl(writer);
    writer.Write("</td><td><small>");
    writer.Write("<a href='http://recaptcha.net/apidocs/captcha/client.html#look-n-
feel' target='_blank'>Look here for instructions</a>");
    writer.Write("</small></td></tr>");

    writer.Write("<tr><th>Custom Translations:</th><td>");
    _customTranslations.RenderControl(writer);
    writer.Write("</td><td><small>");
    writer.Write("<a href='http://wiki.recaptcha.net/index.php/I18n'
target='_blank'>Look here for instructions</a>");
    writer.Write("</small></td></tr>");

    writer.Write("</table>");
}

```



```

/// <summary>
/// Gets the Prevalues from the database.
/// </summary>
/// <value>The configuration.</value>
public SortedList StoredConfiguration
{
    get
    {
        // Read values from the database
        IRecordsReader rr = SqlHelper.ExecuteReader("select alias, value from
cmsDataTypePreValues where datatypenodeid =
@datatypenodeid",SqlHelper.CreateParameter("@datatypenodeid",
_datatype.DataTypeDefinitionId));

        // Array for stored configuration values
        SortedList values = new SortedList();

        // Go through the values from the database and store them in the array.
        while (rr.Read())
        {
            values.Add(rr.GetString("alias"), rr.GetString("value"));
        }

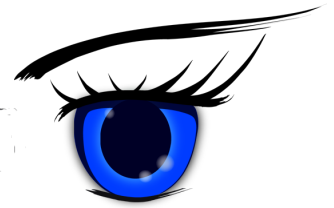
        return values;
    }
}

#endregion

/// <summary>
/// Gets the Umbraco SQL helper.
/// </summary>
/// <value>The SQL helper.</value>
public static ISqlHelper SqlHelper
{
    get
    {
        return Application.SqlHelper;
    }
}
}
}

```

Build the project and re- upload the .dll files to your bin folder.



Now you should have more options in your datatype editor:

| | | |
|----------------------|---|--|
| Theme: | <input type="text" value="white"/> | |
| Private API Key: | <input type="text" value="6Ld_A7o5AAAAAGmMcLgU3WA9J5spb`Y10MskPgilp"/> | Get API keys (You need a Google account) |
| Public API Key: | <input type="text" value="6Ld_A7o5AAAAAFZl34g37hQ09jJLqjsNxDbkDb4F"/> | |
| Language: | <input type="text" value="en"/> | List of possible ISO 639-1 codes List of languages with built-in support (Languages that don't need custom translation) |
| Custom Theme Widget: | <input type="text"/> | Look here for instructions |
| Custom Translations: | <div style="border: 1px solid #ccc; height: 100px; width: 100%;"></div> | Look here for instructions |

Try them out and make comments on my webpage if you experience any problems.