

API GUIDE: TEMPLATECLOUD

Introduction 3

- 1. **About this Document** 3
 - What is it? 3
 - Who should use and read this document? 3
 - Conventions and Style 3
- 2. **What is the TemplateCloud API?** 4
- 3. **Licensing and Legal Preparation** 5
- 4. **Understanding Your Credentials** 6
 - API Key 6
 - Editor Key 6
 - Asset Key 6
- 5. **Development and Production** 6
 - Your website and its data 6
 - Separating development from production 6
 - Your API Key 6

API Requests and Responses 8

- 6. **General Format** 8
 - Request format 8
 - Response format 8
- 7. **Template Discovery** 10
 - [Tag type](#) 10
 - [Tag group](#) 12
 - [Template size](#) 14
 - [Templates](#) 15
- 8. **Template Editing and File Assets** 21
 - templates/<template_id>/buyIndesignfile 21
 - templates/<template_id>/indesignfile 21
 - templates/<template_id>/edit 21
 - [Template instances](#) 21
 - instances/<instance_id>/makePDF 21
 - GET /instances/<instance_id>/pdf 22

Integration 23

- 9. **Mapping your Products** 23
- 10. **Choosing a Template** 24
 - Approaches for selecting a template 24
- 11. **Populating or Customizing a Template** 26
 - Embedding the TemplateCloud Editor 26
- 12. **Buying a PDF** 28
- 13. **PHP (etc) Platform Examples** 29
 - Interface 29
 - Examples 29

FAQ 30

14. Frequently-asked Questions30
Documentation 30
TemplateCloud Capabilities 30
API-consuming Services 31

Index 32

INTRODUCTION

1. About this Document

What is it?

This is a guide for developers wanting to integrate their services with the TemplateCloud API.

This document:

- Prepares you for using TCAPI (TemplateCloud API);
- Points you to online documentation for the API; and
- Provides examples of use, where they are not provided by the online documentation, at <https://developer.templatecloud.com/docs>

Who should use and read this document?

This is for you if you're looking for a way to integrate TemplateCloud's extensive library of templates, powerful customisation tools, and printing services* into your own website.



Getting your App approved

You'll need to obtain an API key before you can use the API.

Get your developer account and API key at:

- <https://developer.templatecloud.com/>

Your application will then be processed for approval.

Conventions and Style

Code or data snippets have been boxed and labelled with their type, e.g.

```
http://api.templatecloud.com/sandbox/tag_types/  
?user_key=ba386e729c5f6870cea89f38e92d7a5c&format=json
```

HTTP request

Code has been highlighted as follows:

- code
- keywords or reserved words
- literal strings
- variables (names of items having values)
- arguments (values of variables)
- comments

2. What is the TemplateCloud API?

TCAPI has three services, which let you:

- display available TemplateCloud templates on your own website;
- present an editor where your customers customize and buy their chosen designs; and
- when they have checked out from your website, you ask TemplateCloud for the PDF.

Here's a wireframe sketch of a typical API-using website:



Choosing a template

Here you may:

- Filter (by keyword, industry type, style, logo options, event, product types, size, and colour);
- Sort results;
- Display Template Listing; and ...
- Edit the Design.



Editing the chosen template

We built these sites using TCAPI:

- <http://www.printing.com/uk/catalogue/designs/leaflets>
- <http://www.drukland.be/Web2PrintProductsNew.aspx>
- <http://www.flyerzone.co.uk/business-cards>
- <http://www.templatecloud.com/uk/designs/>

See also:

- [FAQ chapter, p.30.](#)

3. Licensing and Legal Preparation

This document does not cover the licensing nor legal aspects of TemplateCloud; it documents the API interface itself. For information about business set up, please refer to the TemplateCloud Packages page: <http://www.templatecloud.com/us/entrepreneurs/packages/>

Licence

Apply for your API key at <https://developer.templatecloud.com/get-started>

Then when you have logged in, find your API key at https://developer.templatecloud.com/admin/access_details

4. Understanding Your Credentials

When using the API, you'll provide credentials in your requests.

There are three types of **key** that you use with the API:

API Key

Used for all API requests.

Can be found on https://developer.templatecloud.com/admin/access_details

This key is **the same** on development and live interfaces.

Editor Key

Used as 'salt' in your [asset key](#) when opening the Editor.

On the development interface, the [editor key](#) has the value `bf6b7bb900dcb665497362cbce0a9e90`.

This key is **different** on development and live interfaces; you'll be sent your own [editor key](#) when your production account is ready.

Asset Key

Used when opening the Editor

Calculated for each [instance](#), by making a hex string of the SHA1 hash of the concatenated string of (your [editor key](#)) + (the [instanceid](#) being edited). See "[extraParam: asset_key](#)", p.27.

The [asset key](#) is calculated in the same way on development and live interfaces, but the [editor key](#) will be different on development and live interfaces.

5. Development and Production

Your website and its data

You'll need to replicate some data structures on your website, which correspond with some of those in the TCAPI. At the very least, you'll be storing `id` fields from TemplateCloud, and temporarily holding lists of objects when processing the results of queries. See also:

- Chapter 7, "Template Discovery", p.10.
- Chapter 8, "Template Editing and File Assets", p.21.

Separating development from production

We have provided a developers' sandbox, to help you avoid incurring charges and writing to live data while you're testing.

Development URLs

- API server: <http://api.templatecloud.com/sandbox/> ...

We issue live URLs and credentials only when agreements have been signed.

Your API Key

Find your key at https://developer.templatecloud.com/admin/access_details

Guard your API Key

Your API Key allows your systems to access TemplateCloud. If someone else has your key, they too can access the same services, posing as your business.

It is important that you take reasonable measures to ensure that you do not publish your API Key:

- Use SSL for API calls (use `https://` instead of `http://`).
- If you experience problems due to unauthorised sharing of your API Key, then request a new API Key, close the security hole, and update your services to use your new API Key.
- Do not hard-code your API Key in every request call; instead use a variable containing the key.
 - Specify your API Key once, in a file on a private part of your server, and include it (by reference) in your framework or page code.
- Do not allow your web server to show code that might display the API Key.
 - Do not use debug code in the production environment, which might reveal the contents of variables that contain the API Key.
 - Handle errors correctly, so they do not reveal the code that caused the error. If you want traceability, then handle and privately log the error, and optionally inform the user there was a problem along with a reference number that can be used for support.
 - Try to use syntax checkers and syntax-highlighting code editors, and separate the processing code from the presentation code. This helps to avoid serving code into your HTML pages.
- Do not serve scripts or web pages that contain the API Key, i.e. do not design your systems so that your clients' web browsers are making TCAPI requests directly. Instead, your servers should take minimal requests from your client's pages, and then your servers issue the TCAPI requests to the TemplateCloud API on your clients' behalf.
- Do not inject unprocessed user-supplied strings into expressions, e.g. do not use something like

```
<?php $thatWorked = 'Insecure handy calculator: ' . eval( $userInput ); ?>
```

API REQUESTS AND RESPONSES

6. General Format

At <https://developer.templatecloud.com/docs>, we provide a sandbox for API requests, which documents the API requests, along with convenient entry forms where you can specify values for parameters.

Request format

Use standard HTTP requests to communicate with the API.

GET requests are used to read data from TemplateCloud using the API;

PUT requests try to write information to TemplateCloud using the API.

This example is a request to the API's sandbox handler, which returns a list of [tag types](#):

```
GET http://api.templatecloud.com/sandbox/tag_types/  
?user_key=ba386e729c5f6870cea89f38e92d7a5c&format=json
```

HTTP request

You have the option of specifying the response format using the `format` parameter. Allowed values are `json` or `xml`, the default being `xml`.

You'll need your own `user_key` (see chapter 3, "Licensing and Legal Preparation").

Response format

Error (response code 404)

```
{  
  "status": "error",  
  "message": "Template not found"  
}
```

HTTP response (JSON)

Success (response code 200)

Where one record is returned, it is presented as a lone object, e.g.

```
GET http://api.templatecloud.com/sandbox/tag_types/1  
?user_key=ba386e729c5f6870cea89f38e92d7a5c&format=json
```

HTTP request

...


```
{
  "tag_type": {
    "id": "1",
    "name": "Industry Types"
  }
}
```

HTTP response (JSON)

Where more than one record is returned, the top-level object is an array (having a plural name `tag_types`), containing child `tag type` objects.

e.g. here are the requests and responses for `tag types`:

```
GET http://api.templatecloud.com/sandbox/tag_types/
?user_key=ba386e729c5f6870cea89f38e92d7a5c&format=json
```

HTTP request

```
{
  "tag_types": [
    {
      "id": "1",
      "name": "Industry Types"
    },
    {
      "id": "2",
      "name": "Client"
    },
    etc...
  ]
}
```

HTTP response (JSON)

The XML version is below. Note that the above JSON does not encapsulate the array items as named objects, but the XML necessarily does:

```
GET http://api.templatecloud.com/sandbox/tag_types/
?user_key=ba386e729c5f6870cea89f38e92d7a5c&format=xml
```

HTTP request

```
<response status="ok">
  <tag_types>
    <tag_type>
      <id>1</id>
      <name>Industry Types</name>
    </tag_type>
    <tag_type>
      <id>2</id>
      <name>Client</name>
    </tag_type>
    etc...
  </tag_types>
</response>
```

HTTP response (XML)

7. Template Discovery

See also:

- [Choosing a Template](#), p.24.

Tag type

Parameter	Description
<code>user_key</code>	Your application's key, from https://developer.templatecloud.com/admin/access_details
<code>format</code>	<code>json</code> or <code>xml</code>

`/tag_types/<tagtype_id>/`

After `tag_types/`, specify the **id** of the **tag type** like this:

```
GET http://api.templatecloud.com/sandbox/tag_types/2/?user_key=ba386e729c5f6870cea89f38e92d7a5c&format=json
```

HTTP request

`/tag_types/`

Retrieves all **tag types**.

```
GET http://api.templatecloud.com/sandbox/tag_types/?user_key=ba386e729c5f6870cea89f38e92d7a5c&format=json
```

HTTP request

```
{
  "tag_types": [
    {
      "id": "1",
      "name": "Industry Types"
    },
    {
      "id": "2",
      "name": "Client"
    },
    <more objects of type 'tag type'> ...
  ]
}
```

HTTP response (JSON)

If **one** result is returned:

```
{
  "tag_type": {
    "id": "1",
    "name": "Industry Types"
  }
}
```

HTTP response (JSON)

Relations to `/tag_types/`

- **tag type** in **template**: `template.tag.tagtype_id`
- **tag type** in **tag group**: `taggroup.tagtype_id`

Relations from /tag_types/

- Use `tag_type.id` to get `tag groups` having that `tag type`:
`/tag_groups/...?type_id=<tagtype_id>`

Typical `tag types`

Tag group

`/tag_groups/<taggroup_id>/`

Retrieves a **tag group**.

Parameter	Description
<code>user_key</code>	Your application's key, from https://developer.templatecloud.com/admin/access_details
<code>format</code>	<code>json</code> or <code>xml</code>
<code>include</code>	List more data containers. Possible value: 'tag_type'

After `/tag_groups/`, specify the **id** of the **tag group** like this:

```
GET http://api.templatecloud.com/sandbox/tag_groups/2/?user_key=ba386e729c5f6870cea89f38e92d7a5c&format=json
```

HTTP request

```
{
  "tag_group": {
    "id": "2",
    "name": "Easter",
    "tagtype_id": "8"
  }
}
```

HTTP response (JSON)

`/tag_groups/`

Retrieves all **tag groups**, with optional filters.

Parameter	Description
<code>user_key</code>	Your application's key, from https://developer.templatecloud.com/admin/access_details
<code>format</code>	<code>json</code> or <code>xml</code>
<code>type_id</code>	One tag type ID (optional)
<code>include</code>	List more data containers. Possible value: 'tag_type'

```
GET http://api.templatecloud.com/sandbox/tag_groups/?user_key=ba386e729c5f6870cea89f38e92d7a5c&format=json&type_id=1
```

HTTP request

```
{
  "tag_groups": [
    {
      "id": "1",
      "name": "Other Events",
      "tagtype_id": "8"
    },
    {
      "id": "2",
      "name": "Easter",
      "tagtype_id": "8"
    },
    <more objects of type 'tag group' > ...
  ]
}
```

HTTP response (JSON)

Relations to /tag_groups/

- Use `tag_type.id` to find [tag groups](#) having that [tag type](#):
`/tag_groups/...?type_id=<tagtype_id>`
- Use `template.tag.taggroup_id` to get a [tag group](#) from a [template](#):
`/tag_groups/<taggroup_id>/`

Relations from /tag_groups/

- Use `tag_group.id` to find [templates](#) having [tags](#) belonging to that [tag group](#):
`/templates/?...filter_options=0&type_id=<tagtype_id>`
- Use `tag_group.id` to find [template filters](#) for [templates](#) having [tags](#) in that [tag group](#):
`/templates/?...filter_options=1&type_id=<tagtype_id>`
- Use `tag_group.tagtype_id` to get a single [tag type](#):
`/tag_types/<tagtype_id>/`

Self-relations in /tag_groups/

- Use `tag_group.id` to get a single [tag group](#):
`/tag_groups/<taggroup_id>/`
- Use `tag_group.tagtype_id` to find [tag groups](#) having that [tag type](#):
`/tag_groups/?type_id=<tagtype_id>`

Other relations using /tag_groups/

- Use `template.tag.taggroup_id` to find all [templates](#) having that [tag group](#):
`/templates/?group_id=<group_id>`

Template size

`/template_sizes/<templatesize_id>/`

Retrieves a [template size](#). Note that we do not publish these IDs in the API, so this is of limited use. Instead, just retrieve all [template sizes](#).

`/template_sizes/`

Returns all of the available [template sizes](#).

Parameter	Description
<code>user_key</code>	Your application's key, from https://developer.templatecloud.com/admin/access_details
<code>format</code>	<code>json</code> or <code>xml</code>
<code>templatesize_id</code>	One template size ID (optional; deprecated)
<code>units</code>	<code>imperial</code> or <code>metric</code> (optional)

GET http://api.templatecloud.com/sandbox/template_sizes/?user_key=ba386e729c5f6870cea89f38e92d7a5c&format=json

HTTP request

```
{
  "template_sizes" : [
    {
      "code" : "A4",
      "name" : "A4",
      "width" : "210.0000",
      "height" : "297.0000",
      "bleed" : "1.5000",
      "units" : "mm"
    }, {
      "code" : "A5",
      "name" : "A5",
      "width" : "148.0000",
      "height" : "210.0000",
      "bleed" : "1.5000",
      "units" : "mm"
    },
    <more objects of type 'template size'> ...
  ]
}
```

HTTP response (JSON)

Relations to `/template_sizes/`

- [Template size](#) from [template](#): `template.size.code`, when `include` contains `size`.
- Hidden: The unpublished `template_size.id` can be used as a [template size](#) argument in the API call, to retrieve a single [template size](#):
`/template_sizes/<templatesize_id>`

Templates

`/templates/<template_id>`

Returns the single [template](#).

```
GET http://api.templatecloud.com/sandbox/templates/12345/
?user_key=ba386e729c5f6870cea89f38e92d7a5c&format=json
```

HTTP request

`/templates/`

Returns the available [templates](#), with (many) optional filters.

- If more than one template is returned, then the response is a `templates` object containing many `template` items.
- If `filter_options=0`, the response contains **templates that match** the specified criteria. If `filter_options=1`, the response contains the **values of filter criteria**, i.e. valid tags or other parameter values (see below).
- Where filter criteria are not specified, those aspects are disregarded in the search.

Parameter	Description
<code>user_key</code>	Your application's key, from https://developer.templatecloud.com/admin/access_details
<code>format</code>	<code>json</code> or <code>xml</code>
Search criteria	
<code>template_id</code>	One template ID. Optional; omit to show all available templates.
<code>keywords</code>	Matches to <code>tag.name</code> . Optional.
<code>collection_id</code>	Matches to a collection . Optional. Equivalent to searching for a tag of tag type 7 ('collections').
<code>tag_groups</code>	tag group IDs (optional). Multiple items allowed, comma-separated.
<code>pages</code>	If required, a value to match <code>template.colour.pages</code> (the number of 'sides' or 'PDF pages' in a document). Typical values: <code>1</code> or <code>2</code> .
<code>sizes</code>	If specified, return only templates having matching <code>template.size.code</code> or <code>template.size.id</code> . Multiple items allowed, comma-separated.
<code>colour</code>	If specified, return only templates having colour tag of matching colour: <code>colour = template.tag.colour</code> (hexadecimal colour). Specify the colour as 3-byte hexadecimal, RGB ordering, e.g. <code>0568b4</code> .
<code>designer_id</code>	If specified, return only templates designed by this designer .
<code>exclusive_tag_types</code>	<code>0</code> or <code>1</code> . Default value <code>0</code> : will only list template results that have a match on ALL tag groups you've search for. Override value <code>1</code> : will list ANY template result that matches at least one of the tag groups you've searched for.

Paging and ordering of template results	
<code>limit</code>	For a paged response, specify this as the number of results to return per page. Optional. Typical value: 12.
<code>page</code>	When <code>limit</code> is specified, set the <code>page</code> index of results to return, starting with <code>page = 0</code> . Optional. Default value: 0.
<code>sort_key</code>	Allowed values: <code>rank</code> , <code>price</code> , or <code>date</code> . Default: <code>rank</code> .
<code>sort_order</code>	Allowed values: <code>desc</code> or <code>asc</code> . Defaults: <code>asc</code> for <code>sort_key = rank</code> ; <code>desc</code> for <code>sort_key = price date</code> .
Inclusion of metadata	
<code>include</code>	Additional element metadata items to include in each template object. Multiple items allowed, comma-separated. All options: <code>tags, size, colour, tag_group, images</code>
<code>filter_options</code>	0 or 1. See " <code>filter_options = 0</code> ", below.

include metadata

Include item	Description
<code>size</code>	<p>Details of the <code>template</code>'s dimensions.</p> <pre> "size": { "code": "BC", "name": "Business Card", "width": "85.0000", "height": "55.0000", "bleed": "1.5000", "units": "mm" }, </pre> <p>HTTP response (JSON)</p>
<code>colour</code>	<p>Contains the number of <code>pages</code> (typically 1 or 2), and the number of process inks on the front and reverse (0, 1, or 4).</p> <pre> "colour": { "coloursfront": 4, "coloursreverse": 4, "pages": 2 } </pre> <p>HTTP response (JSON)</p>

tags

Many **tags** may have been associated with each **template**.

```

"tags": [
  {
    "id": "214",
    "name": "Business Cards",
    "sales": "16491",
    "taggroup_id": "60",
    "tagtype_id": "3"
  },
  <another tag>
  ...
]
    
```

HTTP response (JSON)

tag_group

The **tag's** parent **tag group**, contained in the **tag** objects.

```

"tags": [
  {
    ...
    "tag_group": {
      "id": "60",
      "name": "Business Cards",
      "tagtype_id": "3"
    }
  },
  <another tag>
  ...
]
    
```

HTTP response (JSON)

images

URLs for thumbnail images, available in three image sizes per page. Here's the content for the 'small' image:

```

"images": [
  {
    "0": "http://www.templatecloud.com/api/...",
    "1": "http://www.templatecloud.com/api/...",
    "_attrs": {
      "name": "small"
      "size_code": "S"
    }
  },
  <another image>
  ...
]
    
```

HTTP response (JSON)

filter_options = 0

Returns the matching templates

```
GET http://api.templatecloud.com/sandbox/templates/
?user_key=ba386e729c5f6870cea89f38e92d7a5c&format=json
&include=tags%2Csize%2Ccolour%2Ctags%2Ctag_group%2Cimages&filter_options=0
&tag_groups=2
```

HTTP request

```
{
  "templates": {
    "0": {
      "id": "5924",
      "name": "Crunchy Cabbages Business Card ",
      "views": "102",
      "date_added": "2011-10-04 15:48:18",
      "designrank": "110",
      "allow_fotolia_library": true,
      "allow_object_deletion": true,
      "designed_by": "Norm Alizer",
      "designer_id": "693971",
      "price": "20.00",
      "cost_price": "19.00",
      "click_charge": "0.00",
      "price_currency": "USD",
      "tags": [
        <array of objects of type tag>
      ],
      "size": {
        "code": "BC",
        "name": "Business Card",
        "width": "85.0000",
        "height": "55.0000",
        "bleed": "1.5000",
        "units": "mm"
      },
      "images": [
        <array of objects of type image>
      ],
      "colour": {
        "coloursfront": 4,
        "coloursreverse": 4,
        "pages": 2
      }
    },
    <another template>
    ...
  }
}
```

HTTP response (JSON)

filter_options = 1

Returns all of the filters (parameters and their values) available for the given search criteria. Note that this may not be used in combination with a `template_id`.

For keywords that were converted to tags, `tags[].inferred=true`. The remaining keywords are shown in `keywords.refined[]`.

```
GET http://api.templatecloud.com/sandbox/templates/
?user_key=ba386e729c5f6870cea89f38e92d7a5c&format=json
&filter_options=1
```

HTTP request

```
{
  "colours" : [
    {
      "colour" : "86200e",
      "name" : "Maroon"
    },
    <more objects of type 'colour'> ...
  ],
  "sizes" : [
    {
      "name" : "US Business Card (2x3in)",
      "code" : "USABC",
      "count" : "572",
      "selected" : false,
      "area" : 7
    },
    <more objects of type 'size'> ...
  ],
  "tags" : [
    {
      "title" : "Product Types",
      "types" : "PRODUCT",
      "tags" : [
        {
          "name" : "Appointment Cards",
          "code" : "194",
          "count" : "42",
          "selected" : false,
          "inferred" : false
        },
        <more tags of type 'product'> ...
      ]
    },
    {
      "title" : "Industry Types",
      "types" : "INDUSTRY",
      "tags" : [
        {
          "name" : "Arts & Music",
          "code" : "15",
          "count" : "688",
          "selected" : false,
          "inferred" : true
        },
        <more tags of type 'industry'> ...
      ]
    },
    {
      "title" : "Styles",
      "types" : "STYLE",
      "tags" : [
        {
          "name" : "Abstract & Illustrative",
          "code" : "39",
          "count" : "2406",
          "selected" : false,
          "inferred" : false
        }
      ]
    }
  ]
}
```

HTTP response (JSON)

```
    },  
    <more tags of type 'style'> ...  
  ]  
},  
<more objects of type 'tag_type'> ...  
]  
"keywords" : {  
  "refined" : [  
    "test"  
  ],  
}  
}
```

Relations from /templates/

- Use `template.tag.tagtype_id` to get a [tag type](#):
`/tag_types/<tagtype_id>/`
- Use `template.tag.taggroup_id` to get a [tag group](#):
`/tag_groups/<taggroup_id>/`
- Use `template.tag.tag_group.tagtype_id` to get a [tag type](#):
`/tag_types/<tagtype_id>/`
- Use `template.size.name` to get a [template size](#):
`/template_sizes/<template_size.name>/`

Self-relations in /templates/

- Use `template.id` to get the single [template](#):
`/templates/<template_id>/`
- Use `template.tag.taggroup_id` to get all [templates](#) having that [tag group](#):
`/templates/?group_id=<group_id>`
- Use `template.designer_id` to get all [templates](#) by that [designer](#):
`/templates/?designer_id=<designer_id>`

8. Template Editing and File Assets

`/templates/<template_id>/buyIndesignfile`

Buys the template's indesign file.

`/templates/<template_id>/indesignfile`

Fetches the template's indesign file, and returns it as an attachment.

`/templates/<template_id>/edit`

Creates a [template instance](#) from a [template ID](#).

Parameter*	Description
<code>user_key</code>	Your application's key, from https://developer.templatecloud.com/admin/access_details
<code>format</code>	<code>json</code> or <code>xml</code>
<code>template_id</code>	One template ID (required)

GET `http://api.templatecloud.com/sandbox/templates/5277/edit?user_key=ba386e729c5f6870cea89f38e92d7a5c&format=json`

HTTP request

```
{
  "instance_id": "436935"
}
```

HTTP response (JSON)

Template instances

`/instances/<instance_id>/makePDF`

Initiates the creation of a PDF file, and charges your account.

Parameter*	Description
<code>user_key</code>	Your application's key, from https://developer.templatecloud.com/admin/access_details
<code>format</code>	<code>json</code> or <code>xml</code>
<code>queue</code>	<code>0</code> = generate and wait (synchronous) <code>1</code> = queue for later retrieval (asynchronous) – recommended!
<code>skip_buy</code>	Some API consumers have a special permission enabled, to bypass the fees, which can be exercised using <code>skip_buy=1</code> . This option is disabled for the majority of API keys.

PUT `http://api.templatecloud.com/sandbox/instances/50010/makePDF?user_key=ba386e729c5f6870cea89f38e92d7a5c&format=json`

HTTP request

```
{
  "status": "ok",
  "message": "PDF generation in progress",
  "job": "2152428"
}
```

HTTP response (JSON)

No further edits of a [template instance](#) are possible after a call to `makePDF`; further edits must be made to a new [template instance](#) which may be created using call to `edit` with a `template_id`.

GET /instances/<instance_id>/pdf

If the PDF is ready to download, the response to a `GET` request is a URL to the PDF file; otherwise it is an error notification.

It is advised that you poll `/pdf` with a `HEAD` request until the `200` response is returned; feed back the latest result to your web page via `AJAX`.

```
HEAD http://api.templatecloud.com/sandbox/instances/50010/pdf
?user_key=ba386e729c5f6870cea89f38e92d7a5c&format=json
```

HTTP request

```
{
  "status": "error",
  "message": "PDF generation in progress."
}
```

HTTP response 202 (JSON)

```
{
  "status": "error",
  "message": "PDF generation failed."
}
```

HTTP response 500 (JSON)

```
{
  "status": "error",
  "message": "PDF not created (probably instance not found, or some unknown
error)."
}
```

HTTP response 404 (JSON)

```
{
  "status": "ok",
  "message": "PDF ready."
}
```

HTTP response 200 (JSON)

After receiving a `200` response, the file can be retrieved using the equivalent `GET` request:

```
GET http://api.templatecloud.com/sandbox/instances/50010/pdf
?user_key=ba386e729c5f6870cea89f38e92d7a5c&format=json
```

HTTP request

If the file is still available, the `200` response will contain a binary stream. If the file is not available, the response will be an XML/JSON notification of the error (as above).

INTEGRATION

TCAPI is platform-neutral because all communications with TemplateCloud are handled using simple requests and JSON/XML responses. You can develop your website using any competent framework.

Typically, your business will need a website that:

- Presents and allows choice of a template, perhaps with a product in mind;
- Presents the editor to work on an instance of the template;
- Buys and downloads the customized PDF from TemplateCloud.

And of course

- Handles the relationship with your customer: securely processes payments, shows progress, generates correspondence;
- Controls manufacturing and logistics for your goods.

In this guide, we show how to choose a template and customize it using the editor.

9. Mapping your Products

TemplateCloud templates are designed for an item of known size. It is expected that you have a list of products, each product having a size that can be matched to the templates in TemplateCloud.

So that you can present appropriate templates to your customers, you should select only the `/template_sizes/` that have sizes corresponding to your products, i.e. matching `template_size.width`, `template_size.height`, and `template_size.bleed`.

Note that TemplateCloud contains templates in landscape and portrait orientations, so there might be additional matches when height and width are swapped, or compare `[min(h,w), max(h,w)]` of template size and your product.

One approach would be to create a table like this:

Column	Description
<code>template_size_id</code>	ID from <code>/template_sizes/</code> , also used in <code>template.size</code> .
<code>myProduct</code>	Your own product code

With example data like this:

template_size_id	myProduct
A4	StatPremA4
13rd	StatPremComp1
A4	DistroLeafA4

Query your table to retrieve all `template_size_id` values, and feed them into the `sizes` parameter of the `/templates/` API call.

10. Choosing a Template

Approaches for selecting a template

The purpose of your website is to select a template and buy a PDF for your customer. Often your customer will not know which template they want, so your site will need to present some templates, to eventually reduce the selection to the one that the customer wants.

There are many approaches to achieving the objective of reducing your customer's options to **one**.

- Start with **everything** (possibly matching a user-initiated text search), and present filters to help narrow down the options.
- Start with a **few filters enabled**, to restrict your users to a subset of the total TemplateCloud library. Do this if:
 - your site is 'special interest' or is dedicated to a business category, or you have a limited set of templates you want to sell (without the ability to remove those filters);
 - you want to inspire your users on a particular theme (where you make the filters available to switch back to the full collection).
- Start with a suggestion of **one template**. Feature a design, or randomly choose from a small pool of designs, showing only one of them at a time. Your user then has the one template needed to (customise and) progress the order.

Getting a list of templates

TemplateCloud contains thousands of templates, tagged to enable their discovery. Each criterion (size, colour, industry type, number of pages, etc) is a **tag type** having several possible **tags** within that type.

If presented with an untargeted list of thousands of templates, it's unlikely that your customer will find the template that they want, so you should ask TemplateCloud which **tag types** are available, and publish these **tags** as search options on your website, under **tag type** headings. It is conventional to add these as checkboxes alongside your search results.

You'll probably want a page of initial search results, especially if you used a text search as a starting point. It's also advisable to cache the results and present them to the client a page at a time.

Getting a list of tags

Call the `template` method, with `filter_options=1`, which returns the **tags** that are present in the **templates** that would be returned when `filter_options=0`. In other words, this gets a list of **tags**.

```
GET http://api.templatecloud.com/sandbox/templates/  
?user_key=ba386e729c5f6870cea89f38e92d7a5c&format=json  
&filter_options=1
```

HTTP request

The response looks like this:


```
{
  "colours" : [
    {
      "colour" : "86200e",
      "name" : "Maroon"
    },
    more colours ...
  ],
  "sizes" : [
    {
      "name" : "US Business Card (2x3in)",
      "code" : "USABC",
      "count" : "572",
      "selected" : false,
      "area" : 7
    },
    more sizes ...
  ],
  "tags" : [
    {
      "title" : "Product Types",
      "types" : "PRODUCT",
      "tags" : [
        {
          "name" : "Appointment Cards",
          "code" : "194",
          "count" : "42",
          "selected" : false
        },
        more tags of type 'product'...
      ]
    },
    more tag types...
  ]
}
```

HTTP response (JSON)

You can parse this response into an array, and present the options on your pages.

When your customer ticks the filter options, this should initiate a refined search using an API call having `filter_options=0` and the ticked `tags` encoded into parameters. Note that when a `tag` is specified, then it **excludes** all `templates` that **do not** have that `tag` within the `tag type`.

Including meta-information in your search results

To ask for extra information about the templates, you use the `&include` parameter to ask for a list of properties to be returned by, e.g. (`%2C` is a comma)

```
GET http://api.templatecloud.com/sandbox/templates/
?user_key=ba386e729c5f6870cea89f38e92d7a5c&format=json
&include=tags%2Csize%2Ccolour%2Ctags%2Ctag_group%2Cimages
```

HTTP request

11. Populating or Customizing a Template

This information is based on content from our developer website:

- <https://developer.templatecloud.com/howto/embed-tc-editor>

Embedding the TemplateCloud Editor

To embed the Flash editor on your site simple copy-paste the code below and fill all “{ }” blocks accordingly.

You can download the `jquery.flash.js` code below from the following site:

<http://jquery.lukelutman.com/plugins/flash/>

```
<!-- Start Editor -->
<script type="text/javascript"
src="/js/jquery.flash.js"></script>
<script type="text/javascript">
$(document)
  .ready(function () {
    $('#example')
      .flash({
        src:
'http://www.templatecloud.com/templates_frontend/PDCRobot.swf?1325859802',
        id: 'PDCRobot',
        width: 940,
        height: 800,
        quality: 'high',
        bgcolor: '#0000ff',
        allowscriptaccess: 'always',
        allownetworking: 'all',
        allowfullscreen: 'true',
        wmode: 'transparent',
        base: 'templates_frontend',
        flashvars: {
          autoParams: 'friendlynames=true',
          baseURL: 'http://www.templatecloud.com/api/',
          continueURL: '{your_continue_url}',
          cancelURL: '{your_cancel_url}',
          redirectURL: '{your_redirect_url}',
          templateid: '{template_id}',
          instanceid: '{instance_id}',
          customerid: '{your_customer_id}',
          locale: 'tc_GB',
          bleed: '1.5000',
          extraParam: '&asset_key=*****',
          activeButtonColor1: '0xc8d756',
          activeButtonColor2: '0xc8d756',
          fotoliaSearchTerm: 'business'
        }
      }, {
        version: 8
      });
  });
</script>
<div id="example"></div>
<!-- EOF Editor -->
```

HTML/JavaScript

flashvars{ Parameter }	Value, Description
Required	
autoParams	'friendlynames=true' True: we show friendly names like "Your Business Name" instead of our active fields {{business_name}}
baseUrl	The API server address. 'http://www.templatecloud.com/api/' for live 'http://dev.templatecloud.com/api/' for development (see "For development purposes..." below this table)
continueURL cancelURL redirectURL	Where the website sends users after they leave the editor: Continue will accept the edits and continue to refine or complete the order (perhaps to select a product). Cancel will be used if a user quits the editor. This will usually be a step backwards, e.g. to select another template, or review the existing template. Redirect is invoked when there is a fatal error (e.g. bad parameters).
templateid	The ID of the template that is being loaded into the editor.
instanceid	The ID of the template instance that is being loaded into the editor.
customerid	<u>Your</u> own customer ID, assigned to you by the TemplateCloud team. (<u>not</u> an ID for your customer)
locale	Selects a language for notifications and display. Allowed values: 'en', 'fr', 'nl', 'de', 'pt', 'it', 'es', 'se'
bleed	Templates for our production have 1.5 bleed; your SupplieReady products might have a different value.
extraParam: asset_key	= hex(SHA1(concat(<your api_key>, <instanceid>)))
activeButtonColor1	Primary button colour (for the most important actions) RGB hex (C format), e.g. '0xc8d756'
activeButtonColor2	Secondary button colour. RGB hex (C format), e.g. '0xa88080'
allowFotoliaImages	When false , the editor does not offer library images to replace existing images. (JS Boolean, default: true)
fotoliaSearchTerm	The term searched for in our Fotolia editor.
Image chooser - paging	
userImagesPagingRequired	'true' These help preserve client-side memory by loading user images in groups as required. E.g. If the first item is set to 'true', then the editor will load 10 images initially and load images in blocks of 5, as the user scrolls.
defaultNumberOfImages	10
numberOfImagesToAdd	5

For development purposes please use the following variables:

```
src = 'http://dev.templatecloud.com/templates_frontend/PDCRobot.swf?1325859802'
```

```
baseUrl = 'http://dev.templatecloud.com/api/'
```

```
templateid = 10014
```

Please use this ID for testing purposes; only a limited number are activated on the DEV server.

```
instanceid = 436738
```

This is generated using the method "Create Instance", see <https://developer.templatecloud.com/docs#/instances>

```
customerid = 'TCDEMO'
```

Select pricing and currency. Valid demo/development values are currently: 'TCDEMO', 'TCDEMOUS', 'TCDEMOFR', 'TCDEMODE', 'TCDEMOES', 'TCDEMONL', 'TCDEMOSE', 'TCDEMOAU', 'TCDEMONZ', 'TCDEMOIT', 'TCDEMOPT', 'TCDEMOIE'

```
asset_key =
```

Hexadecimal representation of SHA1 hash of `concat('bf6b7bb900dcb665497362cbce0a9e90', instanceid)`.

ExtraParam: asset_key

It's worth explaining how to obtain a value for `asset_key`, in more detail.

- First join the `instanceid` onto your `editor key`, e.g. if `instanceid=356` and the `editor key` is `ba386e729c5f6870cea89f38e92d7a5c`, then the concatenated string is `ba386e729c5f6870cea89f38e92d7a5c356`.
- then obtain the SHA1 hash of the result;
- and then convert that number to a hexadecimal string.

12. Buying a PDF

Protecting your cashflow

A PDF is bought when you call `makePDF`; a charge is made to your account.

You should endeavour to protect your cashflow by ensuring that whenever you are charged the small fee by TemplateCloud, you are charging your customer enough margin to cover the fee.

It might help to design your workflow so that it does not allow tight editing cycles.

Also remember to use the sandbox when developing and testing.

- See **Development and Production**, p.6.

Avoiding PDF exploitation

If further edits are needed after `makePDF`, you'll need to generate a new template instance from a template, which carries a further charge because that uses `makePDF`. As with the "protecting your cashflow" warning, you'll want to avoid creating lots of instances for the same chargeable item on your site. For example, you'll need to take payment before offering PDFs or derived works to your customer.

13. PHP (etc) Platform Examples

Interface

The Pest interface (<https://github.com/educoder/pest>) can simplify API calls. Here's an code snippet that gets and displays TemplateCloud data in PHP:

```
PHP
// Load the library (Pest folder must be in PHP's include path)
require_once 'Pest/PestJSON.php';
include '../data/tcapi_key.php'; // e.g. contains: $tcapi_key = '01234567890abcdef';

// Init the library (put in your TC credentials here; username not needed for dev/sandbox)
$apiClient = new PestJSON( 'https://api.templatecloud.com/v1' );
$apiClient->setupAuth( 'username', 'password' );

// Make a simple get call (this returns tag_types in JSON format)
try {
    $result = $apiClient->get( '/tag_types/?format=json&user_key=' . $tcapi_key );
}
catch (Exception $e) {
    // Handle error. In practice there are several exception types,
    // which you can use to differentiate between different error conditions
    die( $e->getMessage() );
}

// Examine successful result
// This just outputs to the page; you'll want to present it nicely.
echo '<pre>';
print_r( $result );
echo '</pre>';
```

To use the sandbox, the above example is modified slightly:

- Change the API access point to
`$apiClient = new PestJSON('https://api.templatecloud.com/sandbox');`
- You don't need to supply (username, password) at the authentication step.

When your production account is set up, we'll give you the (username, password) for the authentication step.

Examples

Templates

It's likely you'll want to get a list of templates that meet your criteria. Here's an example where we look for templates suitable for two-sided A6. It also returns their meta-data ([tags](#), [colours](#), etc) which you can omit by removing them from the `include` parameter.

```
PHP
...
$result = $apiClient->get(
    '/templates/?format=json' .
    '&user_key=' . $tcapi_key .
    '&include=tags%2Csize%2Ccolour%2Ctags%2Ctag_group%2Cimages' .
    '&pages=2&sizes=A6'
);
...

```

For other requests, see chapter 6, "General Format".

FAQ

14. Frequently-asked Questions

Here are lots of questions we've already been asked, along with their answers.

Documentation

1. **Is there a technical documentation where my engaged developer could refer to estimate the difficulty of the task and how it works?**

Besides this document, go to <https://developer.templatecloud.com/> and request an API key. The API is simple to integrate, and most of your time will usually be spent getting your database and website into the right 'shape'.

TemplateCloud Capabilities

2. **Listing Templates: Is it possible for me to pre-set and exclude some of the Filter? Can I hide the options?**

The beauty of the TemplateCloud API is that it is totally customisable. If you only wanted to sell Business Cards, or blue coloured designs, or leaflets for beauty spas, you only request those thumbnails. You can filter out any designs you don't want to display by price, product type, size, colour type or just about any attribute you can think of.

3. **Can a customer edit an old order?**

Yes, it is possible to create a new instance as a reorder of a previous design. The original stays 'intact', but the copy can be pre filled with same text/images so that they can start from a previous design. Each time you request a new PDF, we charge a design fee.

4. **I'm making a Business Card website where customers order for many employees...?**

Since you are controlling the shopping cart/checkout process, you can treat this as a single order from the client's perspective. For the API you would just create (say) 5 instances and join them to the same basket. Since each card consumes the same service/server load and we pay the same royalties to the designer, we would charge for 5 PDFs.

5. **Do the templates have Spot UV layer?**

We use the same template technology for 'private' client templates. We support spot UV, die-cut shapes, embossing and foil blocking. However, we [at the time of writing] don't have any designs in the TemplateCloud library which currently have any of those finishes. If you were

adding content to the TemplateCloud library, we can teach you how to set these up – it's simple.

API-consuming Services

6. Do I keep my existing checkout and banking process?

Yes. TemplateCloud API provides the three services that help your customer select a template, edit the design, and for you to request a PDF of the customised design. Your website charges your customers.

7. When do I incur transactional charges?

Only when you call `/instances/<instance_id>/makePDF`.

8. Where do I find template prices?

The `/templates/<template_id>` API call gives you the pricing of each template:

`template.price` = the price we advise you to sell to your customer (the "retail price");

`template.click_price` = will always be 0,00 for you and thus redundant;

`template.cost_price` = the price we will charge you for generating the PDF.

9. What is the URL for the live TemplateCloud API?

We do not publish the address of the live TemplateCloud API to prospective developers, until we have assessed the release candidates.

10. Template editing fails. How do I build the hashed 'instance ID'?

We describe how to build extraParams: `asset_key` in detail, at the end of chapter 11, "Populating or Customizing a Template" .

INDEX

/	
/instances/.../getPDF	22
/instances/.../makePDF	21
/tag_groups/	12
/tag_types/	10
/template_sizes/	14
/templates/	15

2	
200 (response)	8, 22
202 (response)	22

4	
404 (response)	8, 22

5	
500 (response)	22

A	
activeButtonColor1	27
activeButtonColor2	27
allow_fotolia_library	18
in template	18
allow_object_deletion	18
in template	18
API	
call	8
key	3
sandbox server	6
TemplateCloud	4
application	
approval	3
approval	3
area	
in size	19
array	9
asc	
sort_order	16
asset_key	
editor	28
attrs	
in image	17
autoParams	27

B	
banking	31
baseURL	27
editor	28
bleed	27
in template_size	14
buyIndesignfile	21

C	
call	8
cancelURL	27
case studies	5
charge	28, 31
checkout	31
click charge	31
click_charge	
in template	18
code	
in template_size	14
colour	
/templates/	15
in template	18
container	
tag group	12
continueURL	27
corporate accounts	30
cost	
template	18
cost price	31
cost_price	
in template	18
count	
in size	19
in tag	19
currency	
template	18
customer_id	
editor	28
customerid	27

D	
date	
/templates/ order	16
date_added	
in template	18
defaultNumberOfImages	27
desc	
sort_order	16
designed_by	

in template	18
designer	15
designer_id	
/templates/	15
in template	18
designrank	
in template	18
designs	24
dev server	6
development	6, 27
flashvars	28
die cut	30
documentation	30
online	3
drukland.be	5

E	
edit	
old instance	30
template	21
editor	4, 26
embossing	30
enumeration	
tagtype.id	11
error	8
example	4
examples	5
exclude (templates)	30
exclusive_tag_types	
/templates/	15
extraParam	27

F	
FAQ	30
filter	24
template	18
templates	30
filter_options	
/templates/	16, 18
template	15
finishing	
die cut	30
embossing	30
foil blocking	30
spot UV	30
flash	
container	26
flashvars	28
Flyerzone.co.uk	5
foil blocking	30
format	8
json	9
xml	9
fotoliaSearchTerm	27

G	
get request	8
getPDF	22

H	
height	
in template_size	14
HTTP	8
HTTPS	7

I	
id	
in tag	17
in tag_group	17
in tag_type	10
in template	18
tag group	12
image	
thumbnail	4
images	
in template	18
template	17
include	
/templates/	16
example	29
include (tag group)	12
include (templates)	30
include=size	14
InDesign file	21
indesignfile	21
indt	21
instance_id	
editor	28
instanceid	27
instances	
template	21
interface	8
items	9

J	
javascript	26
job	
in makePDF	22
JSON	9, 29

K		payment	31	/templates/	16	metadata	25
key		PDF		in template	18	price	31
security	7	generation	28	size (arg of include)	14	selection	24
keywords		PHP	29	size.name		template size	14
/templates/	15	popularity		to template	20	template.colour	16
L		template	18	sizes		template.images	17
limit		presentation	7	/templates/	15	template.size	16
/templates/	16	price		sort_key		template.size.code	15
live	27	/templates/ order	16	/templates/	16	template.tag_group	17
live API	31	in template	18	sort_order		template.tags	17
locale	27	template	31	/templates/	16	template_id	15
M		price_currency		spot UV	30	editor	28
makePDF	21, 28	in template	18	src		template_size	
message		product	23	editor	28	product	23
in getPDF	22	put request	8	SSL	7	template_size.id	14
in makePDF	22	R		status		TemplateCloud	4
metadata	25	rank		in getPDF	22	examples	5
/templates/	16	/templates/ order	16	in makePDF	22	templatecloud.com	5
meta-data		redirectURL	27	suggestion	24	templateid	27
tag group	12	relation		supply price		templates	15
N		tag group	13	template	18	display	4
name		tag type	10	T		example	29
in attrs.image	17	template size	14	tag group	12	templatesize_id	14
in tag	17	reorder	30	tag type	10	thumbnail	4
in tag_group	17	request	8	tag.name		title	
in tag_type	10	response		/templates/	15	in tag	19
in template	18	error	8	tag_group		transaction	31
in template_size	14	ok	8	from tag group	13	type	
tag group	12	retail price		from tag type	13	tag	10
numberOfImagesToAdd	27	template	18	from template	13	types	
O		RGB	15	template	17	in tags	19
ok	8	S		to tag type	13	U	
order		sales		to template	13	units	
templates	16	in tag	17	to template filters	13	in template_size	14
orientation	23	sandbox	6	tag_groups		user_key	8
P		script		/templates/	15	userImagesPagesRequire	
page		editor	26	tag_type		d	27
/templates/	16	search		metadata in		V	
pages		designer	15	tag_groups	12	views	
/templates/	15	paging/	16	tag_types	10	in template	18
paging	16	templates filter	15	taggroup_id	12	W	
editor images	27	search filter		in tag	17	website	4
parameters		template	18	to template	20	examples	5
editor	28	searching	10	tags		width	
parsing	29	security		in tags	19	in template_size	14
		API key	7	in template	18	wireframe	4
		HTTPS	7	template	17	X	
		SSL	7	tagtype		XML	9
		selected		from tag group	10		
		in size	19	from template	10		
		in tag	19	id	11		
		selection	24	to tag group	11		
		selling price		tagtype_id	10		
		template	18	in tag	17		
		server		in tag_group	12, 17		
		development	6	to template	20		
		size		template			
		edit	21	edit	21		
		filter options	15	instances	21		