Shield

Umbraco active security modules

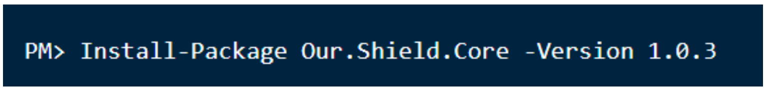
Installation

Installing via NuGet

This Umbraco package can be installed via NuGet

The first part is the Shield framework, which coordinates the different security apps, which can be found here

https://www.nuget.org/packages/Our.Shield.Core/

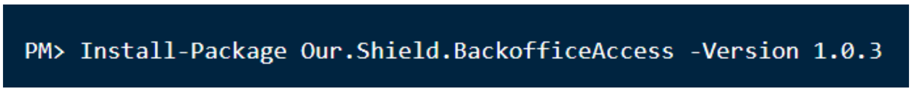


```
PM> Install-Package Our.Shield.Core -Version 1.0.3
```

And the second part is the Shield apps, which provide the active security. Note, there are no restriction on the number of shield apps that can be installed. If you want, install them all using NuGet, to gain the full benefits of what Shield can provide.

- **Backoffice Access**
  Gives you the ability to configure and restrict access to the backoffice access URL.

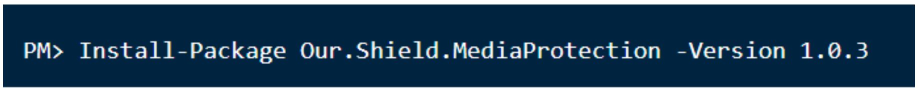
  https://www.nuget.org/packages/Our.Shield.BackofficeAccess

  ```
  PM> Install-Package Our.Shield.BackofficeAccess -Version 1.0.3
  ```

- Media Protection
  Disable Hotlinking and to secure your media to only be accessed by authenticated members.

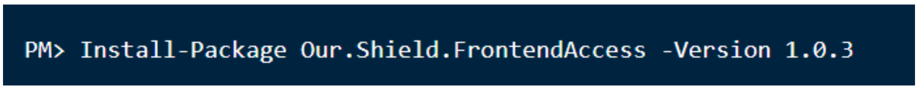
  https://www.nuget.org/packages/Our.Shield.MediaProtection

  ```
  PM> Install-Package Our.Shield.MediaProtection -Version 1.0.3
  ```

- Frontend Access
  Gives you the ability to lock down the frontend to only be accessible by authenticated Umbraco Users and/or restrict via IP address(es)

  https://www.nuget.org/packages/Our.Shield.FrontendAccess

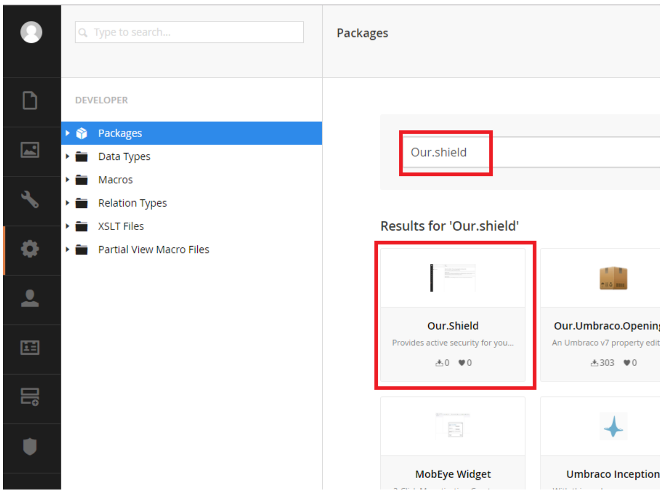
  ```
  PM> Install-Package Our.Shield.FrontendAccess -Version 1.0.3
  ```

Installing via Umbraco Package Manager

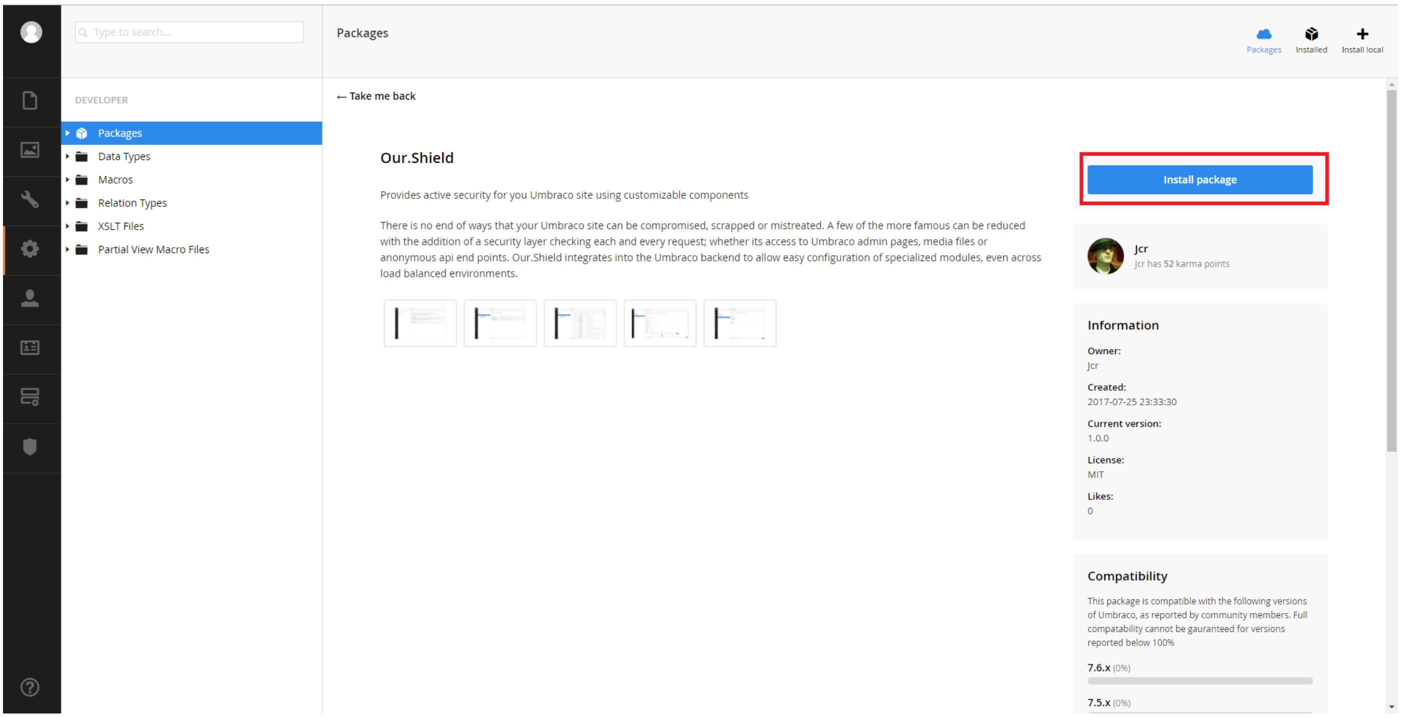
https://our.umbraco.org/projects/backoffice-extensions/ourshield/

This installation contains the Shield framework, and all available Shield apps

First, navigate to the developer section of Umbraco, click on the packages node and search for Shield



Next, Click on the package, and then the install button



Our.Shield

Provides active security for you Umbraco site using customizable components

There is no end of ways that your Umbraco site can be compromised, scrapped or mistreated. A few of the more famous can be reduced with the addition of a security layer checking each and every request; whether its access to Umbraco admin pages, media files or anonymous api end points. Our.Shield integrates into the Umbraco backend to allow easy configuration of specialized modules, even across load balanced environments.

Jcr
Jcr has 52 karma points

Information

Owner:
Jcr

Created:
2017-07-25 23:33:30

Current version:
1.0.0

License:
MIT

Likes:
0

Compatibility

This package is compatible with the following versions of Umbraco, as reported by community members. Full compatability cannot be gauranteed for versions reported below 100%

7.6.x (0%)

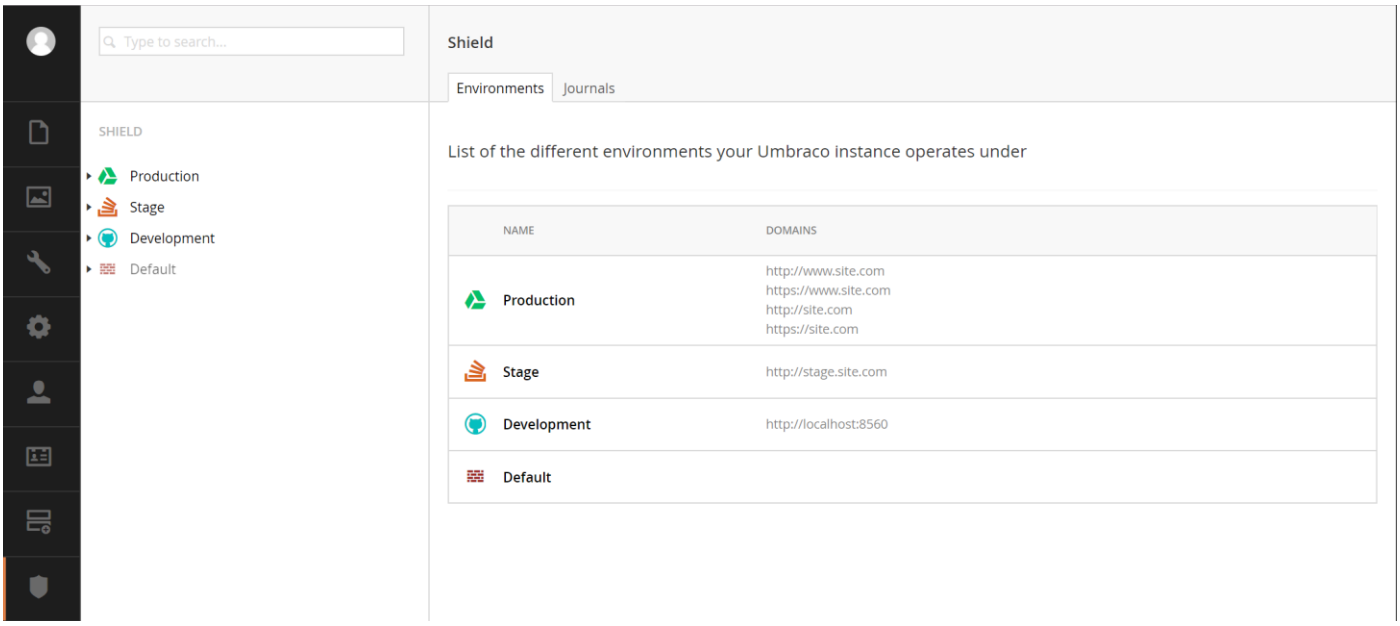
7.5.x (0%)

Afterwards, Shield should be installed

Shield

Shield is the framework for the apps that can be installed to provide the active security. It contains the custom section to be displayed in Umbraco and does the 'heavy' lifting for the installed app(s).

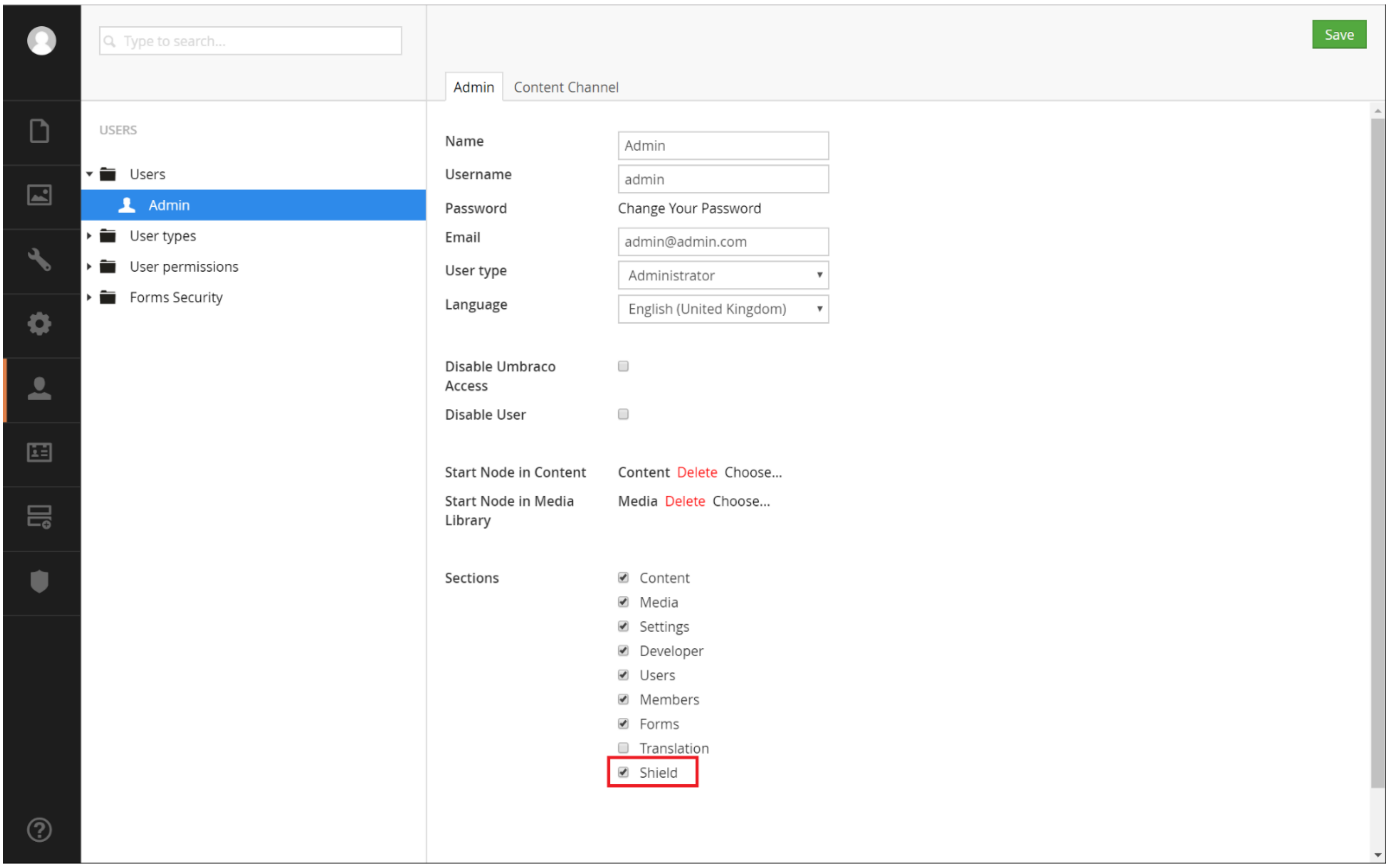
Once installed, you should see a new custom section within the backoffice of Umbraco.



List of the different environments your Umbraco instance operates under

NAME	DOMAINS
Production	http://www.site.com https://www.site.com http://site.com https://site.com
Stage	http://stage.site.com
Development	http://localhost:8560
Default	

If the new section doesn't display, you'll need to allow the currently logged in user to have access to the Shield section via the users' section:



Name	Admin
Username	admin
Password	Change Your Password
Email	admin@admin.com
User type	Administrator
Language	English (United Kingdom)
Disable Umbraco Access	□
Disable User	□
Start Node in Content	Content Delete Choose...
Start Node in Media Library	Media Delete Choose...

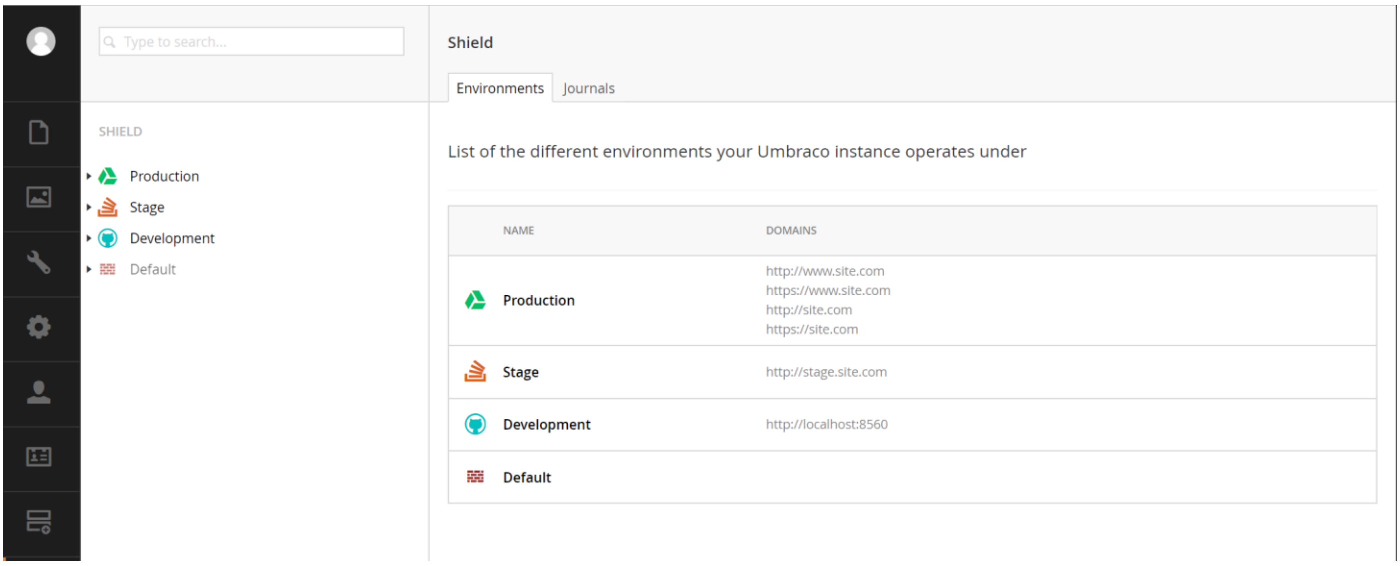
Sections

- ☑ Content
- ☑ Media
- ☑ Settings
- ☑ Developer
- ☑ Users
- ☑ Members
- ☑ Forms
- □ Translation
- ☑ Shield

Environments Dashboard

Initially, a ‘Default’ environment is created, which acts as a catch all environment and responds to all requests. Any app enabled and configured in the ‘Default’ environment will respond and process any request (this can be frontend webpages, backend, media or Web API requests) if none of the previous environments responded because they don’t match the request’s domain. As you create new environments with their own domains, any requests on those domains will be handled by the apps enabled and configured within that environment.

This allows different configuration of apps for your different environments; for example, Hot Linking protection only on your Production environment and Frontend Access restrictions on your Staging environment.



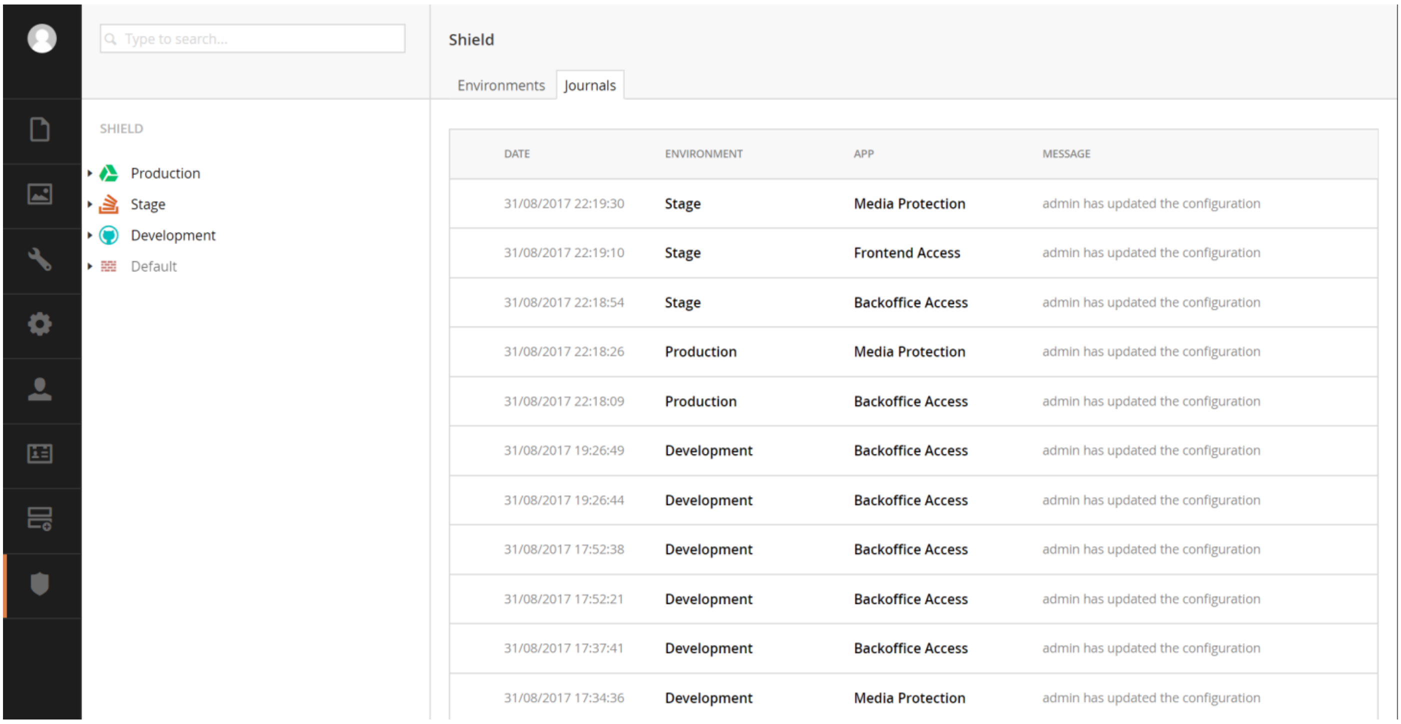
List of the different environments your Umbraco instance operates under

NAME	DOMAINS
Production	http://www.site.com https://www.site.com http://site.com https://site.com
Stage	http://stage.site.com
Development	http://localhost:8560
Default	

Journals Dashboard

The Journal tab will display all journal items (logs) that have been created by the different environment(s) and Shield app(s). A Journal is composed of the following:

- Date & time of when the Journal item was created
- The environment of the app that created this Journal item
- The app that created the Journal item
- A message of why the Journal item was created

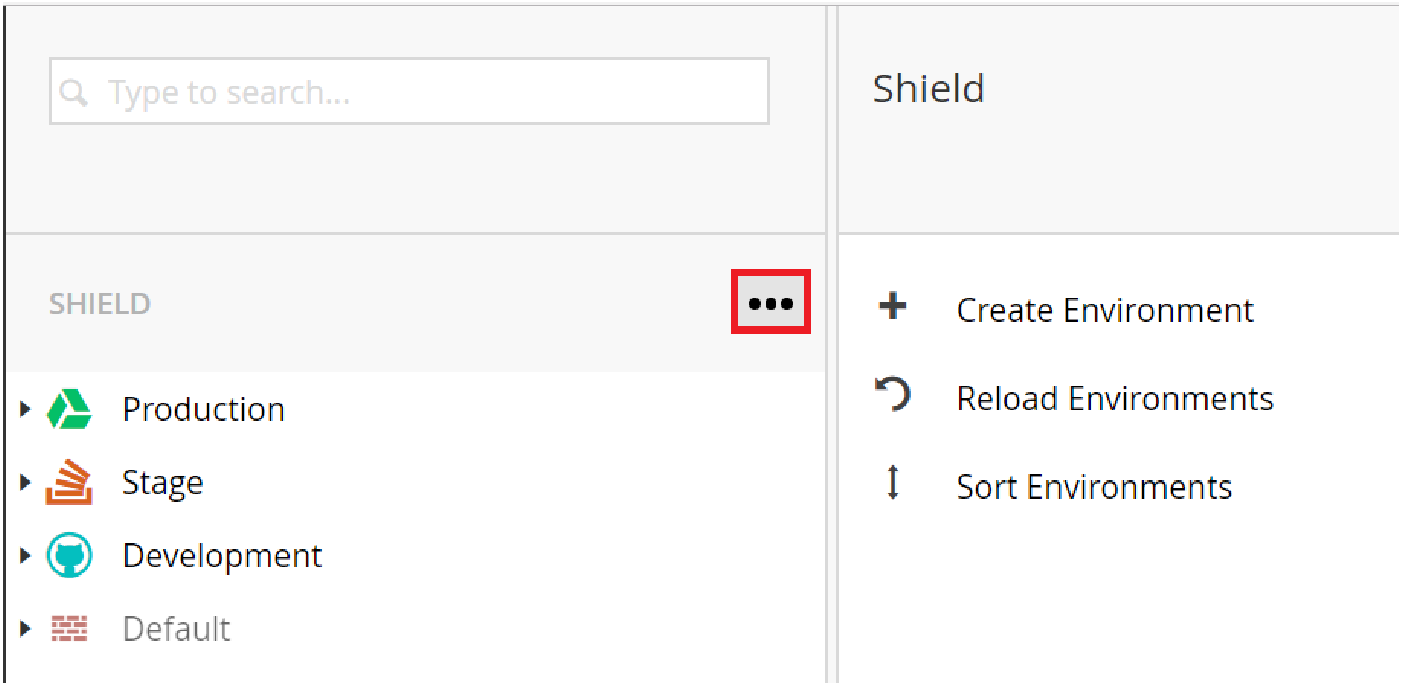


DATE	ENVIRONMENT	APP	MESSAGE
31/08/2017 22:19:30	Stage	Media Protection	admin has updated the configuration
31/08/2017 22:19:10	Stage	Frontend Access	admin has updated the configuration
31/08/2017 22:18:54	Stage	Backoffice Access	admin has updated the configuration
31/08/2017 22:18:26	Production	Media Protection	admin has updated the configuration
31/08/2017 22:18:09	Production	Backoffice Access	admin has updated the configuration
31/08/2017 19:26:49	Development	Backoffice Access	admin has updated the configuration
31/08/2017 19:26:44	Development	Backoffice Access	admin has updated the configuration
31/08/2017 17:52:38	Development	Backoffice Access	admin has updated the configuration
31/08/2017 17:52:21	Development	Backoffice Access	admin has updated the configuration
31/08/2017 17:37:41	Development	Backoffice Access	admin has updated the configuration
31/08/2017 17:34:36	Development	Media Protection	admin has updated the configuration

Tree

The tree will show a listing of the environments configured, with the desired Icon and name with a visual indication of whether or not the environment is active.

Clicking the three dots to Shield's root node:



Gives you the ability to create, reload, and sort the environments

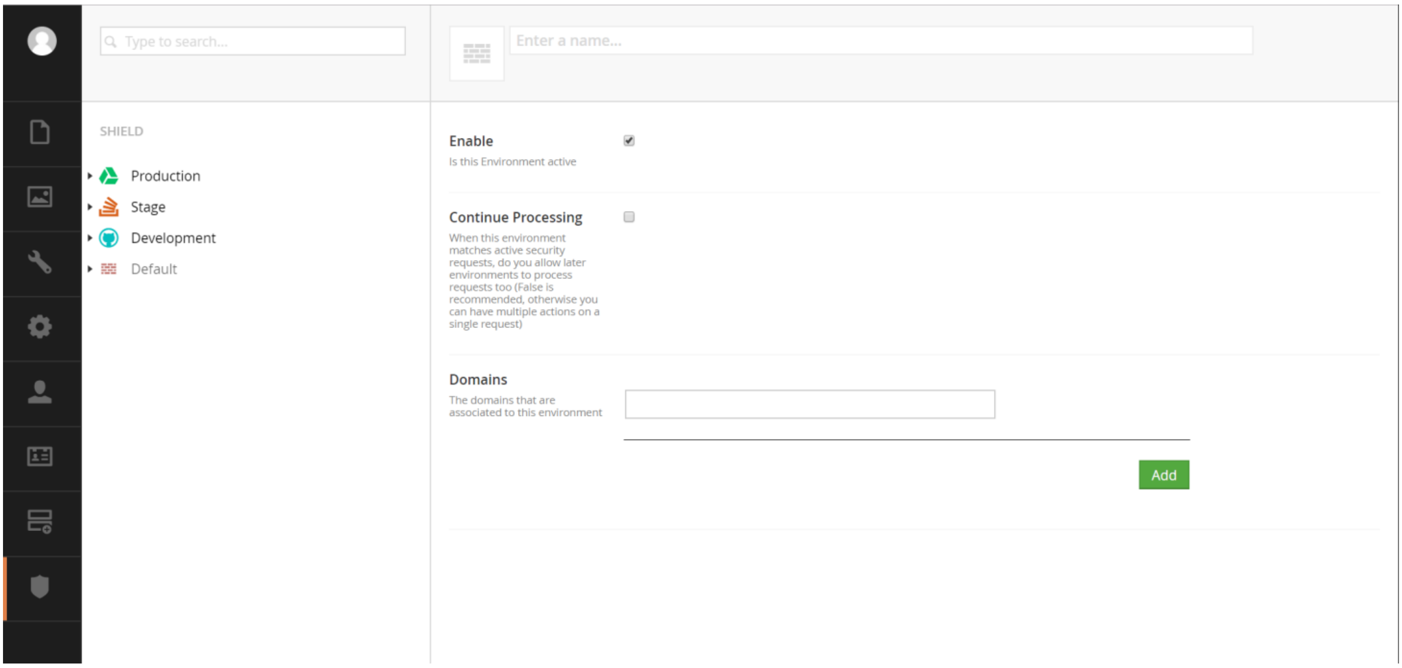
Create Environment

When creating an environment, you'll have the ability to

- Select an **Icon** that identifies visually this environment
- Set an unique **Name** for this environment
- **Enabled** this environment. When disabled all of the apps within the environment are disabled too. This is equivalent of the environment not existing. This can mean that requests that could have been handled by this environment will now be processed by the 'Default' Environment.
- **Continue Processing** allows future environments to process and handle a web request even if this environment has already processed it. This allows the chaining of app configurations across environments. We consider this as advanced behaviour and so suggest to keep this setting as false for simplicity.
- Add a list of **Domains** that define this environment. So for example if your Production environment used www.mydomain.com and www.myotherdomain.com, then for a production environment you would type http://www.mydomain.com and http://www.myotherdomain.com, then whenever Shield processes an active request it could identify those that belong to your Production environment because they match these two domains.

All web requests have a domain, the domain of the request is compared to the list of domains an environment has, if they match then each of the apps for that domain are processed.

If **Continue Processing** is true, then other environments are checked for matching domains also and if they match, then the apps associated with that domain are processed too. And then finally the 'Default' environment is processed.
When **Continue Processing** is false, no further environments are processed including 'Default'.



Type to search...

Enter a name...

SHIELD

- Production
- Stage
- Development
- Default

Enable ☑
Is this Environment active

Continue Processing ☐
When this environment matches active security requests, do you allow later environments to process requests too (False is recommended, otherwise you can have multiple actions on a single request)

Domains
The domains that are associated to this environment

Sort Environments

Once you have multiple environments, you should order them with the Production environment first, and the development environment last (before default). An example could be:

- Production
- Preview
- Staging
- QA
- Development

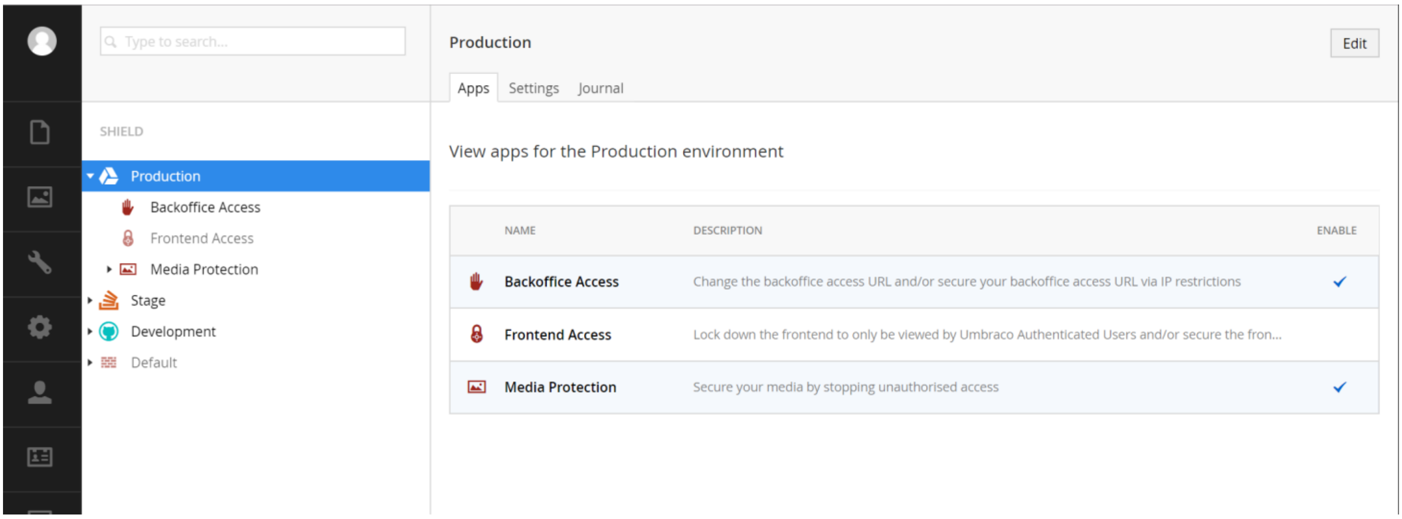
The reason for Production being first, is that this will, ever so slightly, be processed first and will speed up any Live requests – which normally is more important, but it is all very marginal.

Environment Node

An Environment will display all installed apps, the ability to edit an environment and view Journal entries. To edit an environment, there's an 'Edit' button to the right of the environment's name. The Edit view is the same as the create environment view, giving you the full ability to modify the environment as needed.

Apps

The Apps tab will display a listing of the Shield apps that are installed, showing the name, description and whether or not the app is enabled. Clicking on the app name will open up the corresponding app's configuration.



Type to search...

SHIELD

- Production
  - Backoffice Access
  - Frontend Access
  - Media Protection
- Stage
- Development
- Default

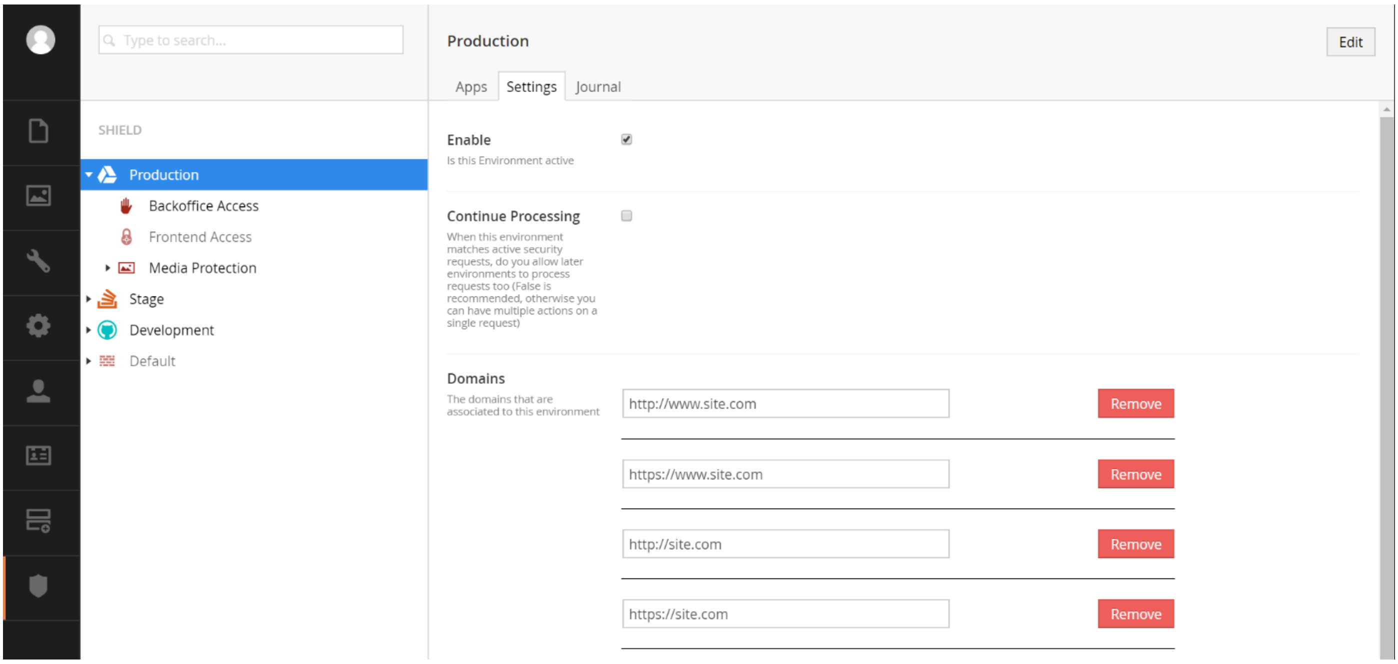
Production

View apps for the Production environment

NAME	DESCRIPTION	ENABLE
Backoffice Access	Change the backoffice access URL and/or secure your backoffice access URL via IP restrictions	✓
Frontend Access	Lock down the frontend to only be viewed by Umbraco Authenticated Users and/or secure the fron...	
Media Protection	Secure your media by stopping unauthorised access	✓

Settings

The Settings tab is very similar to creating/editing an environment, the difference being, on the settings tab, you don’t have the ability to change the icon or edit the name.



Enable ☑
Is this Environment active

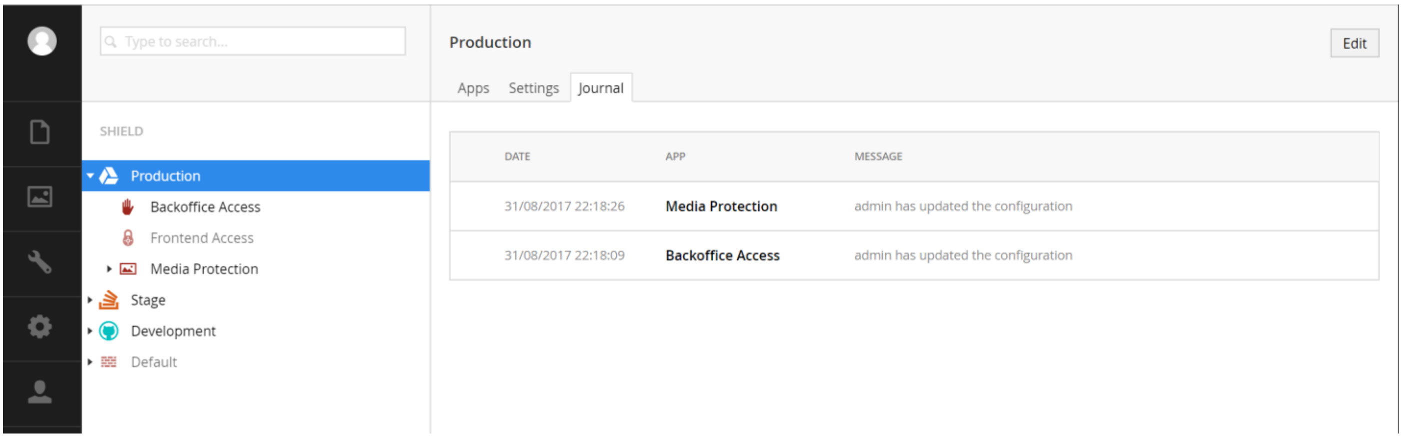
Continue Processing □
When this environment matches active security requests, do you allow later environments to process requests too (False is recommended, otherwise you can have multiple actions on a single request)

Domains
The domains that are associated to this environment

http://www.site.com
https://www.site.com
http://site.com
https://site.com

Journal

Similar to the Journal Dashboard, this will display the Journal items only for the selected environment. The difference being, the environment column is not included and only shows journal entries relevant to the selected environment.



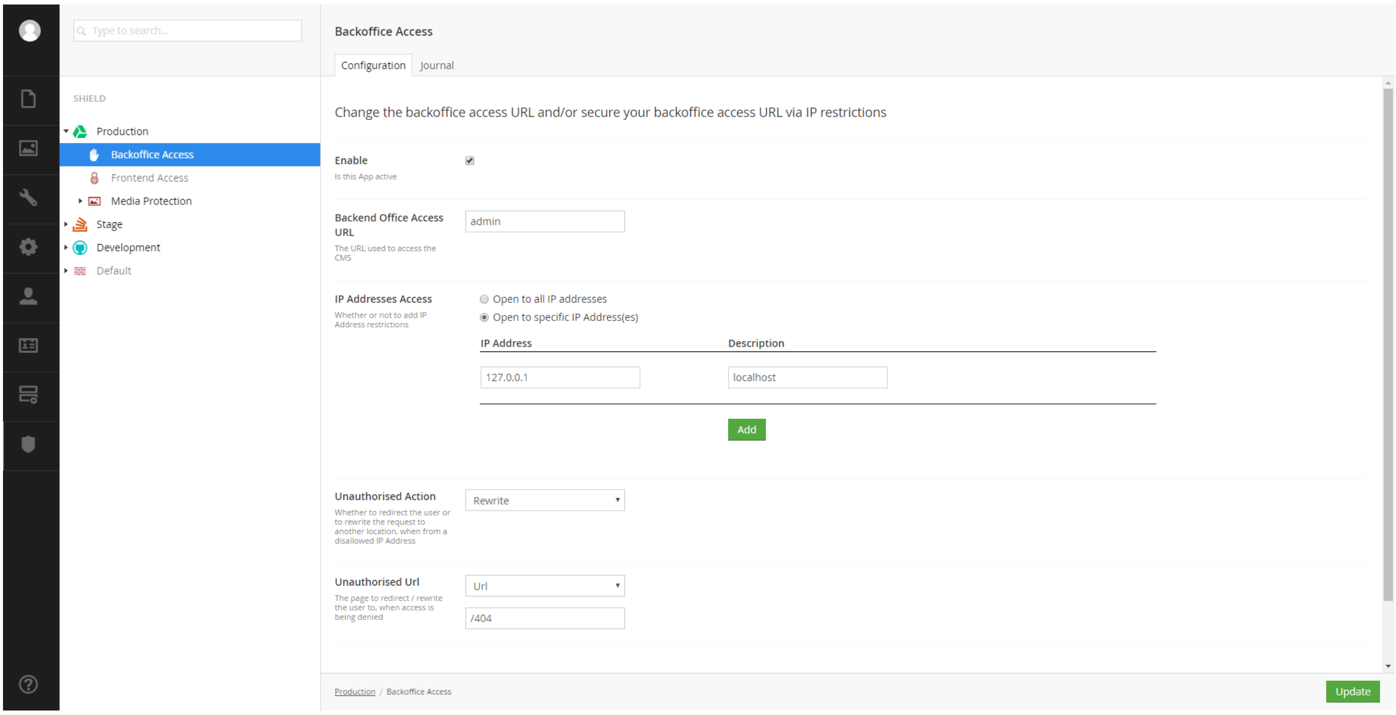
DATE	APP	MESSAGE
31/08/2017 22:18:26	Media Protection	admin has updated the configuration
31/08/2017 22:18:09	Backoffice Access	admin has updated the configuration

Backoffice Access

Backoffice Access grants you the ability to change the backoffice access URL to a URL you desire, with the ability to restrict who can access the URL by a white-list of IP Addresses if desired.

Configuration

- Enable or disable this app. When disabled the URL will return to the predefined default, which is "/umbraco".
- The backoffice access URL you wish to use, to access the admin area of Umbraco. This can be any valid combination of letters or numbers, non-case sensitive. You are not allowed white space, symbols or special characters.
- Whether or not the desired backoffice access URL is open to all IP Addresses, or specific IP Addresses. Selecting to only allow specific IP Addresses will display:
    - White-list table - allowing you to define either IPv4 and/or IPv6 Addresses of whom can access the desired backoffice access URL, with the ability to specify a short description for the IP Address, for future reference. if you required localhost, then you will need to add "127.0.0.1".
    - The unauthorised action - for users that are not within the white-list of valid IP addresses. You can select between redirecting or rewriting to another webpage. The difference is that for the end user, they will see their Url change within the address bar of their browser when redirecting, where rewriting the URL will not change.
    - What type of page unauthorised users are directed to
        - URL – Textbox to allow you to specify the page. i.e. /404
        - XPath – Textbox to allow you to specify the path to the page. i.e. //standardPage[@isDoc AND @nodeName='404 Not Found']
          OR
        - Content Picker – Provides you a content picker to select the desired page from the content section of Umbraco.



Backoffice Access

Change the backoffice access URL and/or secure your backoffice access URL via IP restrictions

Enable ☑
Is this App active

Backend Office Access URL: admin
The URL used to access the CMS

IP Addresses Access
Whether or not to add IP Address restrictions
○ Open to all IP addresses
● Open to specific IP Address(es)

IP Address	Description
127.0.0.1	localhost

Unauthorised Action: Rewrite
Whether to redirect the user or to rewrite the request to another location, when from a disallowed IP Address

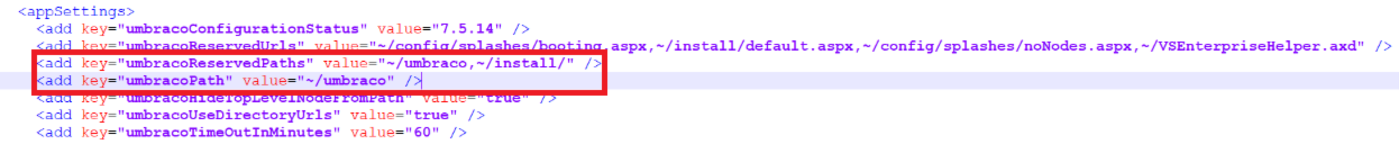
Unauthorised Url: Url
/404
The page to redirect / rewrite the user to, when access is being denied

Journal

Similar to the Journal Dashboard, this will only show the journal entries for the selected app & environment. The list will show warnings, messages and errors that have occurred within the app. This includes all unauthorised attempts to gain access to the backoffice access URL.

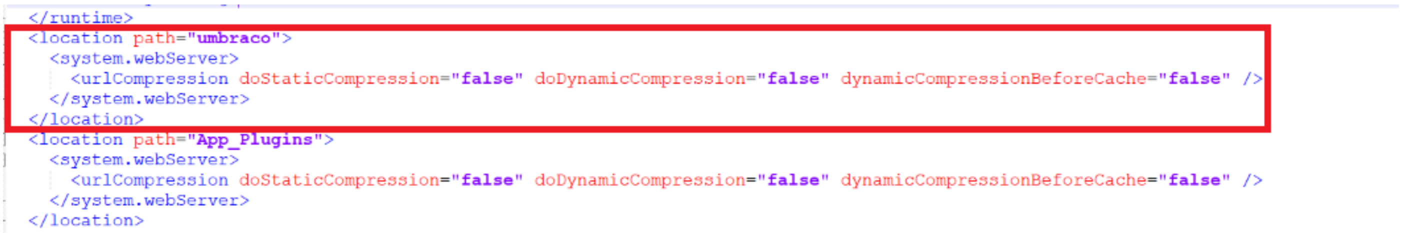
Upgrading Umbraco with Backoffice Access

To upgrade Umbraco while Backoffice Access is enabled or disabled, ensure within the website’s web.config file the ‘umbracoPath’ and ‘umbracoReservedPaths’ app settings are ‘umbraco’ respectively:



```
<appSettings>
  <add key="umbracoConfigurationStatus" value="7.5.14" />
  <add key="umbracoReservedUrls" value="~/config/splashes/booting.aspx,~/install/default.aspx,~/config/splashes/noNodes.aspx,~/VSEnterpriseHelper.axd" />
  <add key="umbracoReservedPaths" value="~/umbraco,~/install/" />
  <add key="umbracoPath" value="~/umbraco" />
  <add key="umbracoHideTopLevelNodeFromPath" value="true" />
  <add key="umbracoUseDirectoryUrls" value="true" />
  <add key="umbracoTimeOutInMinutes" value="60" />
```

As well as the location element (if set, and ensure there is only one!):



```
  </runtime>
  <location path="umbraco">
    <system.webServer>
      <urlCompression doStaticCompression="false" doDynamicCompression="false" dynamicCompressionBeforeCache="false" />
    </system.webServer>
  </location>
  <location path="App_Plugins">
    <system.webServer>
      <urlCompression doStaticCompression="false" doDynamicCompression="false" dynamicCompressionBeforeCache="false" />
    </system.webServer>
  </location>
```

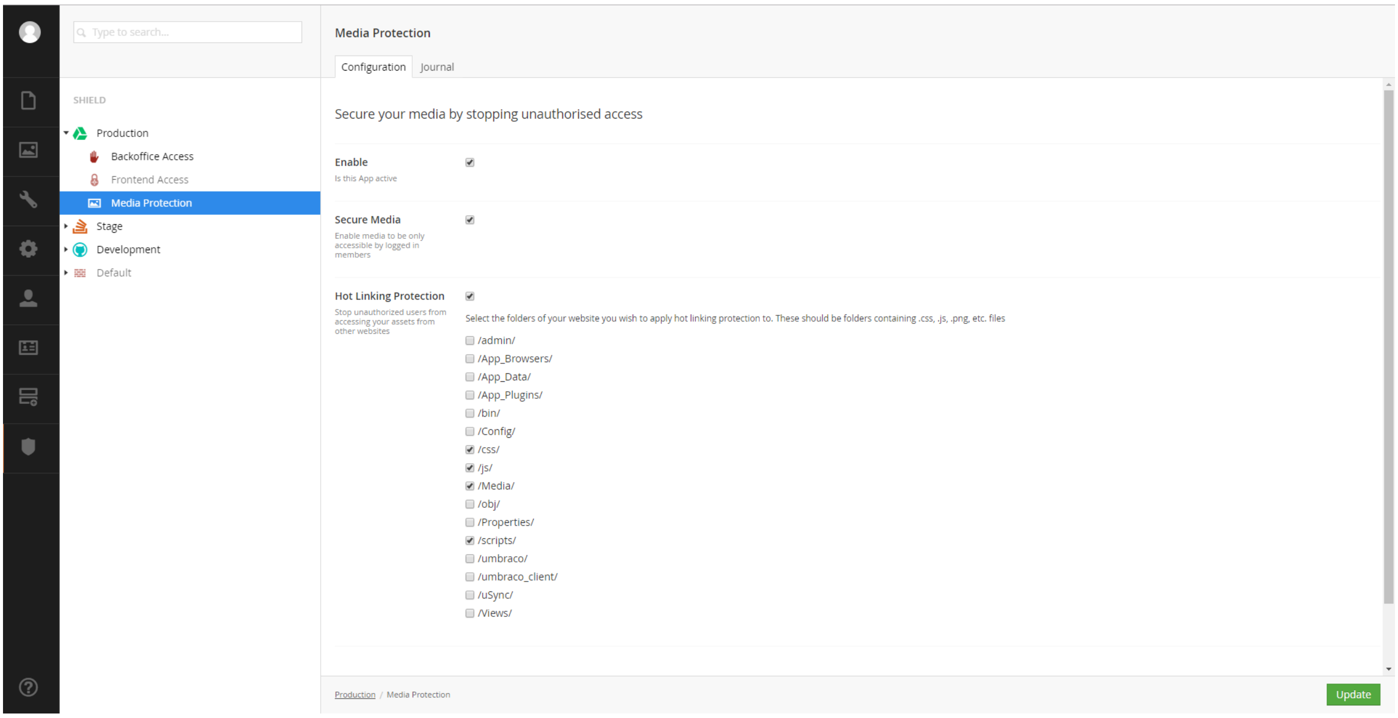
This is because, as part of Backoffice Access functionality, it will set these to whatever your preferred access URL is. You should then be able to upgrade Umbraco the normal way via the installer/upgrader, using the URL ‘/umbraco’. Once the upgrade process has been completed, you’ll find ‘/umbraco’ URL will no longer be accessible again, and will continue to work from your preferred access URL. You don’t need to reset these web.config values back to your preferred access URL, as Backoffice Access will handle this for you on the next app pool restart.

Media Protection

Media Protection gives you the ability to stop other websites from hot linking your media assets and allows you to assign media to only be viewed by authenticated members.

Configuration

- Enable or disable this app. When disabled there will be no active security on your website’s assets.
- When Secure Media is enabled, any Secure Folder, Image or File items that they, themselves have been specifically set to be Members only are restricted to your front-end users that have logged in.
- When Hot linking protection is enabled, it’ll show all the folders at the root of your website, you’ll need to select the folder(s) you desire to add hot linking protection to. Ideally, you should select the folder(s) that contain your website’s assets. i.e. the media folder, folder(s) that contain .css, .js, .png, etc. files



Media Protection

Secure your media by stopping unauthorised access

Enable ☑
Is this App active

Secure Media ☑
Enable media to be only accessible by logged in members

Hot Linking Protection ☑
Stop unauthorized users from accessing your assets from other websites

Select the folders of your website you wish to apply hot linking protection to. These should be folders containing .css, .js, .png, etc. files

☐ /admin/
☐ /App_Browsers/
☐ /App_Data/
☐ /App_Plugins/
☐ /bin/
☐ /Config/
☑ /css/
☑ /js/
☑ /Media/
☐ /obj/
☐ /Properties/
☑ /scripts/
☐ /umbraco/
☐ /umbraco_client/
☐ /uSync/
☐ /Views/

Production / Media Protection

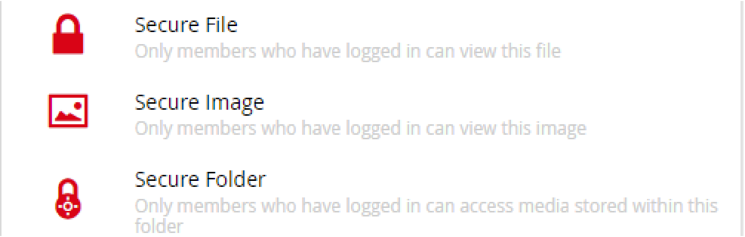
Journal

As per the other Journal listings, it displays journals that have been logged for this environment & app. The list will show warnings, messages and errors that have occurred within the app. This includes all unauthorised attempts to hot link your website’s assets.

Media Types

Once Media Protection has been installed, you should have 3 new media types to use:

1. Secure File
2. Secure Image
3. Secure Folder

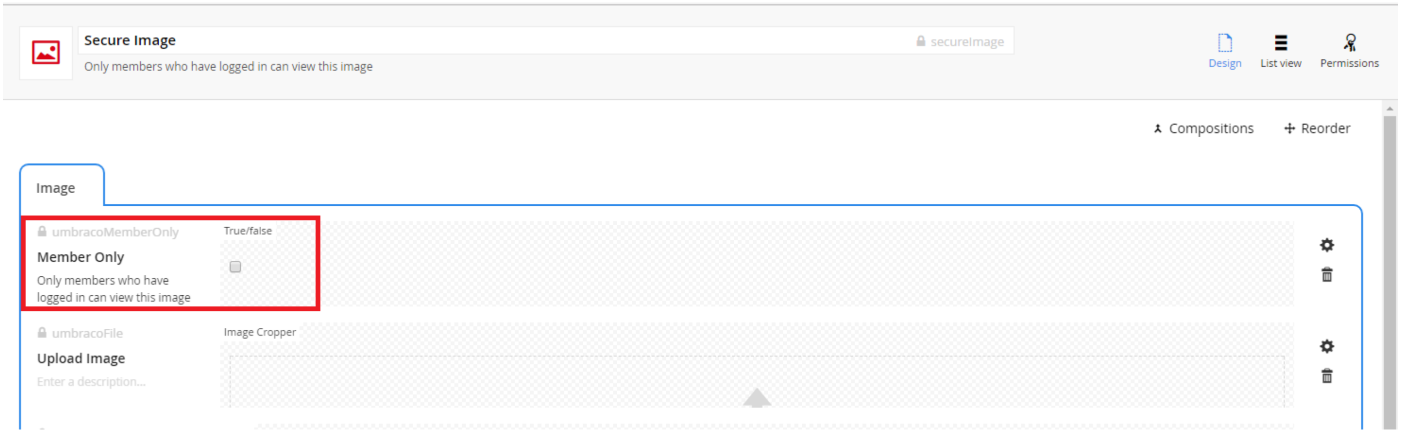


Secure File
Only members who have logged in can view this file

Secure Image
Only members who have logged in can view this image

Secure Folder
Only members who have logged in can access media stored within this folder

These 3 new media types are used in conjunction with the Configuration’s “Secure Media” option. You’re able to create more secure media types by creating a new media type and having a property with a special alias of “umbracoMemberOnly” as type “True/False”.



Secure Image

Only members who have logged in can view this image

secureImage

umbracoMemberOnly

True/false

Member Only

Only members who have logged in can view this image

□

umbracoFile

Image Cropper

Upload Image

Enter a description...

Secure Media

To enable the Secure Media to work as expected, you'll need to create some new media items using one of the above mentioned new media types (or your custom secure media type(s) if you created any). Once the media items have been created, and the "Member Only" tickbox is checked for said media items, as well as the configuration's Secure Media option is enable, only authenticated members can view the media items where the "Member Only" tickbox is checked.

Disabling Secure Media configuration option will allow access to the media items regardless of whether or not the "Member Only" tickbox on a media item is checked.

If you create a Secure Folder media item, and place all your media items in this secure folder, you'll only need to check the "Member Only" tickbox on the Secure Folder item. Media Protection will look at the media item's ancestors (parent nodes), and if an ancestor has the "Members Only" tickbox checked, then all its children are as well. For example, if you had the following media setup:

-Secure Folder

---Secure Image

---Image

---Image

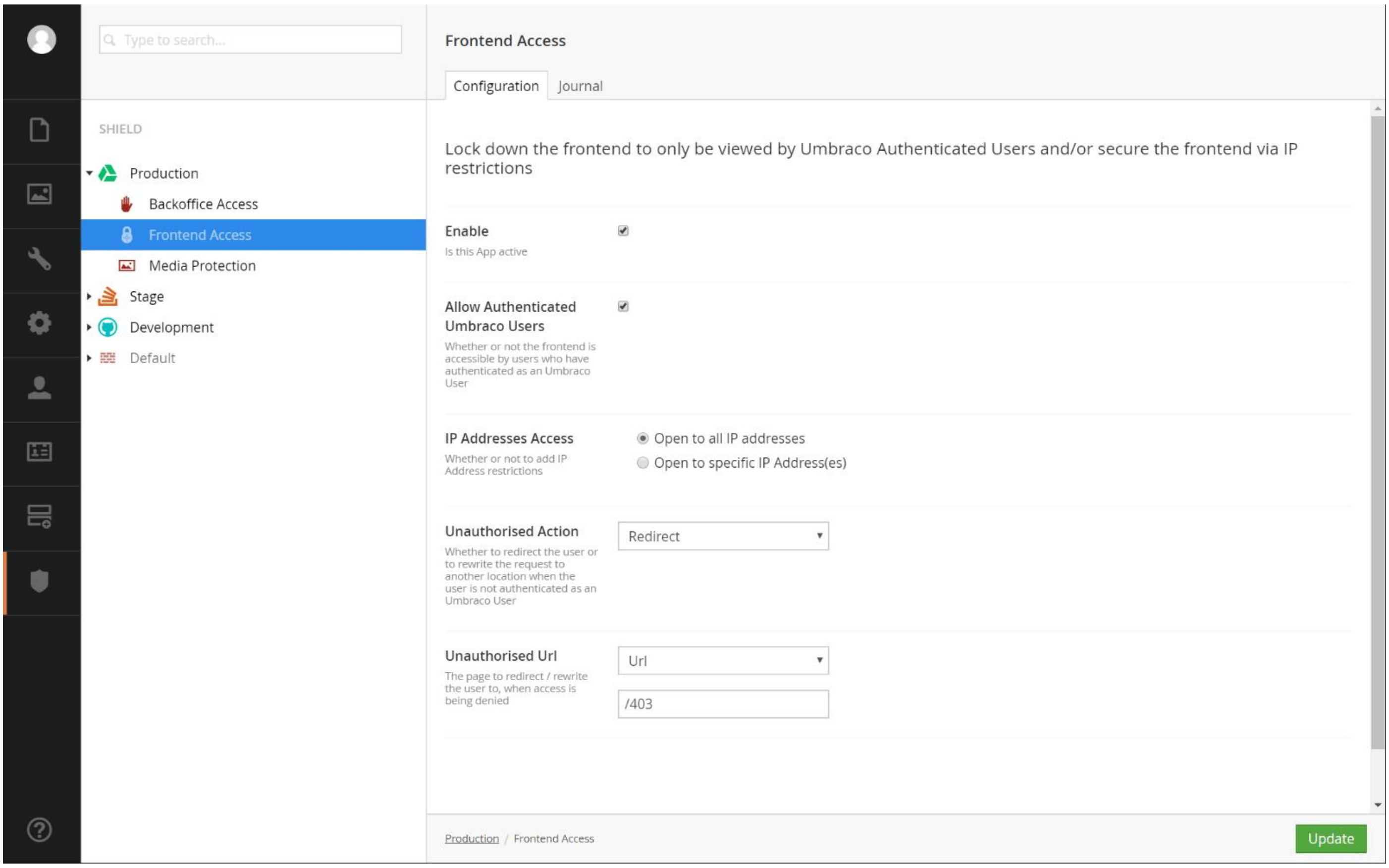
And on the Secure Folder, you have the "Member Only" tickbox checked, then all the children, (the x1 Secure Image & the x2 Image) will only be accessible by authenticated users. The x1 Secure Image item itself doesn't need the "Member Only" tickbox checked.

Frontend Access

Frontend Access gives you the ability to lock down the frontend of your website to either those authenticated Umbraco backend Users and/or IP Address restrictions. Ideally this app should be disabled on your production website (or the default environment), and enabled on your other environments – if you have multiple environments setup.

Configuration

- The ability to enable/disable this app. When disabled this app doesn't limit access to the frontend in any way.
- Whether the Frontend of your website is accessible by authenticated Umbraco Users
- Whether the frontend is accessible by all IP addresses or to specific IP Addresses
- Whether to redirect or rewrite the request to another location
- The URL to redirect or rewrite the request to.



Frontend Access

Lock down the frontend to only be viewed by Umbraco Authenticated Users and/or secure the frontend via IP restrictions

Enable ☑
Is this App active

Allow Authenticated Umbraco Users ☑
Whether or not the frontend is accessible by users who have authenticated as an Umbraco User

IP Addresses Access
Whether or not to add IP Address restrictions
● Open to all IP addresses
○ Open to specific IP Address(es)

Unauthorised Action: Redirect
Whether to redirect the user or to rewrite the request to another location when the user is not authenticated as an Umbraco User

Unauthorised Url: Url
The page to redirect / rewrite the user to, when access is being denied
/403

Production / Frontend Access

Journal

Lists all unauthorised attempts to access the frontend of your website, as well as saves and updates of the configuration.

Extending Shield with your own app

Within Visual Studio (or something similar) you'll need to do the following:

1. Create a new project

2. Install UmbracoCMS & Our.Shield.Core into your project via NuGet

3. Create a class which will be the configuration for your app; the configuration is where you allow the app to be enabled/disabled, allow the end user to edit the configuration, to work the way they desire within the parameters of which your app offers.

Create a class inheriting from **Our.Shield.Core.Models.Configuration**, you will need to decorate the class with Json attributes as this will be serialized/deserialized to and from your view and the database:

Using Media Protection as an example

```
[JsonObject(MemberSerialization.OptIn)]
public class MediaProtectionConfiguration: Configuration
{
    [JsonProperty("enableHotLinkingProtection")]
    public bool EnableHotLinkingProtection { get; set; }

    [JsonProperty("enableMemberOnlyMedia")]
    public bool EnableMembersOnlyMedia { get; set; }
}
```

4. Create a class that will be your app, which uses your configuration created in the previous step.

The new class will need to inherit from **Our.Shield.Core.Models.App** passing your configuration as a generic; using media protection as an example:

```
[AppEditor("/App_Plugins/Shield.MediaProtection/Views/MediaProtection.html")]
public class MediaProtectionApp : App<MediaProtectionConfiguration>
{
    public override string Id => "MediaProtection"

    public override string Name => "Media Protection";

    public override string Description => "Secure your media by stopping unauthorised access";

    public override string Icon => "icon-picture red";

    public override IConfiguration DefaultConfiguration
    {
        get
        {
            return new MediaProtectionConfiguration
            {
                EnableHotLinkingProtection = true,
                EnableMembersOnlyMedia = true
            }
        }
    }
    public override bool Execute(IJob job, IConfiguration c)
    {
        var config = c as MediaProtectionConfiguration;

        //We’ll come back to this in a later step

        return true;
    }
}
```

5. Create the html page, as defined by the relative url in your AppEditor attribute, for best practices, I’d advise you to follow the guide for creating a custom package for Umbraco. i.e. all files should be stored within the ~/App_Plugins/ directory with a package.manifest file

```
[AppEditor("/App_Plugins/Shield.MediaProtection/Views/MediaProtection.html")]
public class MediaProtectionApp : App<MediaProtectionConfiguration>
{
   ...
}
```

And populate with the html that defines your configuration panel

```
<div ng-controller="Shield.Editors.MediaProtection.Edit as vm">
    <div class="umb-el-wrap control-group umb-control-group">
        <label for="hotlinking" class="control-label">
            <span>Hot Linking Protection</span>
            <small>
                Stop unauthorized users from accessing your media from other websites
            </small>
        </label>
        <div class="controls">
            <input type="checkbox" id="hotlinking"
                   ng-model="vm.configuration.enableHotLinkingProtection" />
        </div>
    </div>

    <div class="umb-el-wrap control-group umb-control-group">
        <label for="securemedia" class="control-label">
            <span>Secure Media</span>
            <small>
                Enable media to be only accessible by logged in members
            </small>
        </label>
        <div class="controls">
            <input type="checkbox" id="securemedia"
                   ng-model=" vm.configuration.enableMemberOnlyMedia" />
        </div>
    </div>
</div>
```

6. Create a package.manifest file and populate with your js and css file(s) required by your html view from the previous step.

```
{
  "javascript": [
    "~/App_Plugins/Shield.MediaProtection/Scripts/MediaProtection.js",
  ],
  "css": [
    "~/App_Plugins/Shield.MediaProtection/Css/MediaProtection.css",
  ]
}
```

7. Create the angular module within the js file. This should also contain any validation your configuration requires as angular directives.

```
angular.module('umbraco').controller('Shield.Editors.MediaProtection.Edit',
    ['$scope', function ($scope) {
        var vm = this;
        angular.extend(vm, {
            configuration: $scope.$parent.configuration
        });
    }]
);
```

$scope.$parent.configuration will contain your serialized configuration created in step 3.

8. Now to return to the Execute method created in step 4.

```
public class MediaProtectionApp : App<MediaProtectionConfiguration>
{
    ...

    public override bool Execute(IJob job, IConfiguration c)
    {
        var config = c as MediaProtectionConfiguration;

        ...

        return true;
    }
}
```

First off, you'll want to return true when your app is disabled, this informs the Shield framework that your app has successfully 'set up' to provide security. If we were to return false instead, this will inform the framework to try again, we only want to return false in the worst-case scenarios:

```
public override bool Execute(IJob job, IConfiguration c)
{
    var config = c as MediaProtectionConfiguration;

    if (config.Enable == false)
    {
        return true;
    }

    //this is where you'd want to put your security logic

    ...

    return true;
}
```

9. Replace the comment "//this is where you'd want to put your security logic", with whatever your needs are. Bear in mind though, you cannot access *UmbracoContext.Current* and certain other *Umbraco* classes within this method. This is due to the execute method running in its own thread, and therefore, not all of Umbraco classes have been initialised. However, for example, you could add a watcher on Shield's HttpModule, which provides you a HttpApplication as one of the arguments to the anonymous function of which Job.WatchWebRequest requires.

```
public override bool Execute(IJob job, IConfiguration c)
{
    var config = c as MediaProtectionConfiguration;

    //Unsubscribe any watches on the HttpModule, in-case the app is being disabled
    job.UnwatchWebRequests();

    if (config.Enable == false)
    {
        return true;
    }

    //Check if hotlinking protection is enabled
    //and if so, subscribe a watch to the HttpModule
    if (config.EnableHotLinkingProtection)
    {
        var mediaFolder = VirtualPathUtility.ToAbsolute(new
             Uri(Umbraco.Core.IO.SystemDirectories.Media, UriKind.Relative).ToString()) + "/";

        job.WatchWebRequests(new Regex(mediaFolder, RegexOptions.IgnoreCase), 50, (count, httpApp) =>
        {
            var referrer = httpApp.Request.UrlReferrer;
            if (referrer == null
                || String.IsNullOrWhiteSpace(referrer.Host)
                || referrer.Host.Equals(httpApp.Request.Url.Host,
                        StringComparison.InvariantCultureIgnoreCase))
            {
                //This media is being accessed directly,
                //or from a browser that doesn't pass referrer info,
                //or from our own domain
                //so, allow access
                return WatchCycle.Continue;
            }

            //Someone is trying to hotlink our media
            job.WriteJournal(new JournalMessage($"Access was denied, {referrer.Host} is trying to hotlink
                your media assets"));
            httpApp.Response.StatusCode = (int) HttpStatusCode.Forbidden;
            httpApp.Response.End();
            return WatchCycle.Stop;
        });
    }

    return true;
}
```

Another example of what you could do, is attach your app to the content and/or media services for example.

```
public override bool Execute(IJob job, IConfiguration c)
{
    Umbraco.Core.Services.MediaService.Saved += MediaService_Saved;
    Umbraco.Core.Services.ContentService.Publishing += ContentService_Publishing;
    return true;
}

private void MediaService_Saved(IMediaService sender, Umbraco.Core.Events.SaveEventArgs<IMedia> e)
{
    ...
}

private void ContentService_Publishing(Umbraco.Core.Publishing.IPublishingStrategy sender,
    Umbraco.Core.Events.PublishEventArgs<IContent> e)
{
    ...
}
```