## CHARLOTTE CRAFT CMS MEETUP GROUP

# CRAFT 2 TO 3 MIGRATIONS

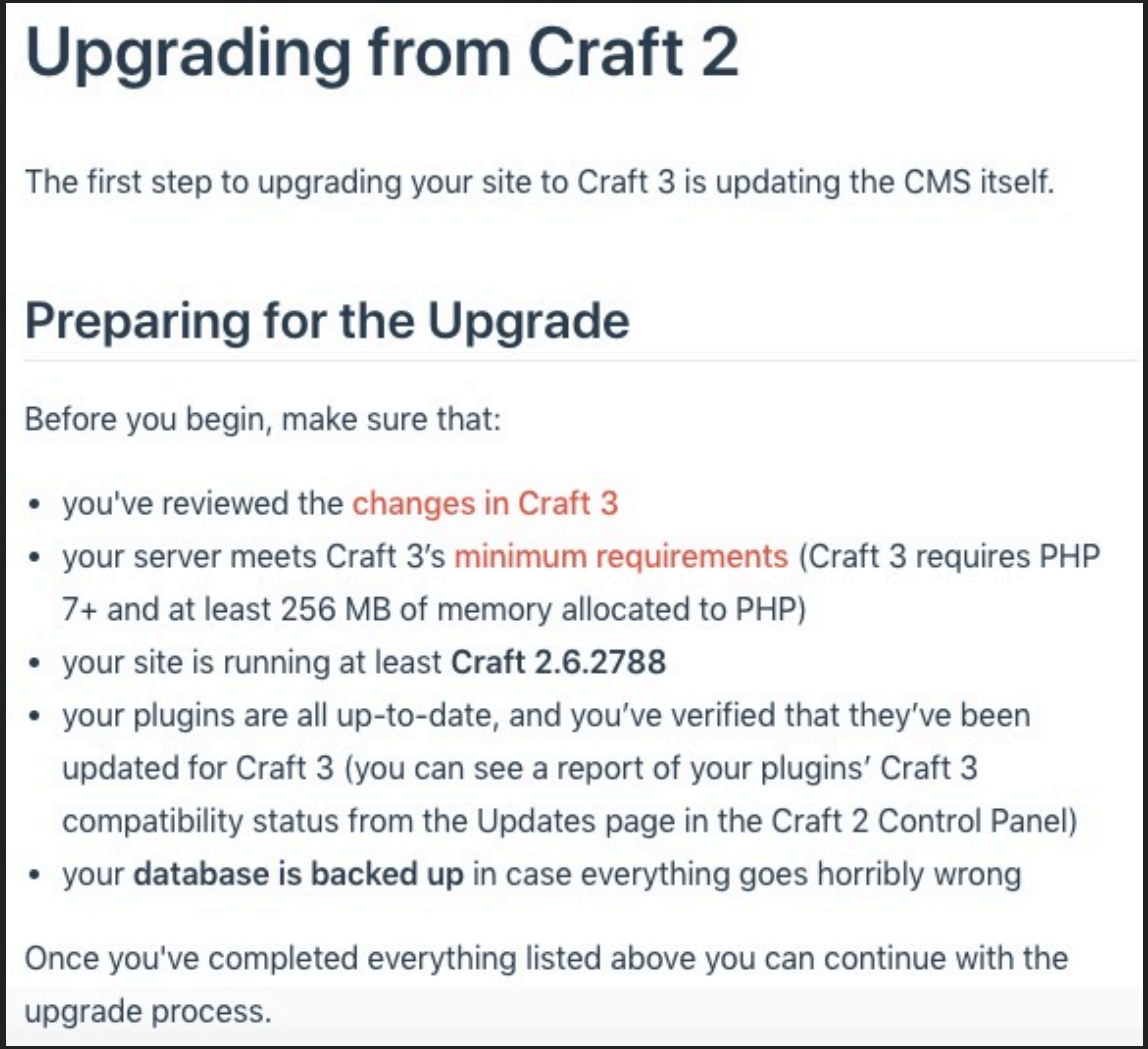
April 2020 - By Justin Holt
@justinholtweb

# AGENDA

- Approaches
- Migrating fields and structures
- Migrating content
- Multi-language/site solutions (locales)
- Data verification
- Template updates
- Routes in Yii 2
- Custom plugins
- Convince your client

# APPROACHES

- Use Craft provided migration tool.
  - Create base Craft 3 install
  - Backup your source
  - Match plugins to Craft 3 versions (Disable any that don’t match)
  - Edit required Craft files (based on guidelines)
  - Cross fingers and toes, perform migration.



# Upgrading from Craft 2

The first step to upgrading your site to Craft 3 is updating the CMS itself.

## Preparing for the Upgrade

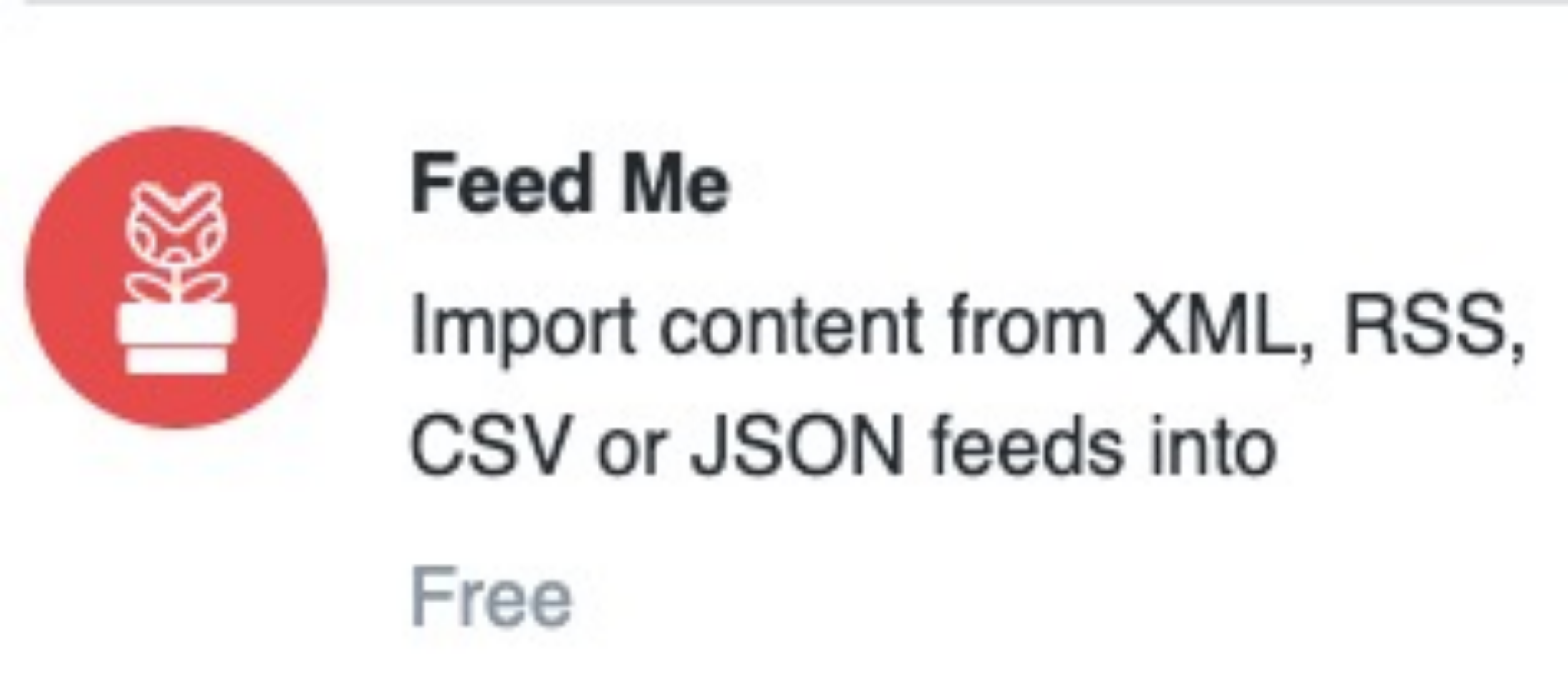
Before you begin, make sure that:

- you've reviewed the changes in Craft 3
- your server meets Craft 3’s minimum requirements (Craft 3 requires PHP 7+ and at least 256 MB of memory allocated to PHP)
- your site is running at least **Craft 2.6.2788**
- your plugins are all up-to-date, and you’ve verified that they’ve been updated for Craft 3 (you can see a report of your plugins’ Craft 3 compatibility status from the Updates page in the Craft 2 Control Panel)
- your **database is backed up** in case everything goes horribly wrong

Once you've completed everything listed above you can continue with the upgrade process.

# APPROACHES

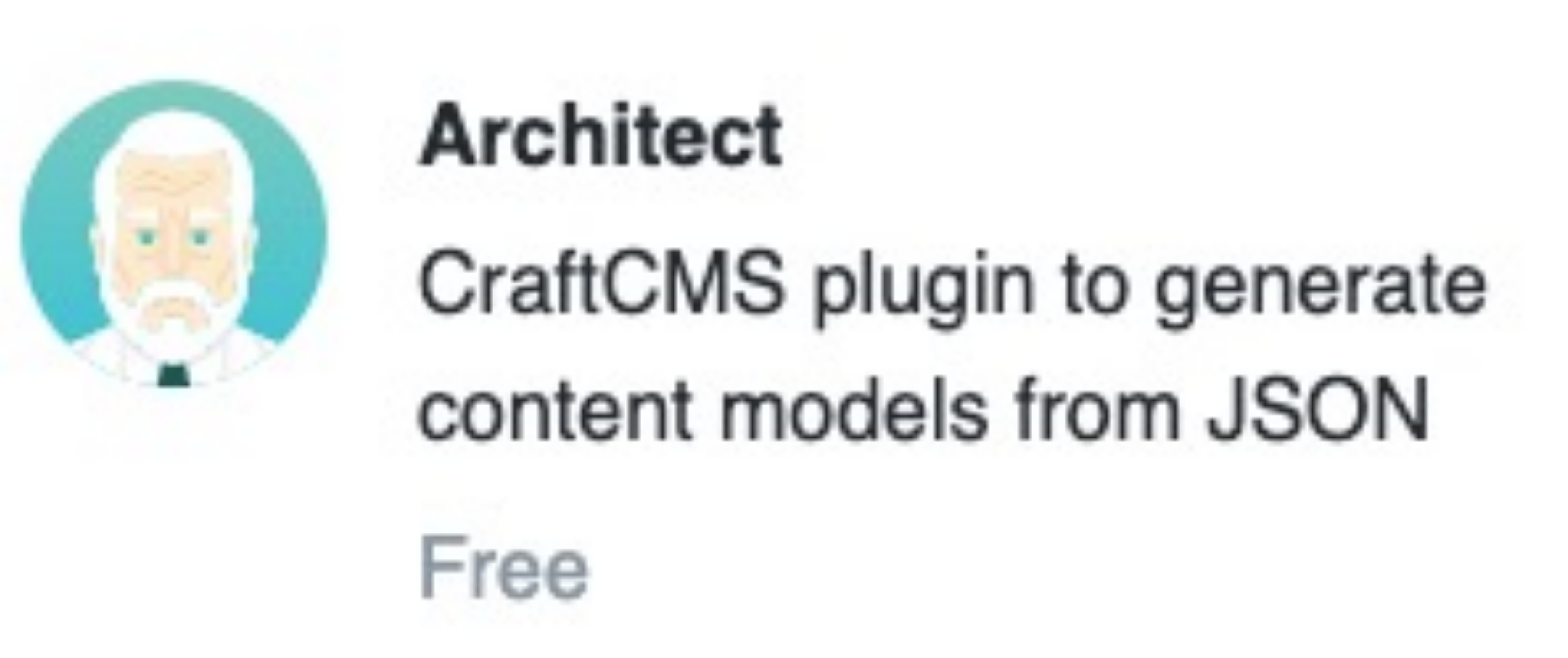
- Start with a new Craft 3 site, manually migrate
  - Best approach for complex sites
  - Best approach when compatible plugins may not be available in Craft 3
  - Best approach for locales to sites
  - Best approach for migrating sites that will continue to receive content updates on source (rolling migrations)



**Feed Me**

Import content from XML, RSS, CSV or JSON feeds into

Free



**Architect**

CraftCMS plugin to generate content models from JSON

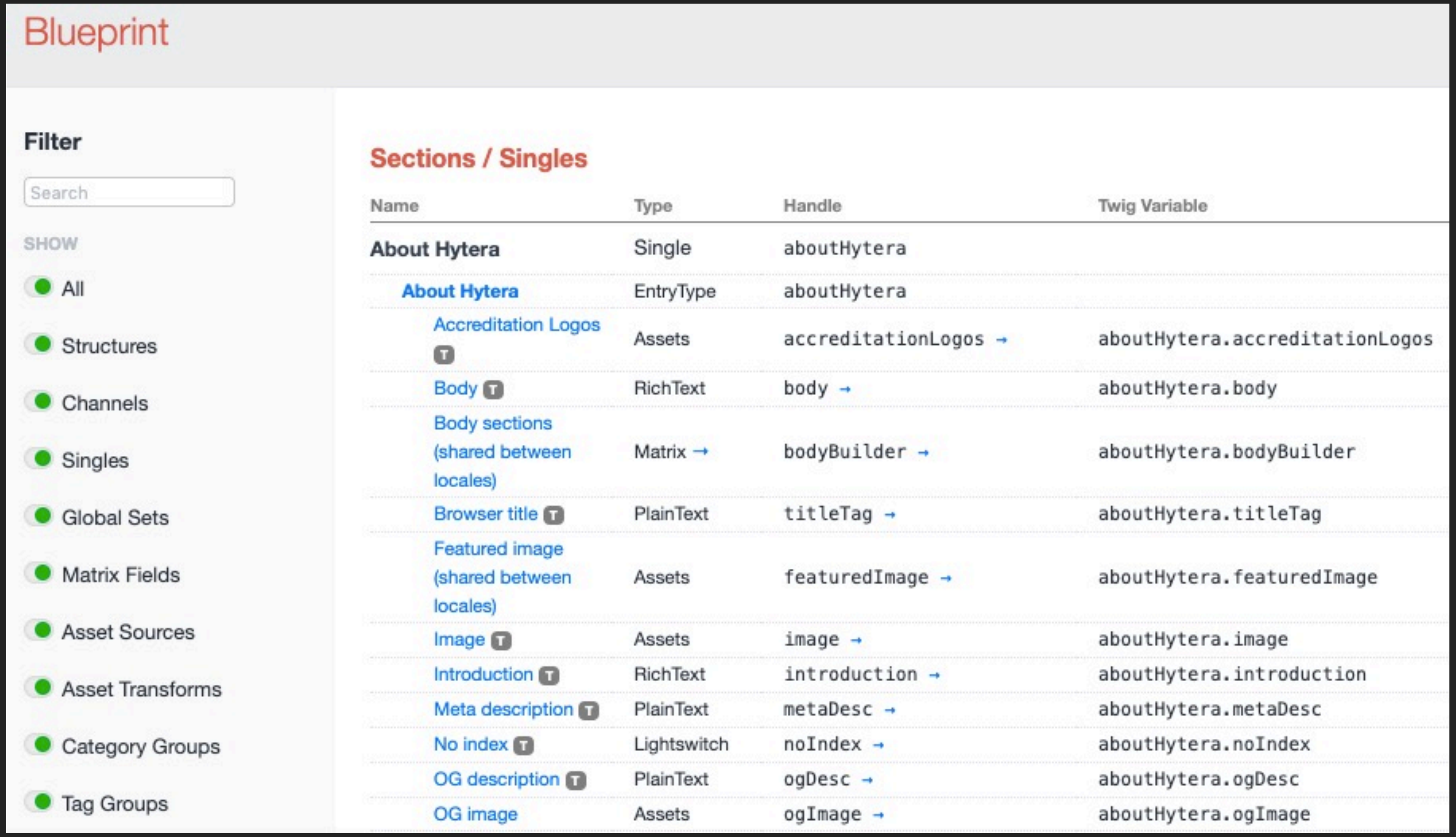
Free

# MIGRATING FIELDS AND STRUCTURE

- First always back up your stable local instance
- Create additional new backups as you perform these suggested processes
    - *Rollbacks of partial imports are never fun to resolve.*
- Install the Redactor plugin so any Rich Text fields can be properly migrated to Redactor fields
- Review all Craft 2 fields and structure, familiarize yourself

# MIGRATING FIELDS AND STRUCTURE

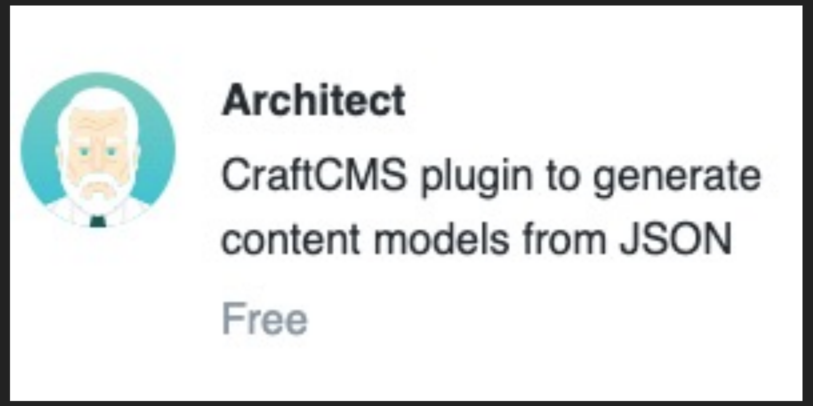
- Craft 2 Blueprint plugin
  - Helps you quickly visualize field structures of:
    - Sections
    - Singles
    - Channels
    - Globals
    - Asset Sources
    - Category Groups
    - Asset sources and transforms



Sections / Singles

| Name | Type | Handle | Twig Variable |
|---|---|---|---|
| **About Hytera** | Single | `aboutHytera` | |
| **About Hytera** | EntryType | `aboutHytera` | |
| Accreditation Logos T | Assets | `accreditationLogos` → | `aboutHytera.accreditationLogos` |
| Body T | RichText | `body` → | `aboutHytera.body` |
| Body sections (shared between locales) | Matrix → | `bodyBuilder` → | `aboutHytera.bodyBuilder` |
| Browser title T | PlainText | `titleTag` → | `aboutHytera.titleTag` |
| Featured image (shared between locales) | Assets | `featuredImage` → | `aboutHytera.featuredImage` |
| Image T | Assets | `image` → | `aboutHytera.image` |
| Introduction T | RichText | `introduction` → | `aboutHytera.introduction` |
| Meta description T | PlainText | `metaDesc` → | `aboutHytera.metaDesc` |
| No index T | Lightswitch | `noIndex` → | `aboutHytera.noIndex` |
| OG description T | PlainText | `ogDesc` → | `aboutHytera.ogDesc` |
| OG image | Assets | `ogImage` → | `aboutHytera.ogImage` |

# MIGRATING FIELDS AND STRUCTURE

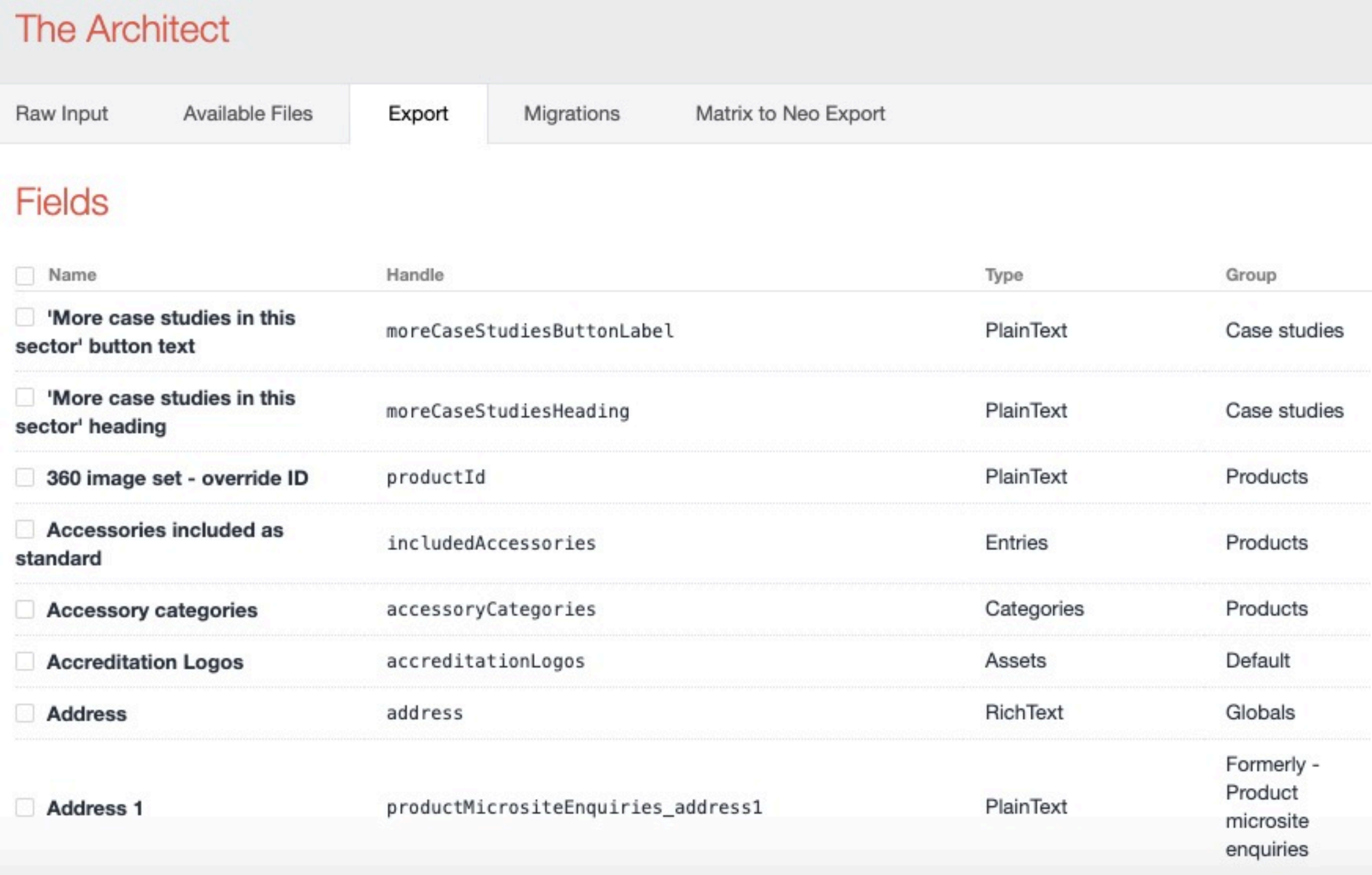


**Architect**

CraftCMS plugin to generate content models from JSON

Free

- Craft 2 The Architect plugin
  - Will import schemas of:
  - Fields
  - Sections
  - Sources
  - Image Transforms
  - Users and User Groups
  - Globals
  - Categories
  - Routes
  - Tags

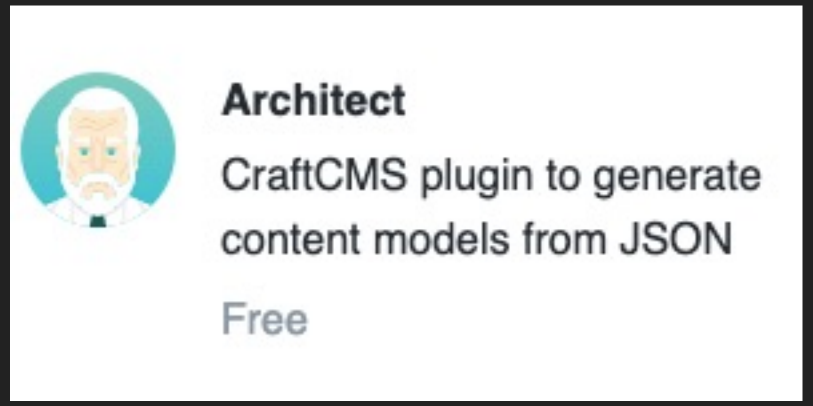


The Architect

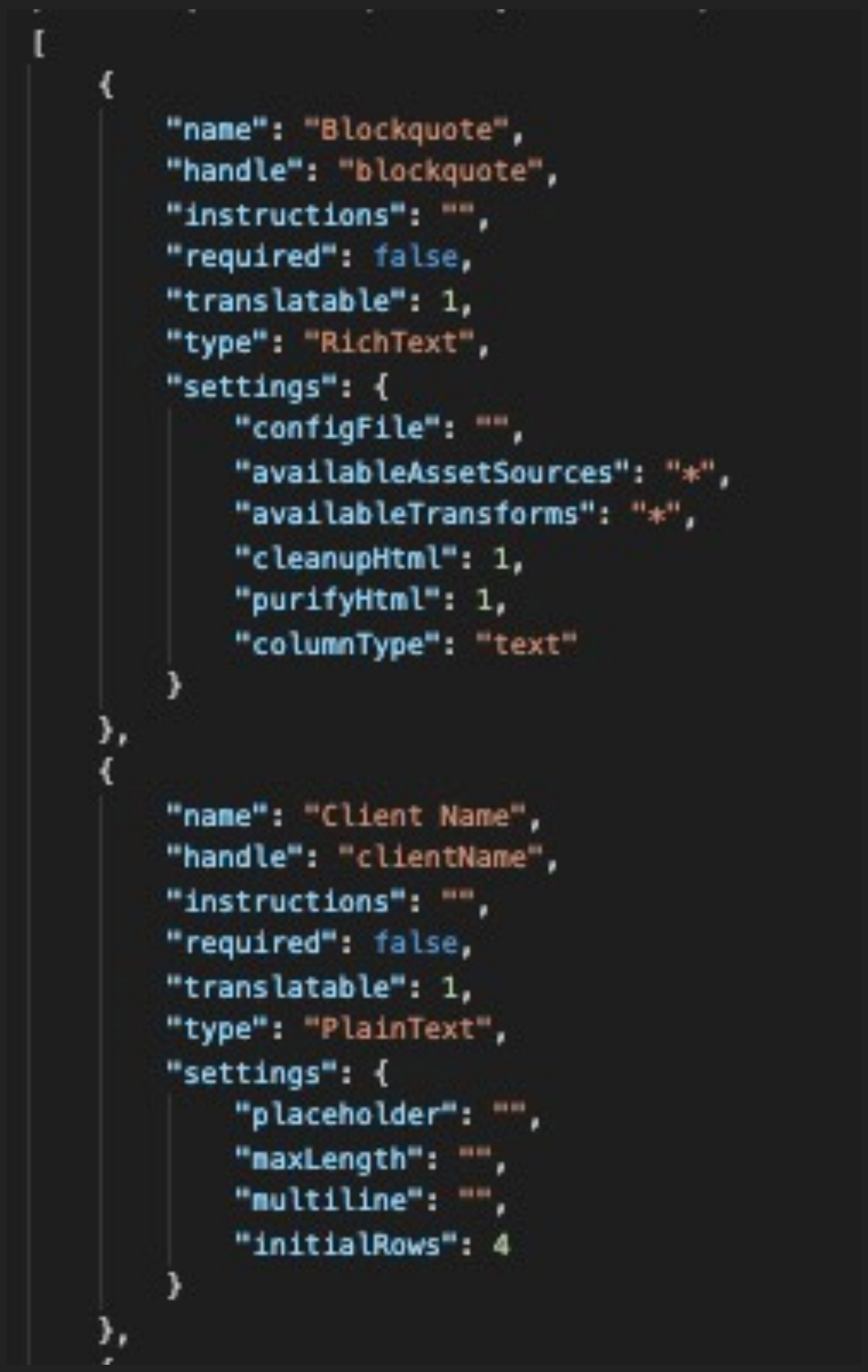
Fields

| ☐ Name | Handle | Type | Group |
|---|---|---|---|
| ☐ 'More case studies in this sector' button text | `moreCaseStudiesButtonLabel` | PlainText | Case studies |
| ☐ 'More case studies in this sector' heading | `moreCaseStudiesHeading` | PlainText | Case studies |
| ☐ 360 image set - override ID | `productId` | PlainText | Products |
| ☐ Accessories included as standard | `includedAccessories` | Entries | Products |
| ☐ Accessory categories | `accessoryCategories` | Categories | Products |
| ☐ Accreditation Logos | `accreditationLogos` | Assets | Default |
| ☐ Address | `address` | RichText | Globals |
| ☐ Address 1 | `productMicrositeEnquiries_address1` | PlainText | Formerly - Product microsite enquiries |

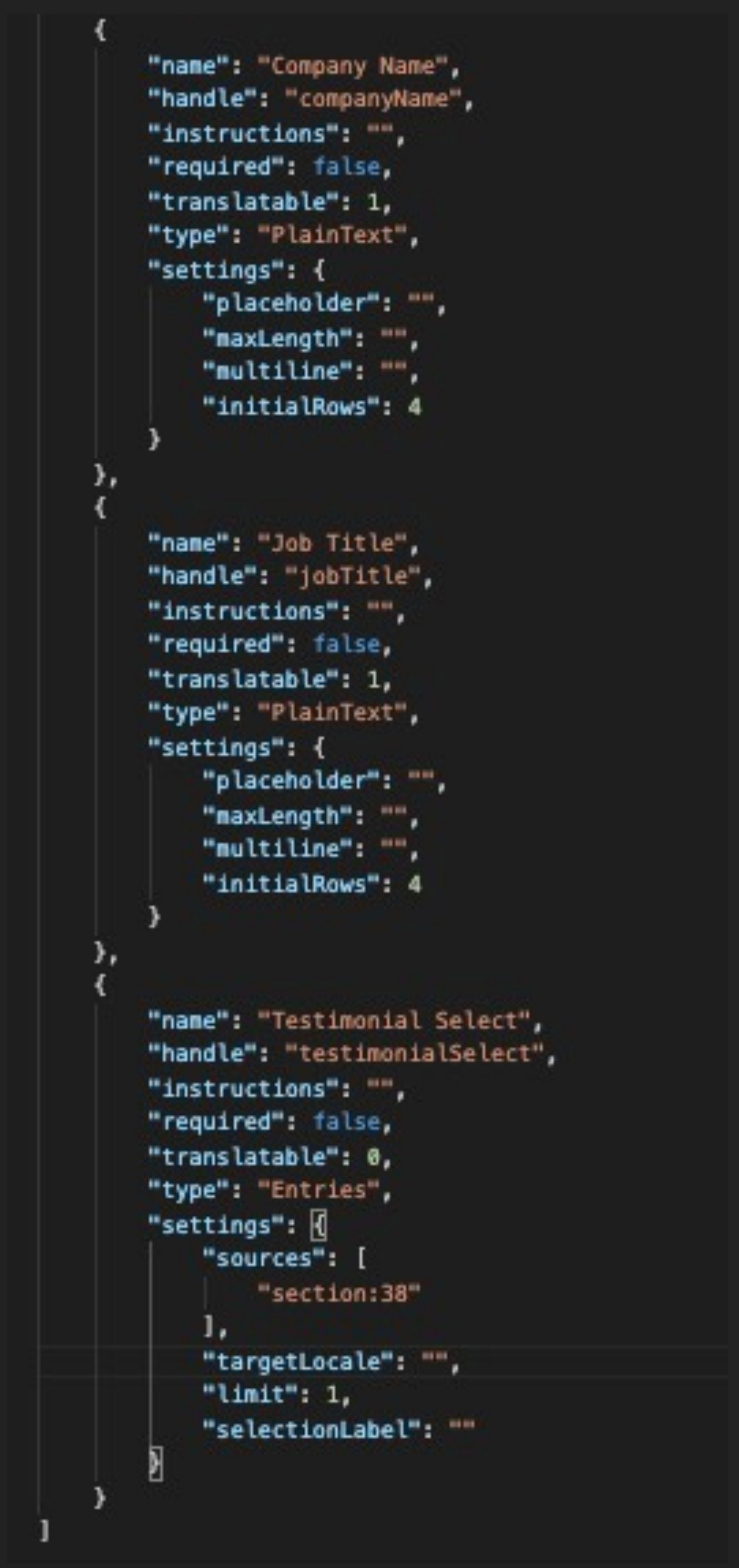
# MIGRATING FIELDS AND STRUCTURE


**Architect**
CraftCMS plugin to generate content models from JSON
Free

- Example export from Architect
  - Be careful to match any changes in scheme for Craft 3
    - Locale is now Site
    - Some settings may not transfer
    - On Craft 3 site, create example entry structure, export JSON to compare differences

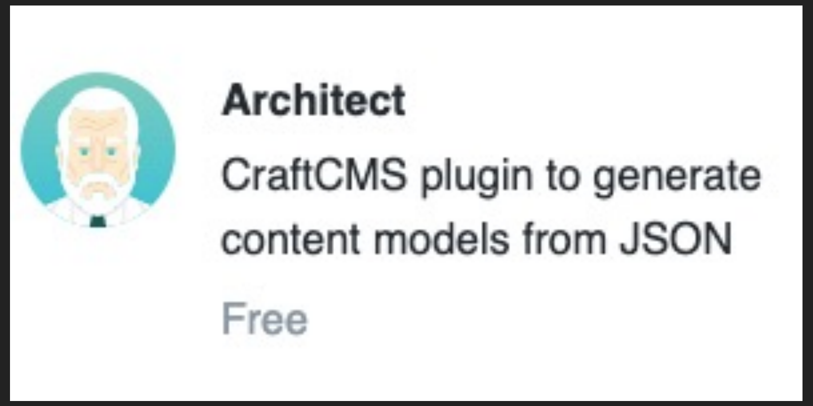


```
[
    {
        "name": "Blockquote",
        "handle": "blockquote",
        "instructions": "",
        "required": false,
        "translatable": 1,
        "type": "RichText",
        "settings": {
            "configFile": "",
            "availableAssetSources": "*",
            "availableTransforms": "*",
            "cleanupHtml": 1,
            "purifyHtml": 1,
            "columnType": "text"
        }
    },
    {
        "name": "Client Name",
        "handle": "clientName",
        "instructions": "",
        "required": false,
        "translatable": 1,
        "type": "PlainText",
        "settings": {
            "placeholder": "",
            "maxLength": "",
            "multiline": "",
            "initialRows": 4
        }
    },
```



```
    {
        "name": "Company Name",
        "handle": "companyName",
        "instructions": "",
        "required": false,
        "translatable": 1,
        "type": "PlainText",
        "settings": {
            "placeholder": "",
            "maxLength": "",
            "multiline": "",
            "initialRows": 4
        }
    },
    {
        "name": "Job Title",
        "handle": "jobTitle",
        "instructions": "",
        "required": false,
        "translatable": 1,
        "type": "PlainText",
        "settings": {
            "placeholder": "",
            "maxLength": "",
            "multiline": "",
            "initialRows": 4
        }
    },
    {
        "name": "Testimonial Select",
        "handle": "testimonialSelect",
        "instructions": "",
        "required": false,
        "translatable": 0,
        "type": "Entries",
        "settings": {
            "sources": [
                "section:38"
            ],
            "targetLocale": "",
            "limit": 1,
            "selectionLabel": ""
        }
    }
]
```

# MIGRATING FIELDS AND STRUCTURE



**Architect**

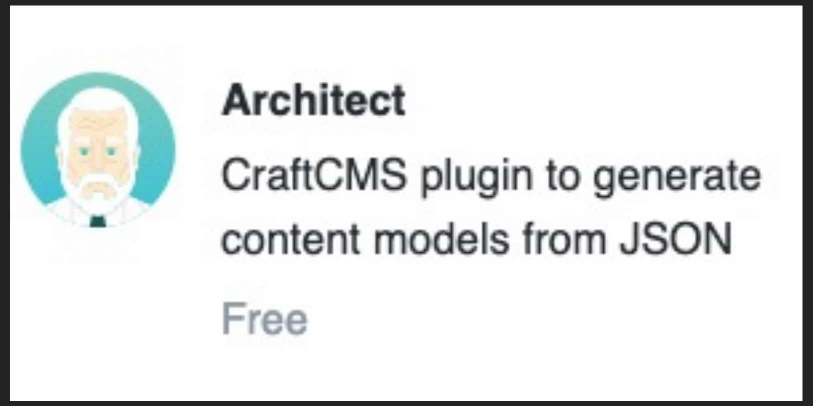
CraftCMS plugin to generate content models from JSON

Free

- Some cautions when using The Architect:
  - Perform imports on Craft 3 in suggested order to prevent missing relationships
  - 1) Asset Sources
  - 2) Image Transforms
  - 3) User Groups
  - 4) Users
  - 5) Tags
  - 6) Categories (may require additional pass after fields)
  - 7) Globals (may require additional pass after fields)
  - 8) Sections (may require additional pass after fields)
  - 9) Fields
  - 10) Rerun your second pass groups

    (to match any associations that may not have been made on initial pass)

# MIGRATING FIELDS AND STRUCTURE



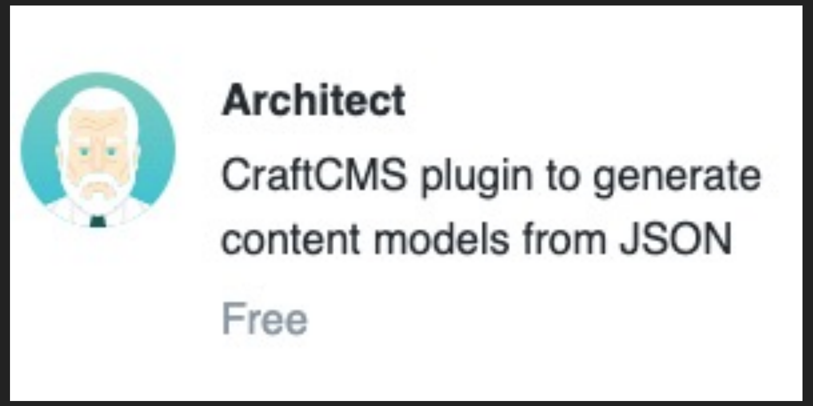
**Architect**

CraftCMS plugin to generate content models from JSON

Free

- Approaching as suggested prevents:
    - Image fields missing their image sources
    - Matrix blocks missing their block fields (this still might happen due to some Craft wackiness, be careful)
    - Category fields missing their category groups and entries
    - Other madness that will cause grey hairs

# MIGRATING FIELDS AND STRUCTURE



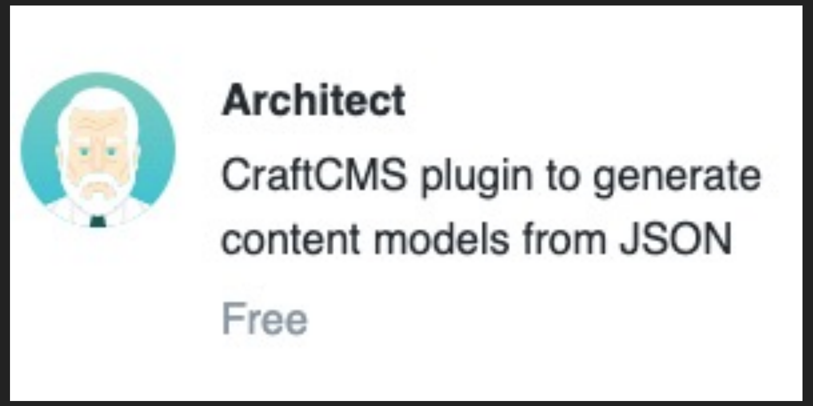
**Architect**

CraftCMS plugin to generate content models from JSON

Free

- Matrix to Neo (magic)
  - If you prefer NEO over complex matrices this tool works wonders to match up block structure
  - Be careful with NEO you can get very complex very quickly
  - Requires that NEO already installed on the new 3 instance

# MIGRATING FIELDS AND STRUCTURE



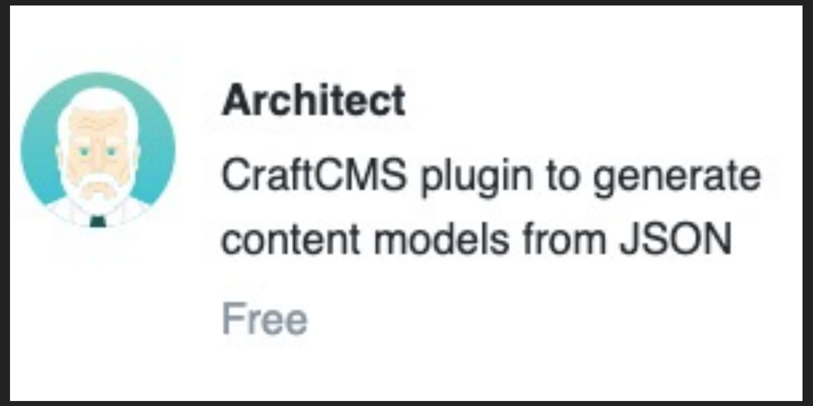
**Architect**

CraftCMS plugin to generate content models from JSON

Free

- Don't use migrations
  - These are intended for migrations within a site, not for transferring content models between sites
- Check the Craft 2 version GitHub issues if you run into anything
  - *Plugin has been archived, so you may have to resolve any issues manually*
- DONT RELY ON ARCHITECT TO ROLLBACK AND USE ITS DB BACKUP AS YOUR ONLY BACKUP *you'll have a bad time real fast*

# MIGRATING FIELDS AND STRUCTURE



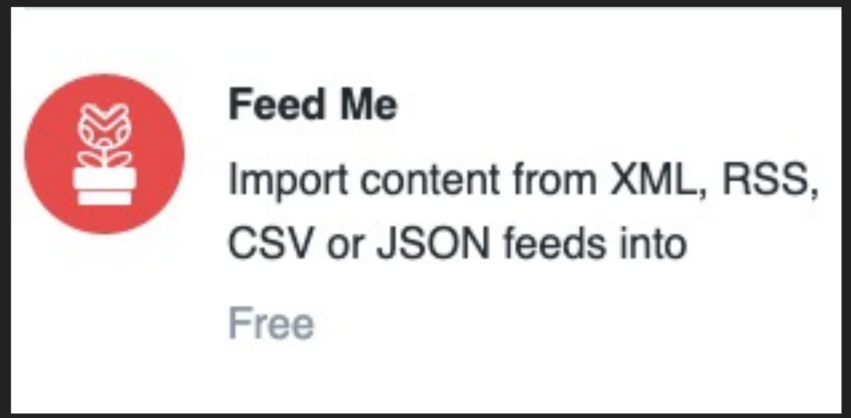
**Architect**

CraftCMS plugin to generate content models from JSON

Free

- Singles will need to be migrated by hand, including field structure :(
- While Architect for Craft 2 can't give you a schema export for Sites (since they don’t exist in 2) you can use their JSON or YAML examples to bake your own solution for locales to sites
  - Unless this is a huge amount of locales to sites, I recommend doing this manually just to save your self some time

# MIGRATING CONTENT



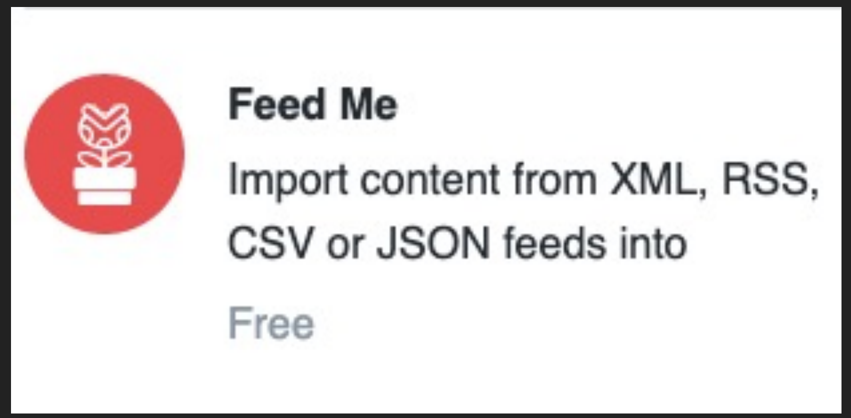
**Feed Me**

Import content from XML, RSS, CSV or JSON feeds into

Free

- Feed Me by Craft
    - Can handle imports of JSON/XML structured data from Craft
    - Be sure to follow Feed Me guidance on structure of your fields
        - Use Value and not Label for lists, radio, etc.
        - Don't use entry IDs for relationships because they won't match instead, use the titles or slugs of the entries

# MIGRATING CONTENT



**Feed Me**

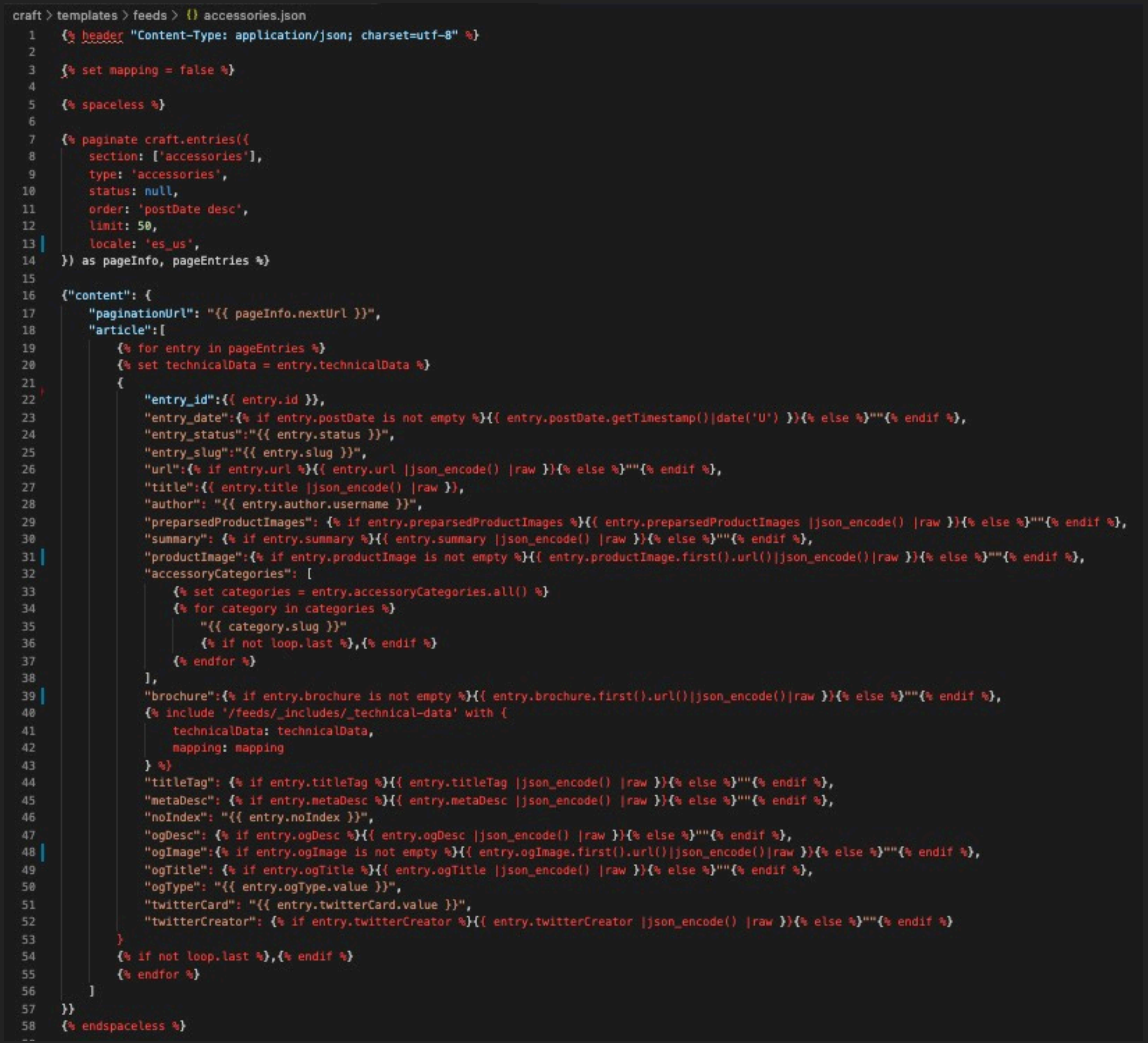
Import content from XML, RSS, CSV or JSON feeds into

Free

- Add a field to your entries on Craft 3 to hold the Craft 2 entry IDs
  - You can then use this field value to check for duplicates when/if you have to rerun the feed to update entries
- Pay extra attention to your Matrix fields when you are structuring them
- Beware that Feed Me doesn't have a way to import the status of blocks, so you will get inactive blocks included in your feed
- When importing Users, you can't import their passwords. Recommendation is to set all users to trigger a password reset on the next login attempt
- Feed Me does NOT support some entry types created by certain plugins, notably SEOMatic data
- Best to follow the same pattern you used when/if you imported fields and structures with Architect

# MIGRATING CONTENT

- Example JSON schema
    - Paginate entries
    - Include locales if needed
    - Spaceless tags
    - Abstract Matrices to reusable templates
    - JSON encode strings



```
{% header "Content-Type: application/json; charset=utf-8" %}

{% set mapping = false %}

{% spaceless %}

{% paginate craft.entries({
    section: ['accessories'],
    type: 'accessories',
    status: null,
    order: 'postDate desc',
    limit: 50,
    locale: 'es_us',
}) as pageInfo, pageEntries %}

{"content": {
    "paginationUrl": "{{ pageInfo.nextUrl }}",
    "article":[
        {% for entry in pageEntries %}
        {% set technicalData = entry.technicalData %}
        {
            "entry_id":{{ entry.id }},
            "entry_date":{% if entry.postDate is not empty %}{{ entry.postDate.getTimestamp()|date('U') }}{% else %}""{% endif %},
            "entry_status":"{{ entry.status }}",
            "entry_slug":"{{ entry.slug }}",
            "url":{% if entry.url %}{{ entry.url |json_encode() |raw }}{% else %}""{% endif %},
            "title":{{ entry.title |json_encode() |raw }},
            "author": "{{ entry.author.username }}",
            "preparsedProductImages": {% if entry.preparsedProductImages %}{{ entry.preparsedProductImages |json_encode() |raw }}{% else %}""{% endif %},
            "summary": {% if entry.summary %}{{ entry.summary |json_encode() |raw }}{% else %}""{% endif %},
            "productImage":{% if entry.productImage is not empty %}{{ entry.productImage.first().url()|json_encode()|raw }}{% else %}""{% endif %},
            "accessoryCategories": [
                {% set categories = entry.accessoryCategories.all() %}
                {% for category in categories %}
                    "{{ category.slug }}"
                    {% if not loop.last %},{% endif %}
                {% endfor %}
            ],
            "brochure":{% if entry.brochure is not empty %}{{ entry.brochure.first().url()|json_encode()|raw }}{% else %}""{% endif %},
            {% include '/feeds/_includes/_technical-data' with {
                technicalData: technicalData,
                mapping: mapping
            } %}
            "titleTag": {% if entry.titleTag %}{{ entry.titleTag |json_encode() |raw }}{% else %}""{% endif %},
            "metaDesc": {% if entry.metaDesc %}{{ entry.metaDesc |json_encode() |raw }}{% else %}""{% endif %},
            "noIndex": "{{ entry.noIndex }}",
            "ogDesc": {% if entry.ogDesc %}{{ entry.ogDesc |json_encode() |raw }}{% else %}""{% endif %},
            "ogImage":{% if entry.ogImage is not empty %}{{ entry.ogImage.first().url()|json_encode()|raw }}{% else %}""{% endif %},
            "ogTitle": {% if entry.ogTitle %}{{ entry.ogTitle |json_encode() |raw }}{% else %}""{% endif %},
            "ogType": "{{ entry.ogType.value }}",
            "twitterCard": "{{ entry.twitterCard.value }}",
            "twitterCreator": {% if entry.twitterCreator %}{{ entry.twitterCreator |json_encode() |raw }}{% else %}""{% endif %}
        }
        {% if not loop.last %},{% endif %}
        {% endfor %}
    ]
}}
{% endspaceless %}
```

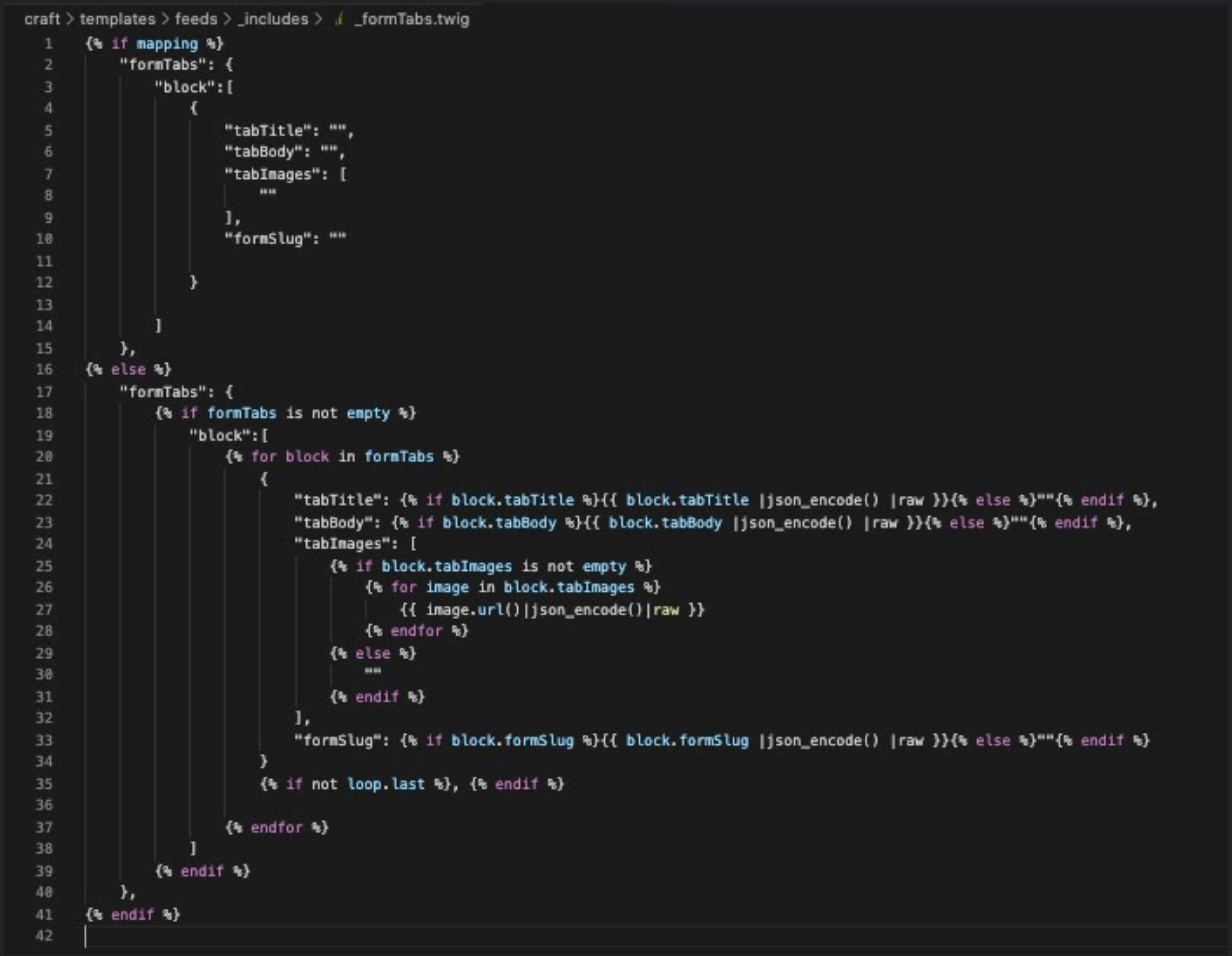
**Feed Me**

Import content from XML, RSS, CSV or JSON feeds into

Free

# MIGRATING CONTENT

- Example Matrix schema
  - DON'T name your blocks Feed Me will loop the block structure in Craft 3 site, you will map fields.
  - If block field names are same, rename them to be unique.
  - Arrays can be empty
  - Loop index for JSON



craft > templates > feeds > _includes > _formTabs.twig

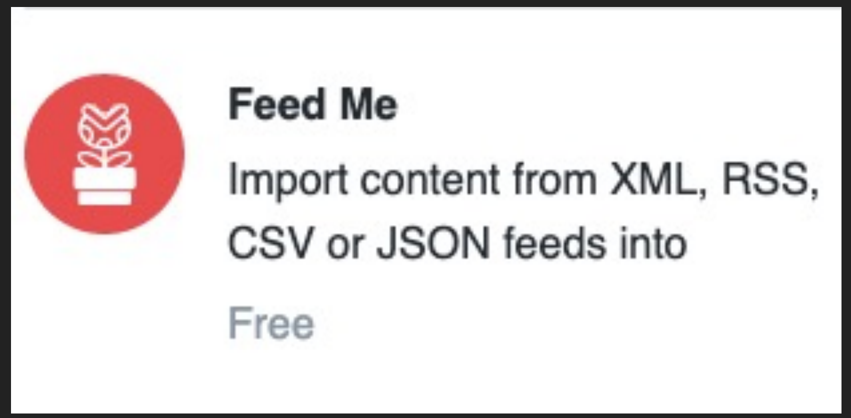
```
{% if mapping %}
    "formTabs": {
        "block":[
            {
                "tabTitle": "",
                "tabBody": "",
                "tabImages": [
                    ""
                ],
                "formSlug": ""

            }

        ]
    },
{% else %}
    "formTabs": {
        {% if formTabs is not empty %}
            "block":[
                {% for block in formTabs %}
                    {
                        "tabTitle": {% if block.tabTitle %}{{ block.tabTitle |json_encode() |raw }}{% else %}""{% endif %},
                        "tabBody": {% if block.tabBody %}{{ block.tabBody |json_encode() |raw }}{% else %}""{% endif %},
                        "tabImages": [
                            {% if block.tabImages is not empty %}
                                {% for image in block.tabImages %}
                                    {{ image.url()|json_encode()|raw }}
                                {% endfor %}
                            {% else %}
                                ""
                            {% endif %}
                        ],
                        "formSlug": {% if block.formSlug %}{{ block.formSlug |json_encode() |raw }}{% else %}""{% endif %}
                    }
                    {% if not loop.last %}, {% endif %}

                {% endfor %}
            ]
        {% endif %}
    },
{% endif %}

```

**Feed Me**

Import content from XML, RSS, CSV or JSON feeds into

Free

# MIGRATING CONTENT

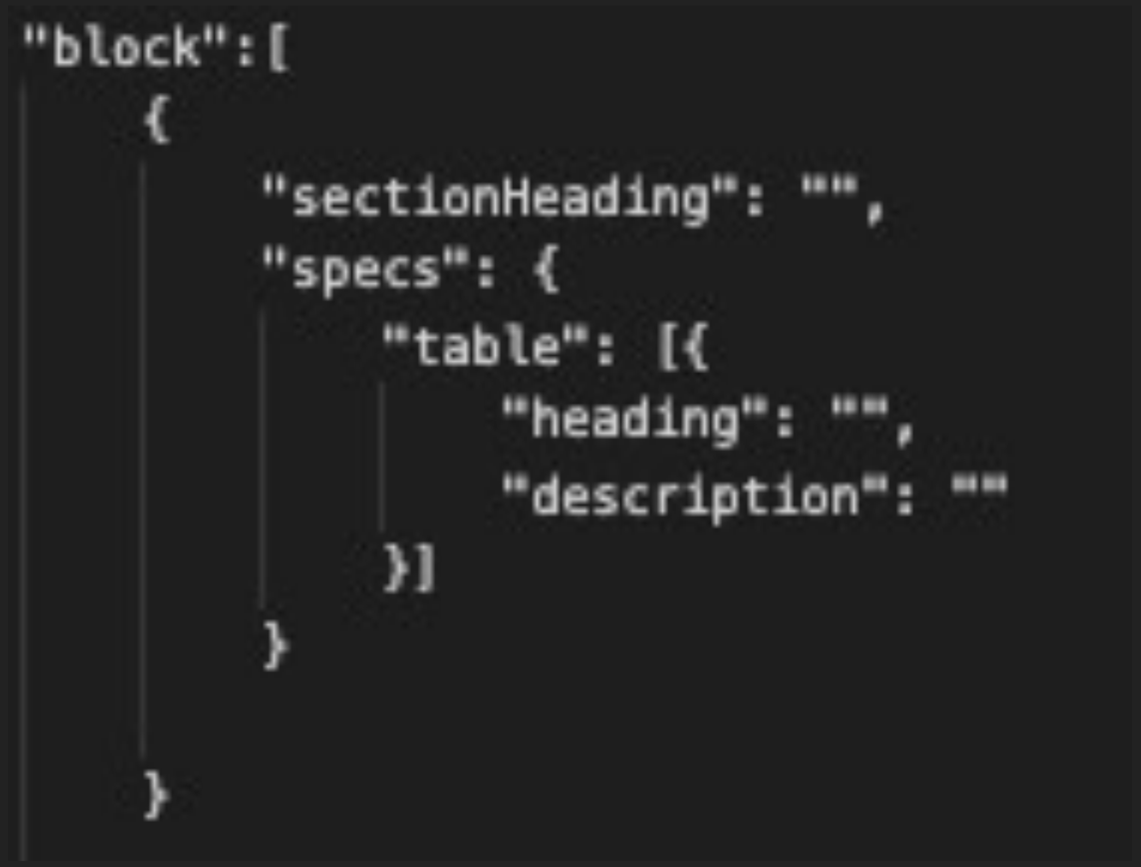


**Feed Me**

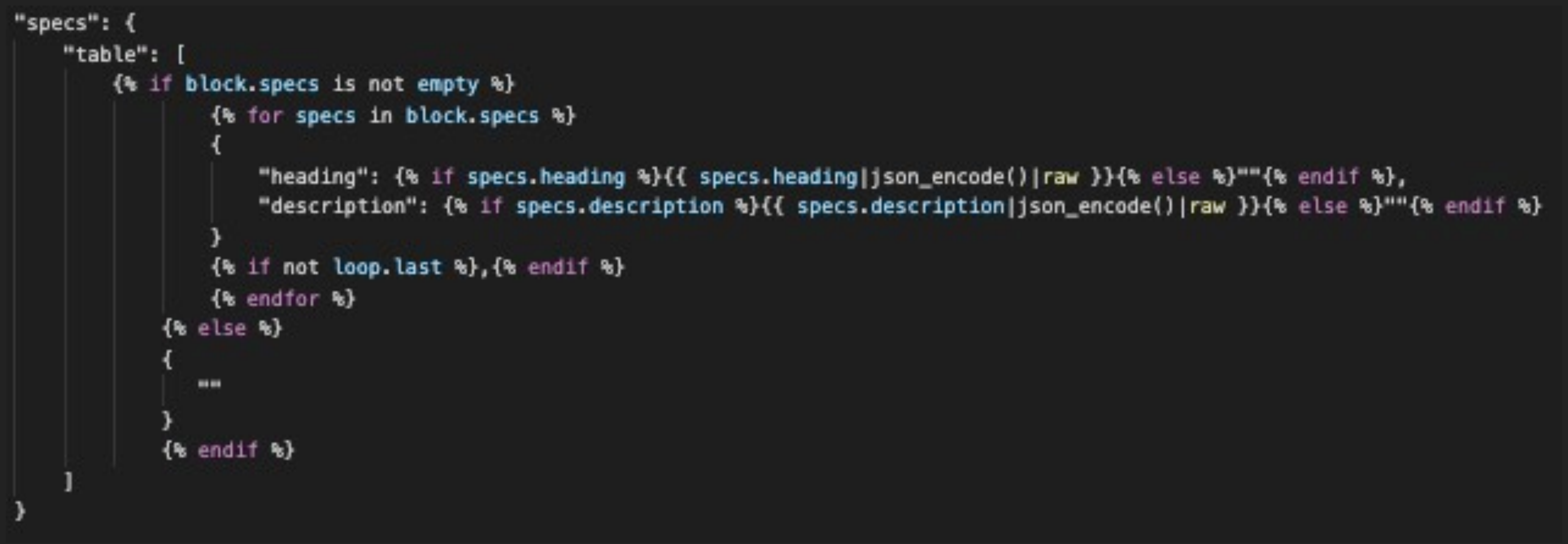
Import content from XML, RSS, CSV or JSON feeds into

Free

- Example Matrix schema
    - Table fields require special structure.



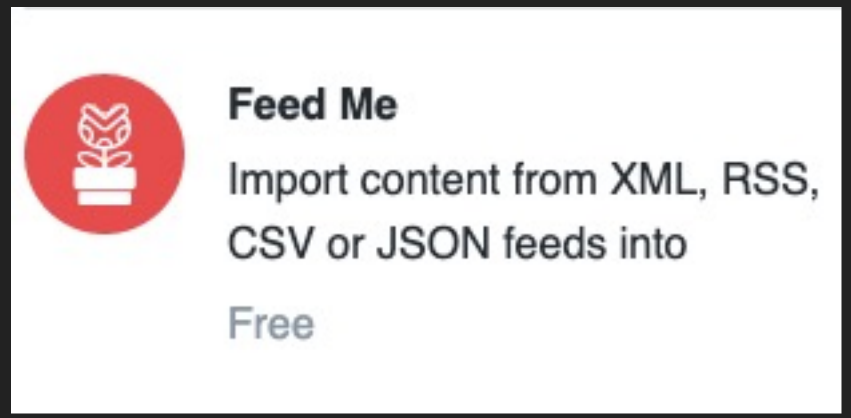
```
"block":[
    {
        "sectionHeading": "",
        "specs": {
            "table": [{
                "heading": "",
                "description": ""
            }]
        }

    }
```



```
"specs": {
    "table": [
        {% if block.specs is not empty %}
                {% for specs in block.specs %}
                {
                    "heading": {% if specs.heading %}{{ specs.heading|json_encode()|raw }}{% else %}""{% endif %},
                    "description": {% if specs.description %}{{ specs.description|json_encode()|raw }}{% else %}""{% endif %}
                }
                {% if not loop.last %},{% endif %}
                {% endfor %}
            {% else %}
            {
               ""
            }
            {% endif %}
    ]
}
```

# MIGRATING CONTENT

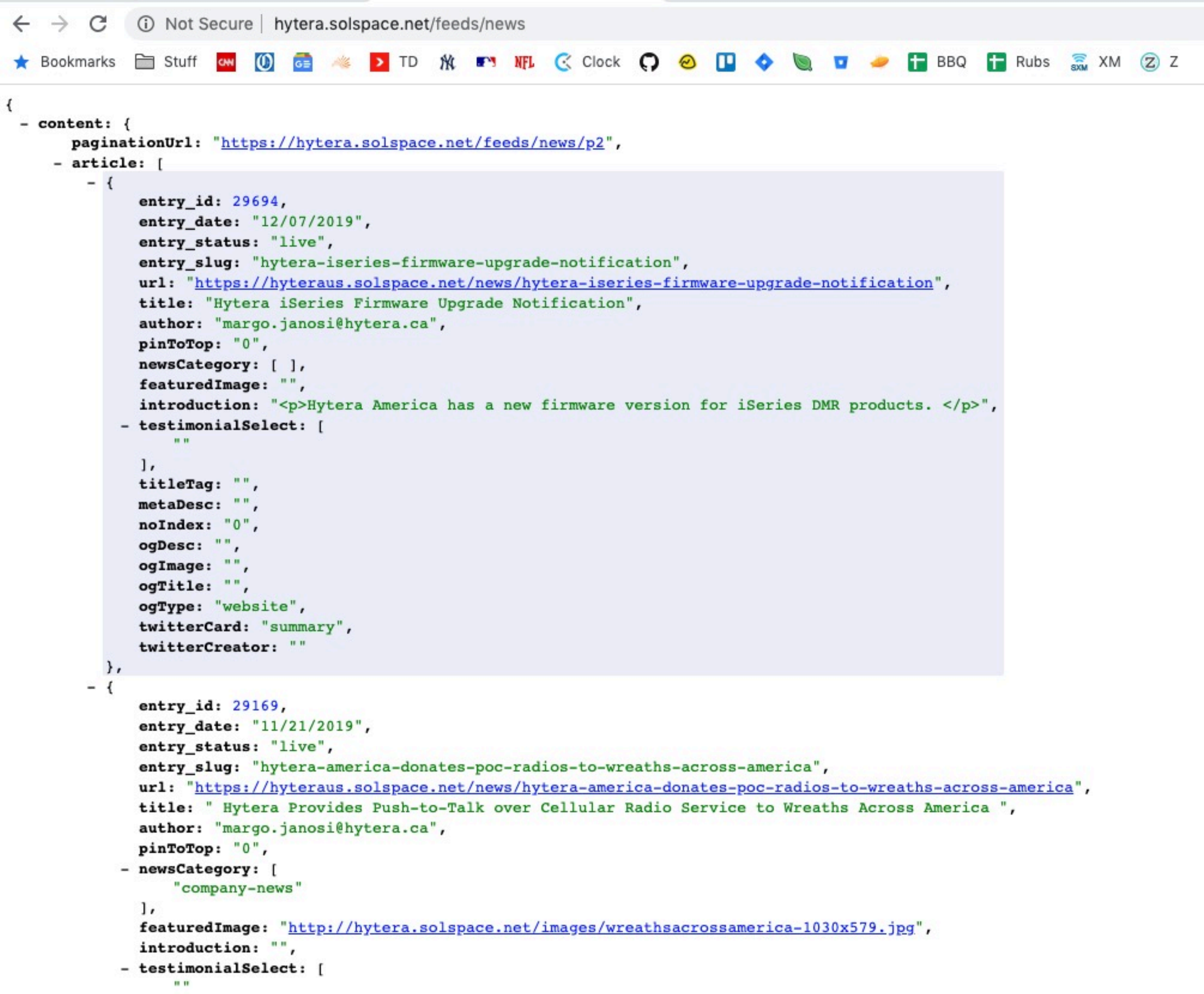
- Compiled JSON feed example:



**Feed Me**

Import content from XML, RSS, CSV or JSON feeds into

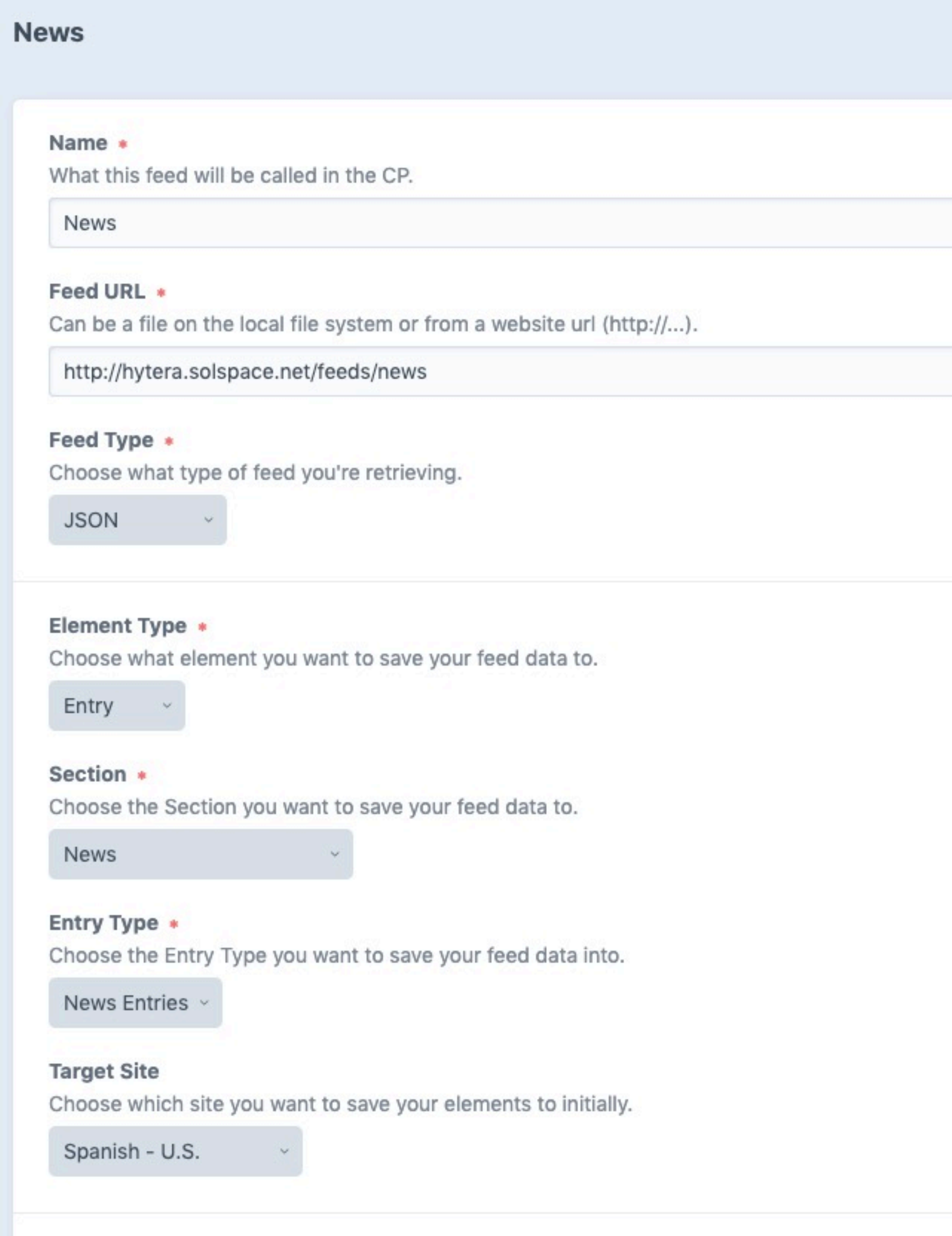
Free



```
{
 - content: {
       paginationUrl: "https://hytera.solspace.net/feeds/news/p2",
     - article: [
         - {
               entry_id: 29694,
               entry_date: "12/07/2019",
               entry_status: "live",
               entry_slug: "hytera-iseries-firmware-upgrade-notification",
               url: "https://hyteraus.solspace.net/news/hytera-iseries-firmware-upgrade-notification",
               title: "Hytera iSeries Firmware Upgrade Notification",
               author: "margo.janosi@hytera.ca",
               pinToTop: "0",
               newsCategory: [ ],
               featuredImage: "",
               introduction: "<p>Hytera America has a new firmware version for iSeries DMR products. </p>",
             - testimonialSelect: [
                   ""
               ],
               titleTag: "",
               metaDesc: "",
               noIndex: "0",
               ogDesc: "",
               ogImage: "",
               ogTitle: "",
               ogType: "website",
               twitterCard: "summary",
               twitterCreator: ""
           },
         - {
               entry_id: 29169,
               entry_date: "11/21/2019",
               entry_status: "live",
               entry_slug: "hytera-america-donates-poc-radios-to-wreaths-across-america",
               url: "https://hyteraus.solspace.net/news/hytera-america-donates-poc-radios-to-wreaths-across-america",
               title: " Hytera Provides Push-to-Talk over Cellular Radio Service to Wreaths Across America ",
               author: "margo.janosi@hytera.ca",
               pinToTop: "0",
             - newsCategory: [
                   "company-news"
               ],
               featuredImage: "http://hytera.solspace.net/images/wreathsacrossamerica-1030x579.jpg",
               introduction: "",
             - testimonialSelect: [
                   ""
```

# MIGRATING CONTENT

- Feed Me step 1



**News**

**Name** *
What this feed will be called in the CP.
News

**Feed URL** *
Can be a file on the local file system or from a website url (http://...).
http://hytera.solspace.net/feeds/news

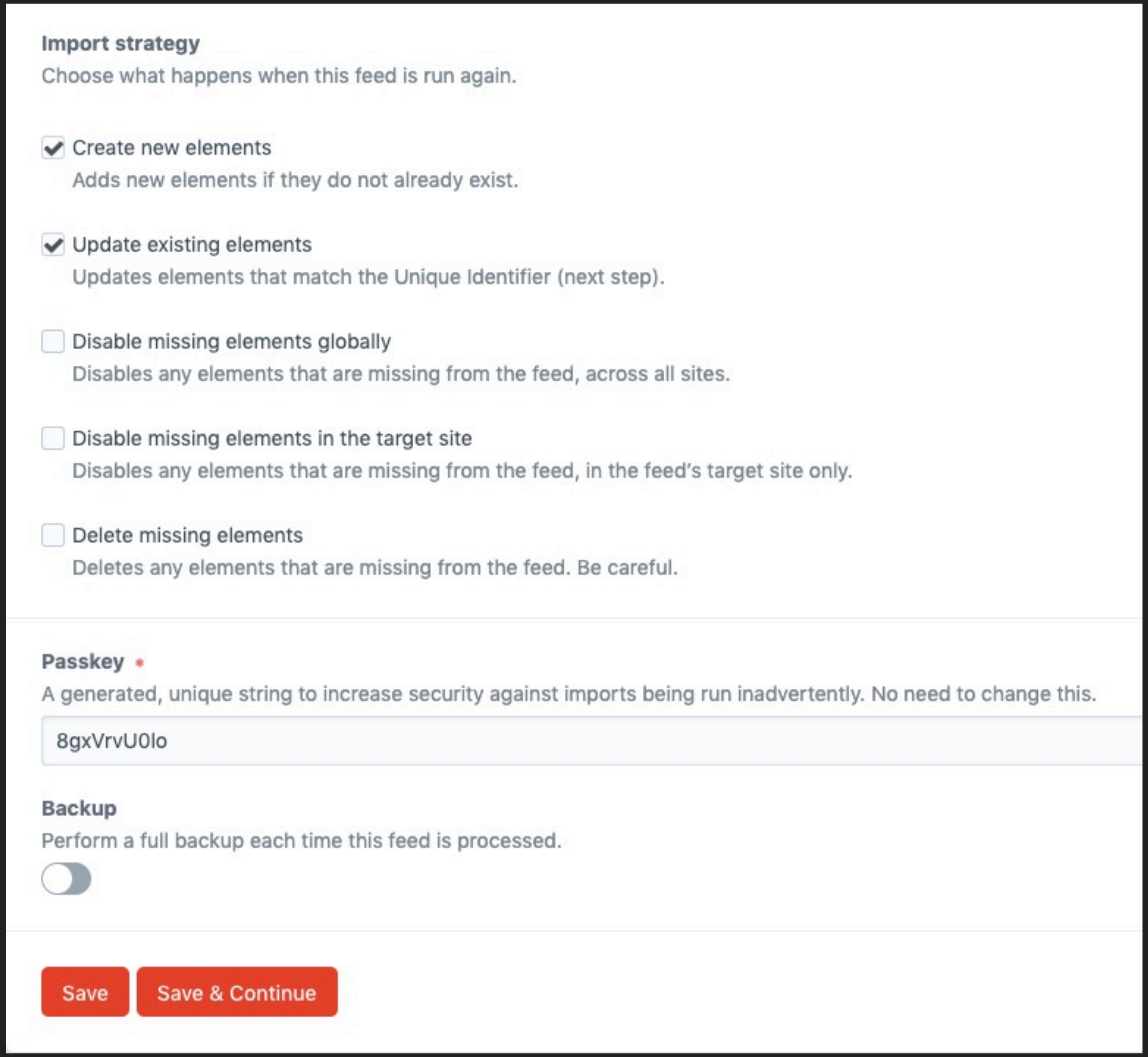
**Feed Type** *
Choose what type of feed you're retrieving.
JSON

**Element Type** *
Choose what element you want to save your feed data to.
Entry

**Section** *
Choose the Section you want to save your feed data to.
News

**Entry Type** *
Choose the Entry Type you want to save your feed data into.
News Entries

**Target Site**
Choose which site you want to save your elements to initially.
Spanish - U.S.



**Import strategy**
Choose what happens when this feed is run again.

☑ Create new elements
Adds new elements if they do not already exist.

☑ Update existing elements
Updates elements that match the Unique Identifier (next step).

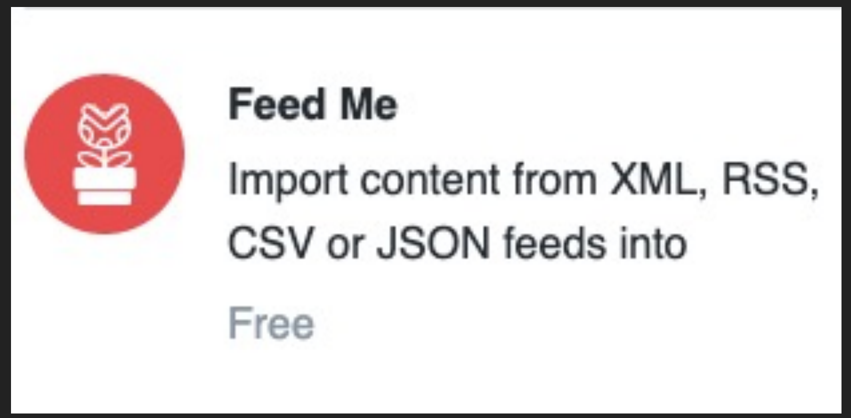
☐ Disable missing elements globally
Disables any elements that are missing from the feed, across all sites.

☐ Disable missing elements in the target site
Disables any elements that are missing from the feed, in the feed's target site only.

☐ Delete missing elements
Deletes any elements that are missing from the feed. Be careful.

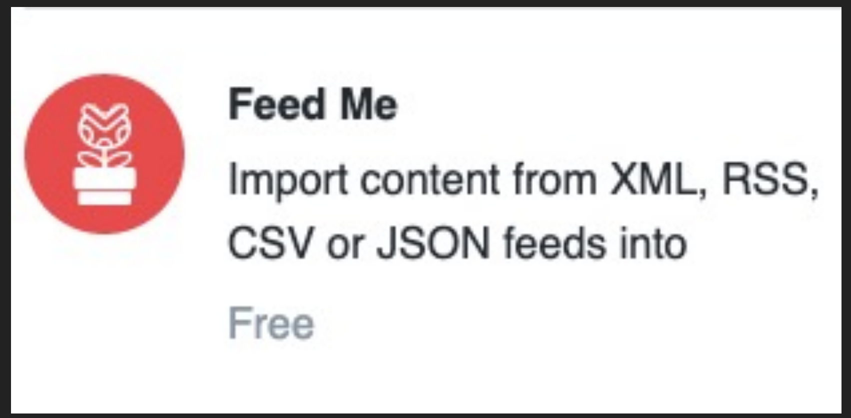
**Passkey** *
A generated, unique string to increase security against imports being run inadvertently. No need to change this.
8gxVrvU0lo

**Backup**
Perform a full backup each time this feed is processed.
○



**Feed Me**
Import content from XML, RSS, CSV or JSON feeds into
Free

# MIGRATING CONTENT

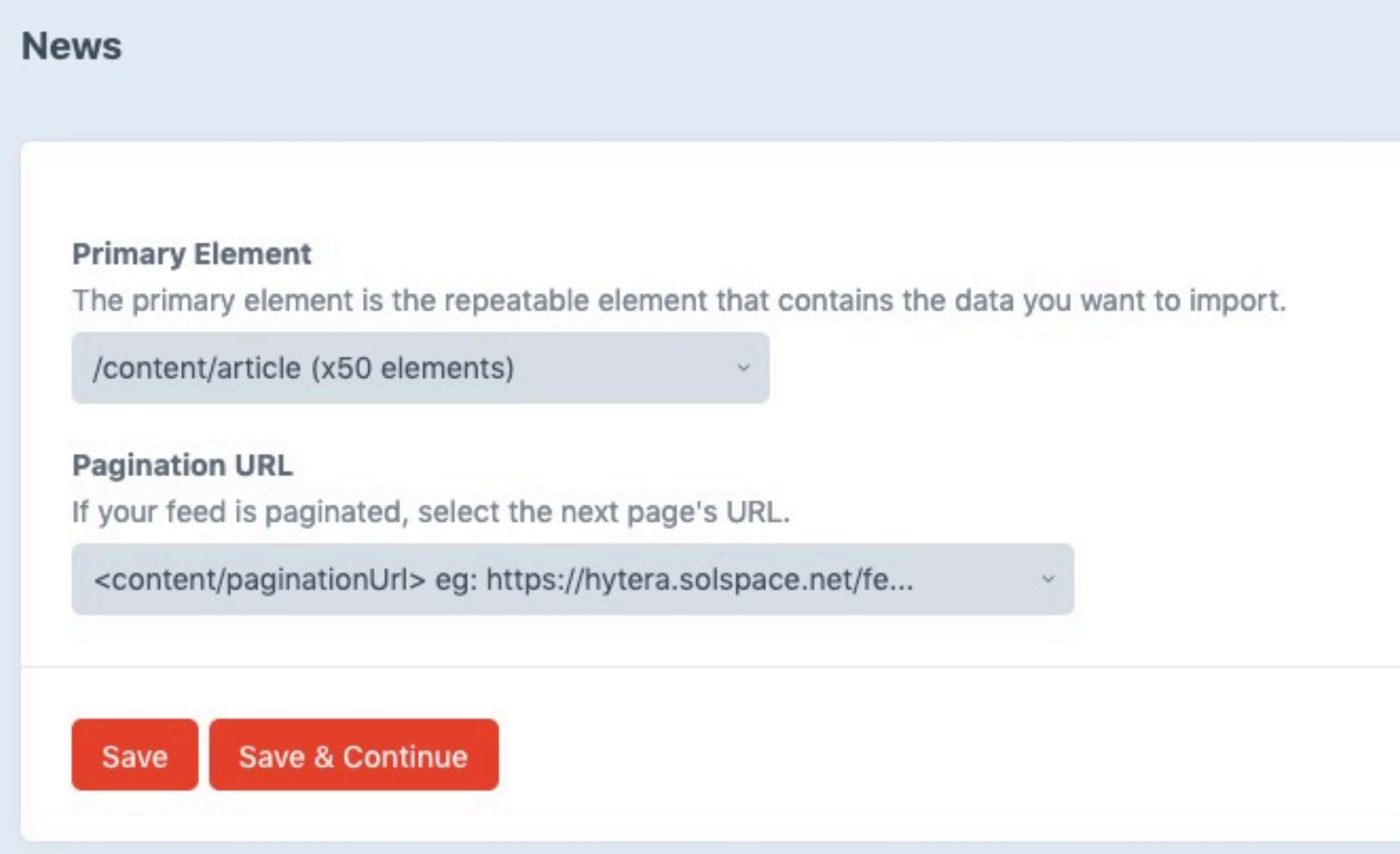


**Feed Me**

Import content from XML, RSS, CSV or JSON feeds into

Free

- Feed Me step 2



**News**

**Primary Element**

The primary element is the repeatable element that contains the data you want to import.

/content/article (x50 elements)

**Pagination URL**

If your feed is paginated, select the next page's URL.

<content/paginationUrl> eg: https://hytera.solspace.net/fe...

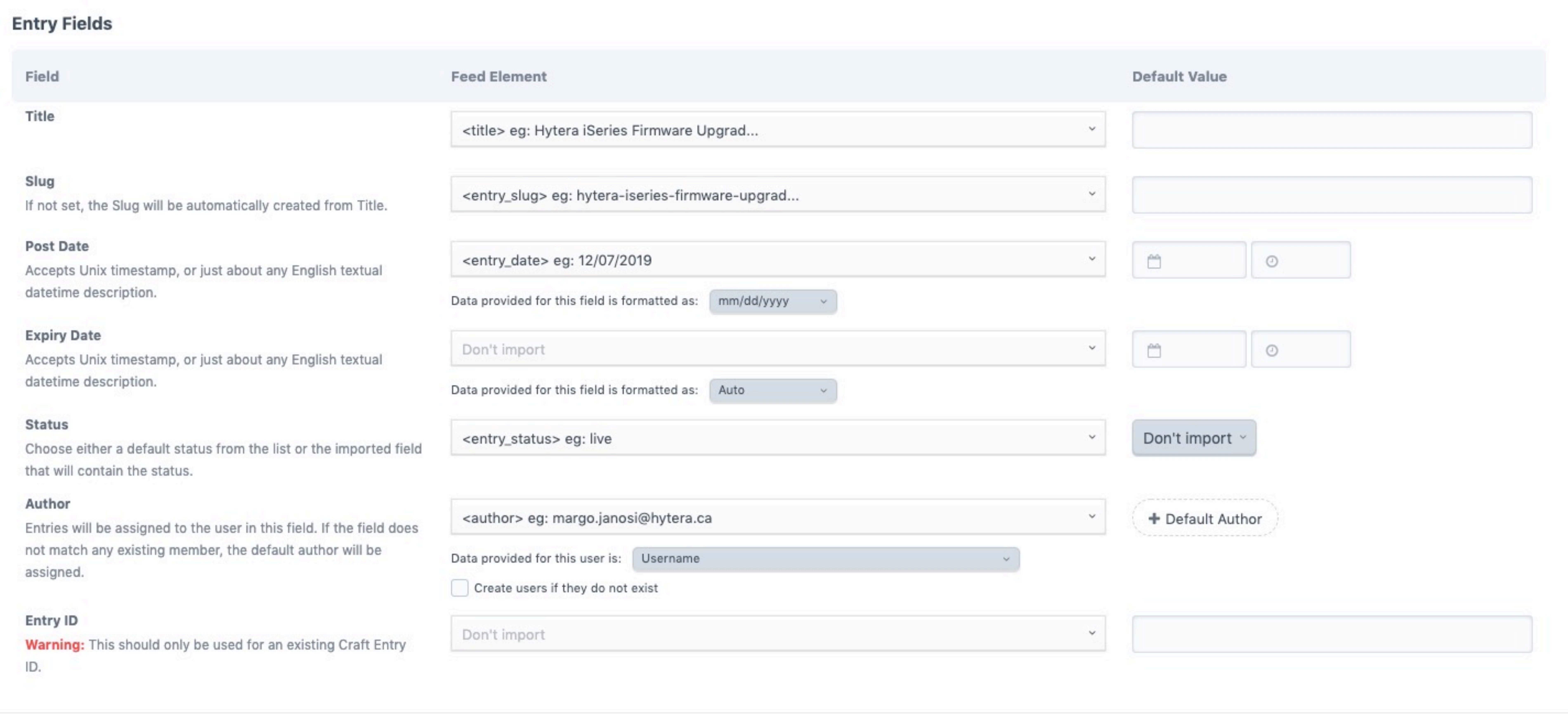
# MIGRATING CONTENT

- Feed Me step 3

**Feed Me**

Import content from XML, RSS, CSV or JSON feeds into

Free



**Entry Fields**

| Field | Feed Element | Default Value |
|---|---|---|
| **Title** | <title> eg: Hytera iSeries Firmware Upgrad... | |
| **Slug**<br>If not set, the Slug will be automatically created from Title. | <entry_slug> eg: hytera-iseries-firmware-upgrad... | |
| **Post Date**<br>Accepts Unix timestamp, or just about any English textual datetime description. | <entry_date> eg: 12/07/2019<br>Data provided for this field is formatted as: mm/dd/yyyy | |
| **Expiry Date**<br>Accepts Unix timestamp, or just about any English textual datetime description. | Don't import<br>Data provided for this field is formatted as: Auto | |
| **Status**<br>Choose either a default status from the list or the imported field that will contain the status. | <entry_status> eg: live | Don't import |
| **Author**<br>Entries will be assigned to the user in this field. If the field does not match any existing member, the default author will be assigned. | <author> eg: margo.janosi@hytera.ca<br>Data provided for this user is: Username<br>☐ Create users if they do not exist | + Default Author |
| **Entry ID**<br>**Warning:** This should only be used for an existing Craft Entry ID. | Don't import | |

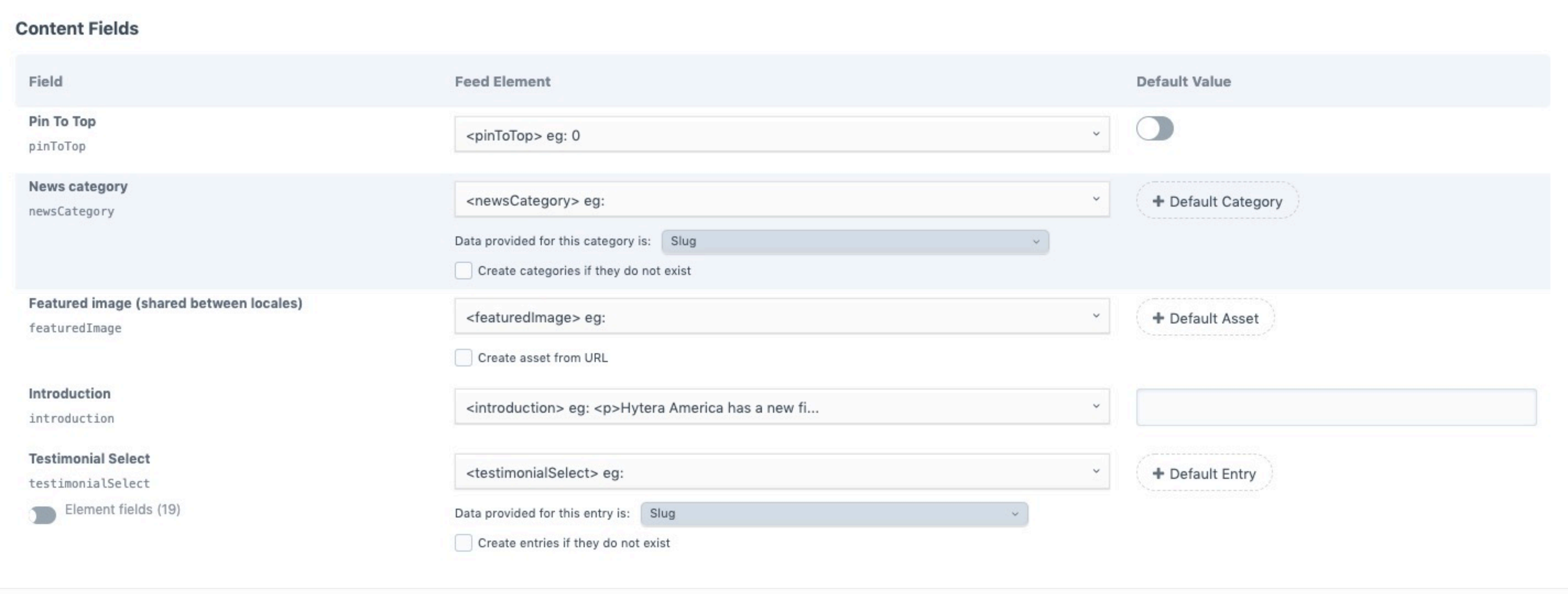
# MIGRATING CONTENT

- Feed Me step 3

**Feed Me**

Import content from XML, RSS, CSV or JSON feeds into

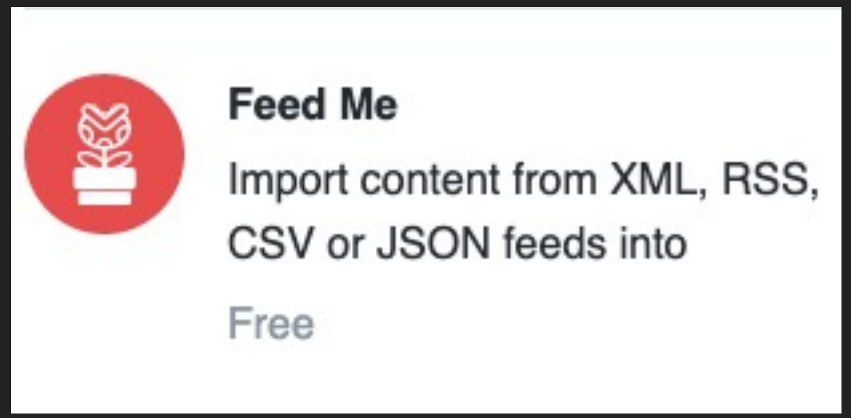
Free



**Content Fields**

| Field | Feed Element | Default Value |
|---|---|---|
| **Pin To Top**<br>`pinToTop` | <pinToTop> eg: 0 | |
| **News category**<br>`newsCategory` | <newsCategory> eg:<br>Data provided for this category is: Slug<br>☐ Create categories if they do not exist | + Default Category |
| **Featured image (shared between locales)**<br>`featuredImage` | <featuredImage> eg:<br>☐ Create asset from URL | + Default Asset |
| **Introduction**<br>`introduction` | <introduction> eg: <p>Hytera America has a new fi... | |
| **Testimonial Select**<br>`testimonialSelect`<br>Element fields (19) | <testimonialSelect> eg:<br>Data provided for this entry is: Slug<br>☐ Create entries if they do not exist | + Default Entry |

# MIGRATING CONTENT

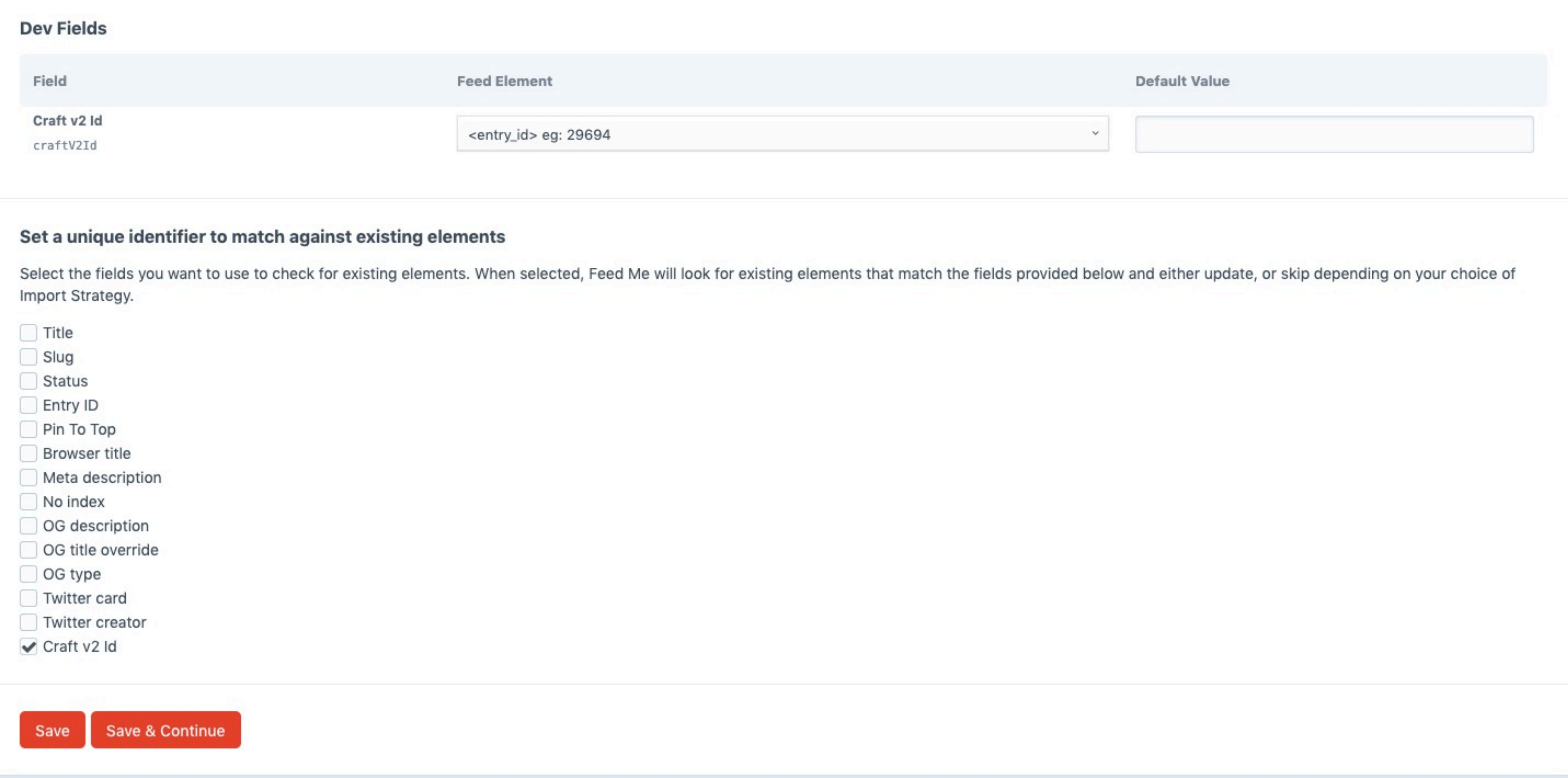


**Feed Me**

Import content from XML, RSS, CSV or JSON feeds into

Free

- Feed Me step 3



### Dev Fields

| Field | Feed Element | Default Value |
|---|---|---|
| **Craft v2 Id**<br>`craftV2Id` | <entry_id> eg: 29694 | |

### Set a unique identifier to match against existing elements

Select the fields you want to use to check for existing elements. When selected, Feed Me will look for existing elements that match the fields provided below and either update, or skip depending on your choice of Import Strategy.

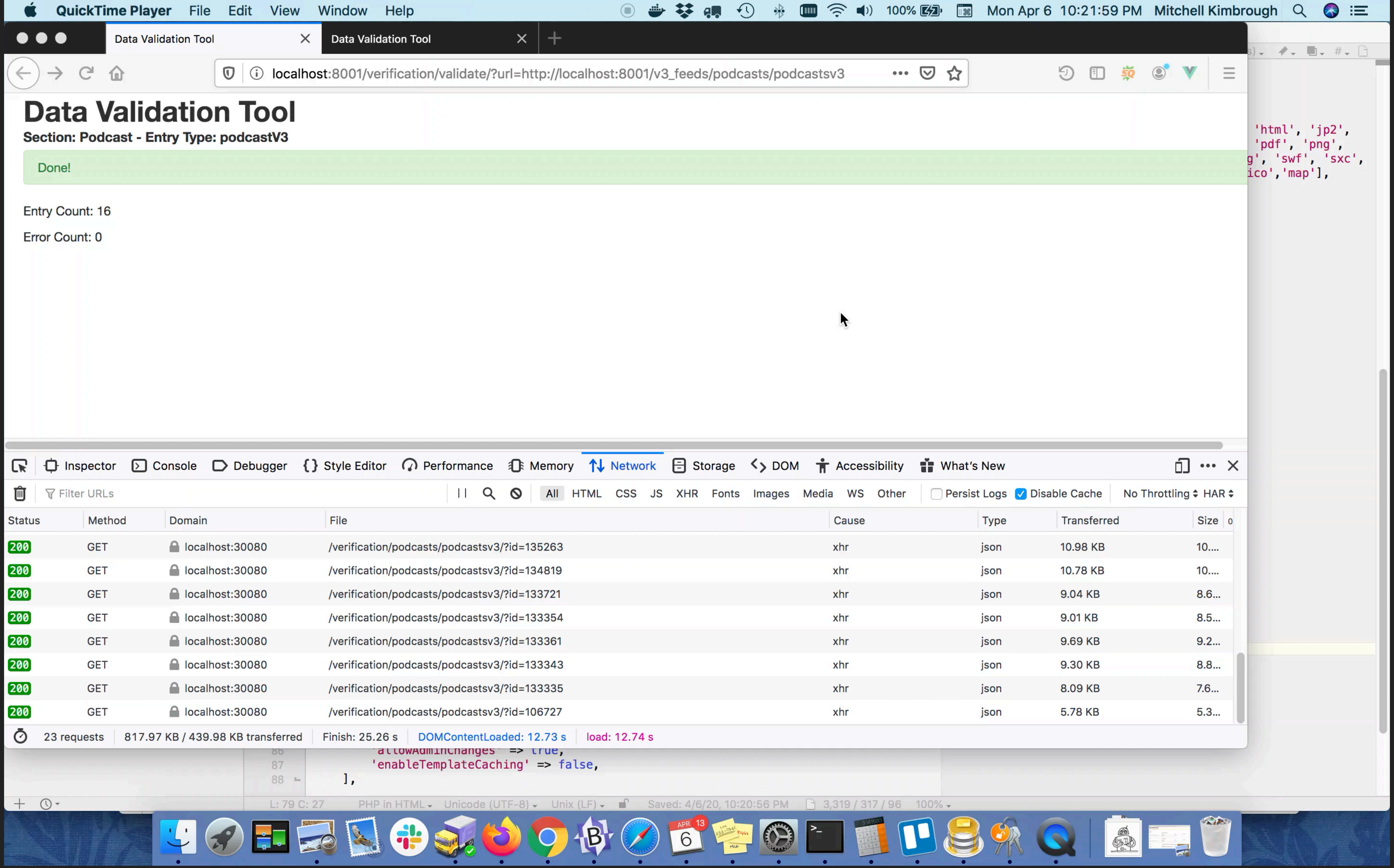
- ☐ Title
- ☐ Slug
- ☐ Status
- ☐ Entry ID
- ☐ Pin To Top
- ☐ Browser title
- ☐ Meta description
- ☐ No index
- ☐ OG description
- ☐ OG title override
- ☐ OG type
- ☐ Twitter card
- ☐ Twitter creator
- ☑ Craft v2 Id

# MULTI-LANGUAGE/SITE SOLUTIONS (LOCALES)

- Locales are now supported via Sites in Craft 3. The PHP constant CRAFT_LOCALE is now CRAFT_SITE.
    - You also now don't have to have a separate index.php and site/locale web root directories.
    - Static localization files now need to be in subdirectories within your root translations/ directory.
- Feed Me can target language/site on reruns of feeds.
    - Initially target the main site/language. You can set FeedMe to first check Title for duplicates.
    - After initial import, edit settings for your JSON feed and target a different language.
    - Then you can edit your FeedMe feed settings to target the appropriate language, and change the Duplicates settings to check ONLY the Craft 2 entry ID field you added earlier.
    - This allows Craft to find each entry, match the Craft 2 ID so its not a new entry (unless it needs to be), and then import the new content to the appropriate language/site.
    - Repeat steps as needed per language.

# DATA VERIFICATION

- Upcoming tool from Solspace to verify imports of large datasets.



QuickTime Player File Edit View Window Help
Mon Apr 6 10:21:59 PM Mitchell Kimbrough
Data Validation Tool
Data Validation Tool
Inspector Console Debugger Style Editor Performance Memory Network Storage DOM Accessibility What's New
All HTML CSS JS XHR Fonts Images Media WS Other
Persist Logs Disable Cache No Throttling HAR

localhost:8001/verification/validate/?url=http://localhost:8001/v3_feeds/podcasts/podcastsv3

## Data Validation Tool

**Section: Podcast - Entry Type: podcastV3**

Done!

Entry Count: 16

Error Count: 0

| Status | Method | Domain | File | Cause | Type | Transferred | Size |
|---|---|---|---|---|---|---|---|
| 200 | GET | localhost:30080 | /verification/podcasts/podcastsv3/?id=135263 | xhr | json | 10.98 KB | 10.... |
| 200 | GET | localhost:30080 | /verification/podcasts/podcastsv3/?id=134819 | xhr | json | 10.78 KB | 10.... |
| 200 | GET | localhost:30080 | /verification/podcasts/podcastsv3/?id=133721 | xhr | json | 9.04 KB | 8.6... |
| 200 | GET | localhost:30080 | /verification/podcasts/podcastsv3/?id=133354 | xhr | json | 9.01 KB | 8.5... |
| 200 | GET | localhost:30080 | /verification/podcasts/podcastsv3/?id=133361 | xhr | json | 9.69 KB | 9.2... |
| 200 | GET | localhost:30080 | /verification/podcasts/podcastsv3/?id=133343 | xhr | json | 9.30 KB | 8.8... |
| 200 | GET | localhost:30080 | /verification/podcasts/podcastsv3/?id=133335 | xhr | json | 8.09 KB | 7.6... |
| 200 | GET | localhost:30080 | /verification/podcasts/podcastsv3/?id=106727 | xhr | json | 5.78 KB | 5.3... |

23 requests | 817.97 KB / 439.98 KB transferred | Finish: 25.26 s | DOMContentLoaded: 12.73 s | load: 12.74 s

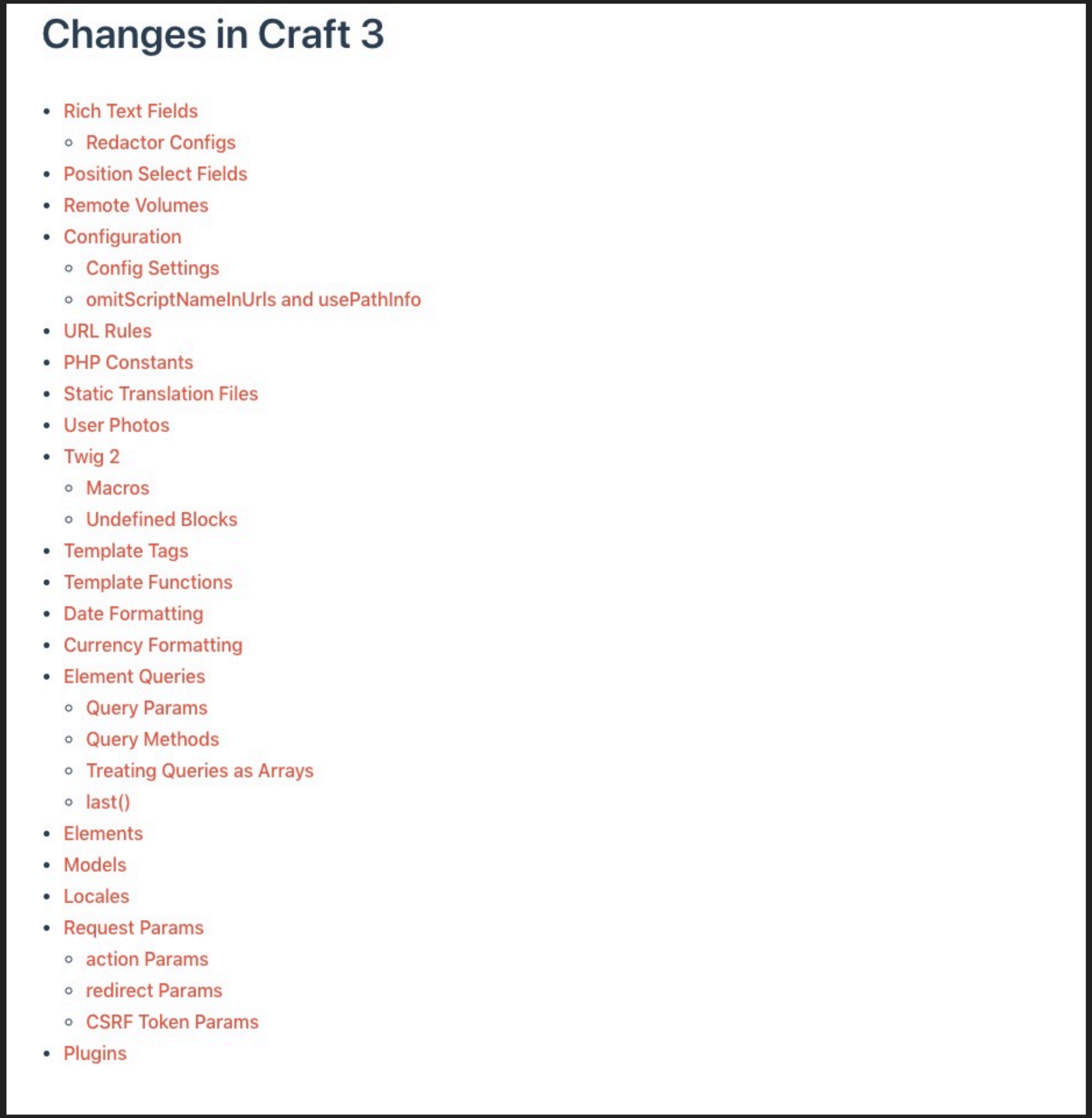
```
        'enableTemplateCaching' => false,
    ],
```

# TEMPLATE UPDATES (BY NO MEANS AN EXHAUSTIVE LIST)

- Craft deprecation errors list is your friend here.
- .all() or .one() on everything.
- URL requests including segment requests have new syntax
- Macros must be explicitly imported in the templates that use them, each time. You can't globally load them anymore in inherited templates.
- Frontend forms and JavaScript calling/submitting to controller actions will now require:
    - Action params must be kebab-case instead of camelCase.
    - Redirect params must be hashed now, also some of the redirect param tokens have changed.
- CSRF token protection is now enabled by default, so your forms and JavaScript calling/submitting to controller actions will need to include the token.

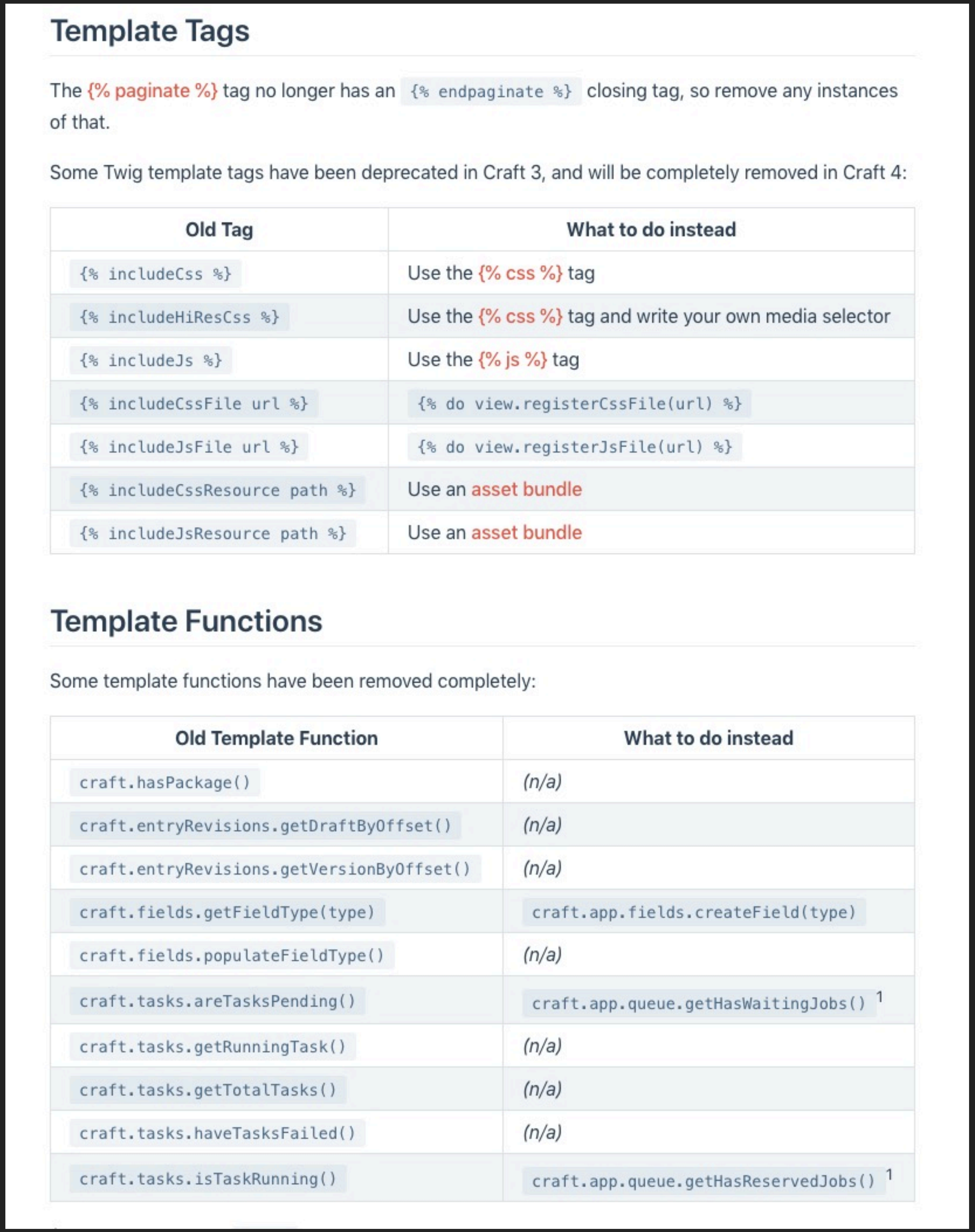
# TEMPLATE UPDATES (BY NO MEANS AN EXHAUSTIVE LIST)

- https://docs.craftcms.com/v3/changes-in-craft-3.html



## Changes in Craft 3

- Rich Text Fields
  - Redactor Configs
- Position Select Fields
- Remote Volumes
- Configuration
  - Config Settings
  - omitScriptNameInUrls and usePathInfo
- URL Rules
- PHP Constants
- Static Translation Files
- User Photos
- Twig 2
  - Macros
  - Undefined Blocks
- Template Tags
- Template Functions
- Date Formatting
- Currency Formatting
- Element Queries
  - Query Params
  - Query Methods
  - Treating Queries as Arrays
  - last()
- Elements
- Models
- Locales
- Request Params
  - action Params
  - redirect Params
  - CSRF Token Params
- Plugins



## Template Tags

The {% paginate %} tag no longer has an `{% endpaginate %}` closing tag, so remove any instances of that.

Some Twig template tags have been deprecated in Craft 3, and will be completely removed in Craft 4:

| Old Tag | What to do instead |
|---|---|
| `{% includeCss %}` | Use the {% css %} tag |
| `{% includeHiResCss %}` | Use the {% css %} tag and write your own media selector |
| `{% includeJs %}` | Use the {% js %} tag |
| `{% includeCssFile url %}` | `{% do view.registerCssFile(url) %}` |
| `{% includeJsFile url %}` | `{% do view.registerJsFile(url) %}` |
| `{% includeCssResource path %}` | Use an asset bundle |
| `{% includeJsResource path %}` | Use an asset bundle |

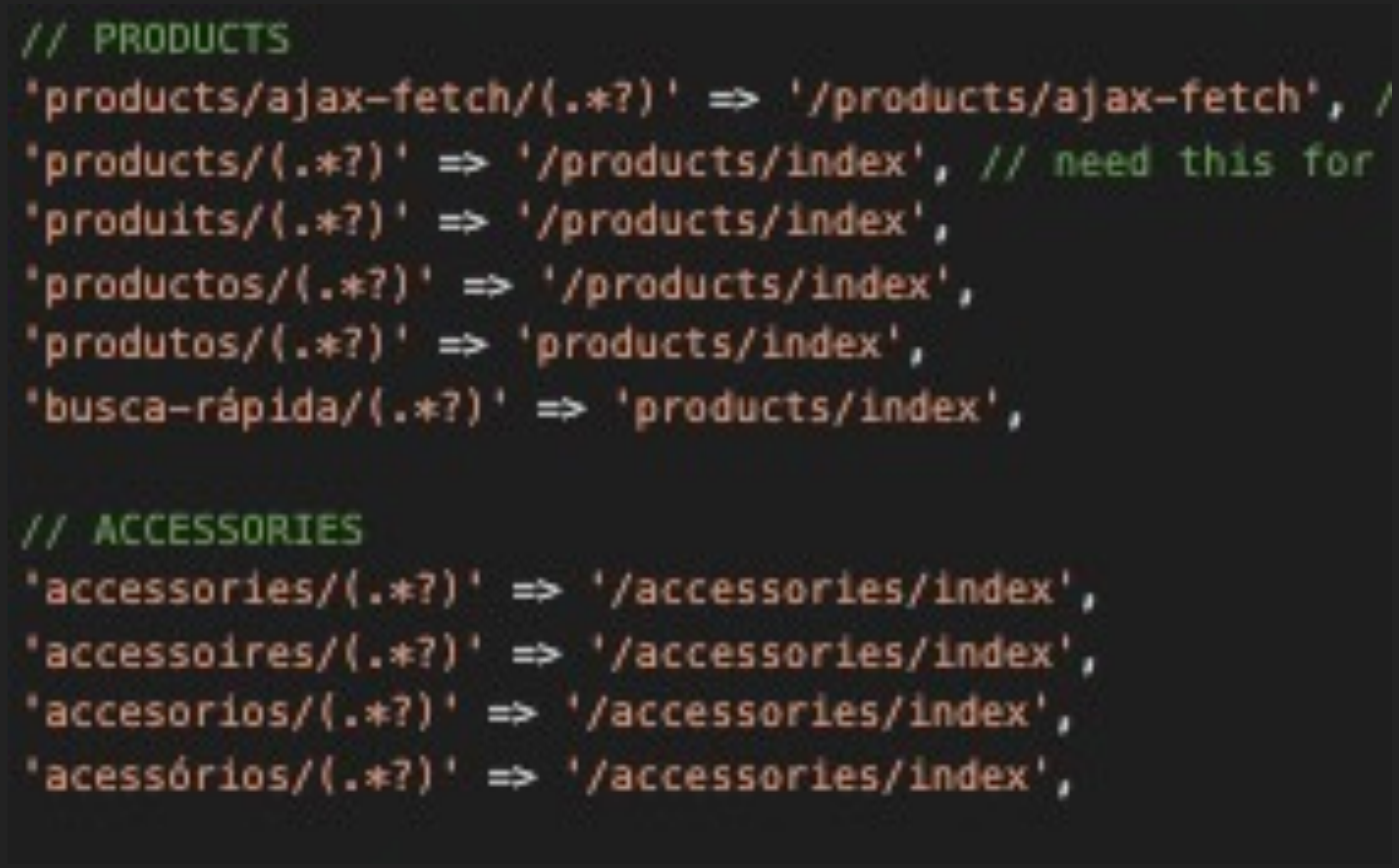
## Template Functions

Some template functions have been removed completely:

| Old Template Function | What to do instead |
|---|---|
| `craft.hasPackage()` | *(n/a)* |
| `craft.entryRevisions.getDraftByOffset()` | *(n/a)* |
| `craft.entryRevisions.getVersionByOffset()` | *(n/a)* |
| `craft.fields.getFieldType(type)` | `craft.app.fields.createField(type)` |
| `craft.fields.populateFieldType()` | *(n/a)* |
| `craft.tasks.areTasksPending()` | `craft.app.queue.getHasWaitingJobs()` [1] |
| `craft.tasks.getRunningTask()` | *(n/a)* |
| `craft.tasks.getTotalTasks()` | *(n/a)* |
| `craft.tasks.haveTasksFailed()` | *(n/a)* |
| `craft.tasks.isTaskRunning()` | `craft.app.queue.getHasReservedJobs()` [1] |

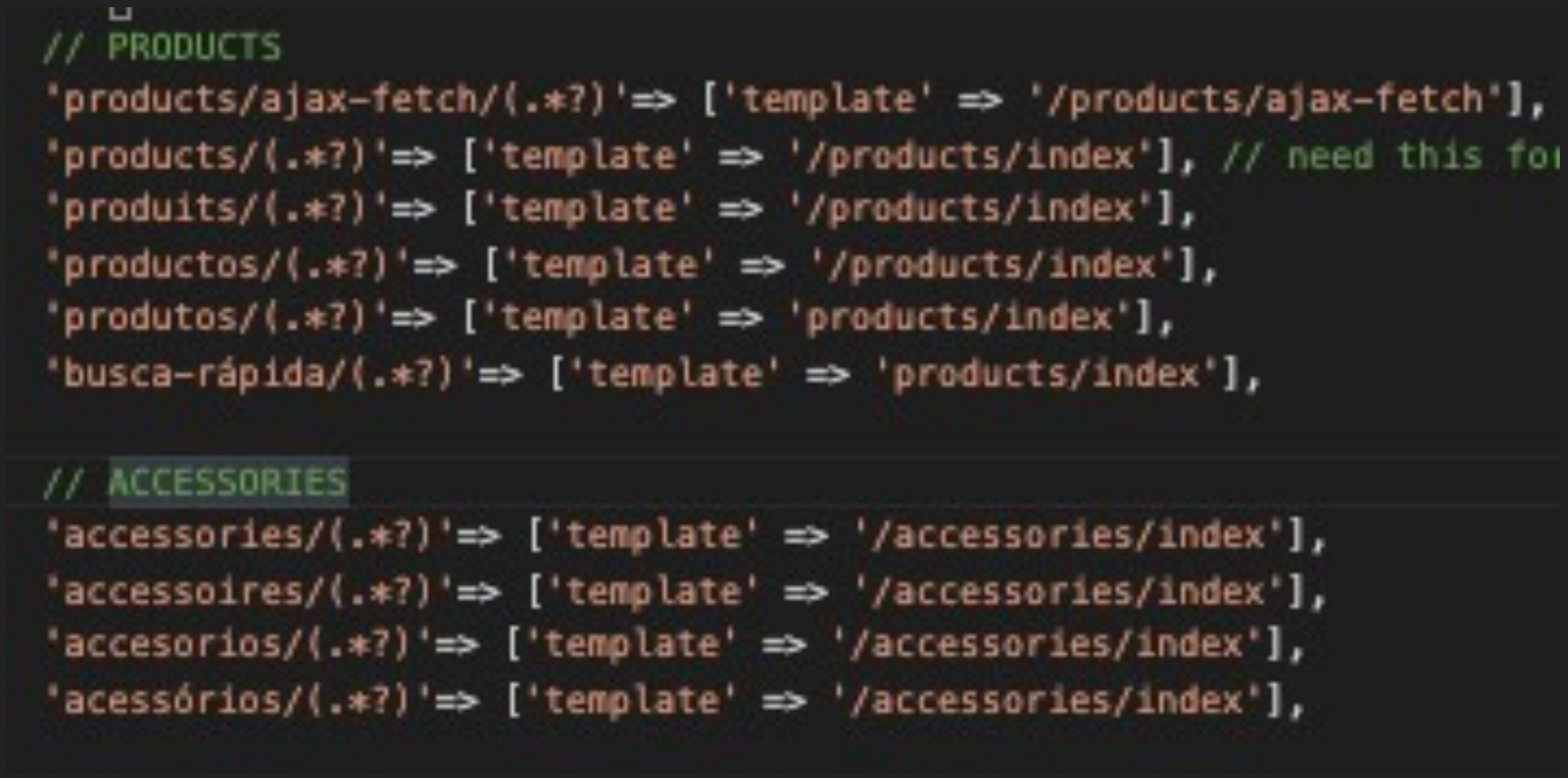
# ROUTES IN YII 2

- Routes now need to use the Yii 2 pattern-route syntax
- [https://docs.craftcms.com/v3/changes-in-craft-3.html#url-rules](https://docs.craftcms.com/v3/changes-in-craft-3.html#url-rules)



```
// PRODUCTS
'products/ajax-fetch/(.*?)' => '/products/ajax-fetch', /
'products/(.*?)' => '/products/index', // need this for
'produits/(.*?)' => '/products/index',
'productos/(.*?)' => '/products/index',
'produtos/(.*?)' => 'products/index',
'busca-rápida/(.*?)' => 'products/index',

// ACCESSORIES
'accessories/(.*?)' => '/accessories/index',
'accessoires/(.*?)' => '/accessories/index',
'accesorios/(.*?)' => '/accessories/index',
'acessórios/(.*?)' => '/accessories/index',
```

CRAFT 2 ROUTES



```
// PRODUCTS
'products/ajax-fetch/(.*?)'=> ['template' => '/products/ajax-fetch'],
'products/(.*?)'=> ['template' => '/products/index'], // need this fo
'produits/(.*?)'=> ['template' => '/products/index'],
'productos/(.*?)'=> ['template' => '/products/index'],
'produtos/(.*?)'=> ['template' => 'products/index'],
'busca-rápida/(.*?)'=> ['template' => 'products/index'],

// ACCESSORIES
'accessories/(.*?)'=> ['template' => '/accessories/index'],
'accessoires/(.*?)'=> ['template' => '/accessories/index'],
'accesorios/(.*?)'=> ['template' => '/accessories/index'],
'acessórios/(.*?)'=> ['template' => '/accessories/index'],
```

CRAFT 3 ROUTES

# CUSTOM PLUGINS

- You're going to have to rewrite them
- Consider offloading some less complex versions to Modules which can also be autoloaded
- Pluginfactory.io can help you scaffold your new plugin structure or module structure
- You have to adhere to PSR-4 coding standards
- Get up to speed with Composer
- Take some deep breaths. Beer might help also
- [https://docs.craftcms.com/v3/extend/updating-plugins.html#high-level-notes](https://docs.craftcms.com/v3/extend/updating-plugins.html#high-level-notes)

# CONVINCE YOUR CLIENT

- This might be the hardest part of the entire Craft 2 to Craft 3 migration. Some approaches that have been wins in my experience:
- Clients that value the security of their stack would be especially interested in end-of-life support for Craft 2 which is getting closer and closer each month
- Clients that have invested heavily in custom plugin functionality would be eager to ensure the long-term viability of that work. Making them realize that support and ongoing development resources/manpower will steadily decline as you approach and exceed end of life for Craft 2
- Clients that are especially tuned to data channel/social integrations and publishing should also be made aware of the same declining viability/support/manpower aspects. The next big social channel they want to target in a few months might not be available with a plugin in Craft 2
- Clients using third-party plugins are also at risk as support for those plugins will be dwindling as plugin developers move their efforts to supporting Craft 3 versions
- Special note for developers, before approaching this **ensure that the plugins have comparable Craft 3 versions or find an alternative**

# WRAP UP AND SOME ADDITIONAL RESOURCES

- Craft Migration guide: https://docs.craftcms.com/v3/upgrade.html
- Ryan's course on upgrading at Mijingo: https://www.youtube.com/watch?v=i029_DNFsmc
- Ben's video at Straight Up Craft: https://www.youtube.com/watch?v=T6wBZbH0oDU
- Additional articles from Ben: https://straightupcraft.com/articles/upgrading-from-craft-2-to-craft-3-should-i-update and https://straightupcraft.com/articles/upgrading-from-craft-2-to-craft-3-update-everything-on-craft-2
- Michael Westwood blog: https://medium.com/@mijewe/migrating-a-craft-2-site-to-craft-3-1e2a3d8ebae7
- Garret Winder blog: https://simplygoodwork.com/articles/updating-your-website-to-craft-cms-3