# upload

```
configuration
     images
         upload
```

Configures the uploading of local images to your application using Textbox.io's built in upload mechanism.

This property can be configured to either:

- Upload Images using a form POST
- Upload Images via a Custom JavaScript Upload Handler Function

## Properties

To use Textbox.io's built-in upload mechanism, use `upload.url`, `upload.basePath`, and `upload.credentials`.

| Property | Type | Default | Properties |
|---|---|---|---|
| `url` | String | | Defines the location of your POST acceptor script. This is the URL to which images will be uploaded. |
| `basePath` | String | | [Optional] Defines the base path of images uploaded in your application. This path is combined with the path returned from your POST acceptor script. |
| `credentials` | Boolean | false | [Optional] `true`: sends cookies with the upload POST (sets the XHR credentials flag). `false`: does not send cookies. |

To use a custom JavaScript handler function, use `upload.handler`. Note that defining `upload.handler` disables Textbox.io's built in upload mechanism.

| Property | Type | Default | Properties |
|---|---|---|---|
| `handler` | Function | | Configures the uploading of images to your application using a custom Javascript upload handler. |

## Examples

### Using Textbox.io's Built-in Upload Mechanism

The simplest way to handle images is use automatic background uploads. A simple `configuration` object is used to define the upload location for images. `basePath` is used in conjunction with the value returned by the POST handler to link to the uploaded image within the editor's content. More information is available in the Handling Local Images guide.

```
var config = {
        images : {
                upload : {
                        url : 'http://example.com/path/to/POSTAcceptor.php',
            basePath : 'http://example.com/path/to/images'
                }
        }
};

var editor = textboxio.replace('#targetId', config);
```

### Using a Custom JavaScript Upload Handling Function

If images must be uploaded using more than a simple HTTP POST, the entire upload process can be replaced. For example, you may wish to perform additional client side validation or manipulation on the images prior to their upload.  The handler function is passed three arguments; Data (object), Success (function) and Failure (function).

#### Data Object Properties

The first argument is an object providing information about the image that has been inserted. All properties are functions.

| Property Function | Type | Description |
|---|---|---|
| `data.blob()` | Blob | JavaScript Blob instance representing the image binary data |
| `data.base64()` | String | A copy of the image binary data pre-converted to base64 for your convenience |
| `data.filename()` | String | The actual filename, where possible, otherwise a generated filename based on the MIME type |
| `data.id()` | String | A unique identifier for this image |

### Success Function Parameters

The second argument is a function that indicates the upload has successfully completed

| | | |
|---|---|---|
| `url` | String | The URL to use as the image src. No post-processing is performed on this value (the `basePath` configuration property is ignored). |

### Failure Function Parameters

The third argument is a function that indicates the upload failed.

| | | |
|---|---|---|
| `message` | String (optional) | A message to display to the user in an error banner.<br><br>This value is optional - if you do not wish to translate your error a generic message is available which has been translated into all supported languages. |

```
var config = {
        images : {
                upload : {
                        handler : function (data, success, failure) {
                                // For example, if myuploader.upload() returns a promise, e.g. jQuery ajax
                                myuploader.upload(data.blob(), data.filename()).then(function (<upload
response>) {
                                        success(<response image url>);
                                }, function () {
                                        failure("my failure message");
                                });
                        }
                }
        }
};

var editor = textboxio.replace('#targetId', config);
```