

REST Content API Forms

Dean Gonzalez & Freddy Montes





Agenda

- Basic components of a content API submit
- How form data can be prepared
- Saving binary files
- Relating content
- Response verification of successful content submission
- Form demonstration that creates/updates content

Resources: REST Content API Documentation

All the methodologies covered in this webinar are mentioned in the “Save Content using Rest API” documentation.

Documentation:

<http://dotcms.com/docs/latest/save-content-using-rest-api>

Configuration: Allowing Front End Content

Front end content contribution is NOT allowed in dotCMS, by default. However, changing the following property to “true” in the dotmarketing-config.properties file will enable it:

[REST_API_CONTENT_ALLOW_FRONT_END_SAVING=true](#)

Configuration: Allowing Front End Content

Two permission settings are required for front end content contribution. The “CMS Anonymous” role must have the following:

“Add to” permissions on the Site, and “Edit” on child Content/Files on the Site

“Edit” permissions on the Content Type

Configuration: Permissions

Home Tasks & Dashboard | **Site Browser** Pages, Folders & Files | Content Manage your Site | Calendar Events Management | Forms & Polls Forms, Polls & Reports | Content Types Workflows & Categories | System Settings & Config

Global Page demo.dotcms.com Site Browser Add/Edit Host

Content Permissions History

Site: demo.dotcms.com

+ Add Role + Add User [Reset Permissions](#)

	View	Add to	Edit	Publish	Edit Permissions	Vanity URL
CMS Anonymous	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>
Select what permissions "CMS Anonymous" will have on the following children:						
Folders	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NA	<input type="checkbox"/>	NA
Containers	<input type="checkbox"/>	NA	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NA
Templates	<input type="checkbox"/>	NA	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NA
Templates-Layouts	<input type="checkbox"/>	NA	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NA
Pages	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NA
Legacy Files	<input checked="" type="checkbox"/>	NA	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NA
Links	<input checked="" type="checkbox"/>	NA	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NA
Content Type	<input checked="" type="checkbox"/>	NA	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NA
Content/Files	<input checked="" type="checkbox"/>	NA	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NA

[Apply Changes](#) [Reset all children permissions to match the above?](#)

Structure: REST API Form Basics

Things to know:

1. Use any JavaScript framework (jQuery, NodeJS, Angular...)
2. Json, XML, & Form UrlEncoded content types
3. Multipart Form Data is supported
4. Regardless of methodology, the form sends a data map of labels and values that match the format of the Content Type fields

Adding Files: Saving Binary Files

How Binary files are submitted:

dotCMS REST API supports PUT/POST requests using HTML5 [FormData\(\)](#)

Files must be appended in the Content Type binary field order

Ex: var dataform=new [FormData\(\)](#);

```
dataform.append('file', document.getElementById('photo1').files[0]);
```

```
dataform.append('file', document.getElementById('photo2').files[0]);
```


XMLHttpRequest advanced features - LS

Global

66.78% + 23.01% = 89.79%

Adds more functionality to XHR (aka AJAX) requests like file uploads, transfer progress information and the ability to send form data. Previously known as **XMLHttpRequest Level 2**, these features now appear simply in the XMLHttpRequest spec.

Current aligned

Usage relative

Show all

IE / Edge	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
8							4.1	
9		31					4.3	
¹ 10		42					4.4	
¹ 11	38	43	7.1		¹ 7.1		4.4.4	
Edge	39	44	8	30	8.4	8	40	42
	40	45	9	31	9			
	41	46		32				
	42	47						

Notes

Known issues (3)

Resources (4)

Feedback

¹ Partial support refers to not supporting `json` as `responseType`

² Partial support refers to not supporting `.timeout` and `.ontimeout`

³ Partial support refers to not supporting `blob` as `responseType`

Adding Files: Saving Binary Files

```
$.ajax({  
  url: '/api/content/save/1', (SAVE OR PUBLISH)  
  type: 'PUT', (PUT OR POST)  
  cache:false,  
  data: dataform,  
  //To avoid non-strings being sent thru jQuery.param()  
  processData: false,  
  contentType: false,  
  beforeSend: function (request){  
  },  
});
```

*Read More: <http://www.mattlunn.me.uk/blog/2012/05/sending-formdata-with-jquery-ajax>

Validation: Response Header Values

DotCMS REST API forms support submitting multiple contentlets using the same form, in addition to creating relationships between content. Additional content submissions can be submitted after the successful response from the first content submission.

How to validate the successful submission of the first contentlet:

```
success: function(data,status,xhr) {  
  
var a = xhr.getAllResponseHeaders\(\);
```

Validation: Response Header Values

On successful content creation, 3 header values are returned:

Location: Location is a pointer to where the content can be obtained using the content REST API

Identifier: The identifier of the submitted content object

Inode: The inode of the submitted content version

Data Aggregation: Relating Content

Use the following format to set related form data:

```
var departmentData = {  
    'ipAddress': '$!{request.getRemoteAddr()}',    (ip info is often useful)  
    'stName': 'Department',  
    'languageld': 1,  
    'Department-Employee': '+structureName:Department +identifier:' + id  
};
```

Updating existing Content

Use Velocity to test for the presence of a logged in user or a content identifier being passed to the page

Match the `userId` to a matching `contentlet` field or use the `$dotcontent.find($id)` tool to match existing content and load it into the form values

Submit the form, adding the content identifier to the form data to find and update the matching existing content

Resources: Where to access this information

This presentation is posted on the dotCMS videos page:

<http://dotcms.com/videos/index>

Our YouTube Channel: <https://www.youtube.com/user/dotCMS>

Documentation: <http://dotcms.com/docs/latest/save-content-using-rest-api>

REST API FORM DEMONSTRATION