

---

# **Django Smithy Documentation**

***Release 0.2.0***

**Jamie Counsell**

**Jul 22, 2019**



---

## Contents

---

<b>1</b>	<b>Django Smithy</b>	<b>3</b>
1.1	Documentation . . . . .	3
1.2	Quickstart . . . . .	3
1.3	Features . . . . .	4
1.4	Running Tests . . . . .	4
1.5	Credits . . . . .	5
<b>2</b>	<b>Installation</b>	<b>7</b>
<b>3</b>	<b>Usage</b>	<b>9</b>
<b>4</b>	<b>History</b>	<b>11</b>
4.1	0.1.0 (2019-03-16) . . . . .	11
4.2	0.2.0 (2019-03-07) . . . . .	11



Contents:



# CHAPTER 1

---

## Django Smithy

---

Craft HTTP requests in the Django Admin. Django Smithy is an HTTP Request templating system that allows developers to build systems to send abstract messages without additional development.

### 1.1 Documentation

The full documentation is at <https://django-smithy.readthedocs.io>.

### 1.2 Quickstart

Install Django Smithy:

```
pip install django-smithy
```

Then, create a request template to send:

Add request blueprint

Name:

Method:

Url:

Body: 

```
1 {
2   "name": "{{myobject.name}}"
3 }
```

#### HEADERS

NAME	VALUE	DELETE?
<input type="text" value="Authorization"/>	<input type="text" value="token {{token}}"/>	<input type="button" value="✕"/>
<a href="#">+ Add another Header</a>		

#### QUERY PARAMETERS

NAME	VALUE	DELETE?
<a href="#">+ Add another Query parameter</a>		

#### COOKIES

NAME	VALUE	DELETE?
------	-------	---------

## 1.3 Features

- Design requests in the Django Admin panel
- Send requests with whatever context you'd like
- Use Django's templating system everywhere
- Define variables to be added to the context

## 1.4 Running Tests

Does the code actually work?

```
source <YOURVIRTUALENV>/bin/activate
(myenv) $ pip install tox
(myenv) $ tox
```



## 1.5 Credits

Tools used in rendering this package:

- [Cookiecutter](#)
- [cookiecutter-djangopackage](#)



## CHAPTER 2

---

### Installation

---

At the command line:

```
$ easy_install django-smithy
```

Or, using pip:

```
$ pip install django-smithy
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv django-smithy  
$ pip install django-smithy
```



## CHAPTER 3

---

### Usage

---

Before sending requests, you must first create a blueprint. Request blueprints drive all the requests sent through Django Smithy.

Once a blueprint is created, you can call the `send` method and provide a dictionary of context. This context will be passed to all properties of the request and interpreted using Django's template language.

Django template syntax is supported in pretty much everything, including the name and value of headers, cookies, and query parameters, as well as the request body and URL. This makes it easy to make dynamic requests that contain contextual data.

Variables are just static values that allow you to easily reuse or swap out data without needing to provide it in code. This is great for storing things like API keys, but it's important to note the security impacts of this, as the API key will be visible in plain text to any user with access to this section of the Django Admin.

```
# Retrieve the blueprint
# (will change depending on your use case)
blueprint = RequestBlueprint.objects.first()

# Send the request and get the record
record = blueprint.send(context = {
    'something': 'some value',
    'more_stuff': 0
})

# Or get all records for that blueprint
RequestRecord.from_blueprint(blueprint)
```

Once sent, a request will generate a `RequestRecord` with details of the response. The `RequestRecord` can be used to determine if a request failed or not and handle appropriately.



#### 4.1 0.1.0 (2019-03-16)

- First release on PyPI. No tests are present, use at your own risk!

#### 4.2 0.2.0 (2019-03-07)

- Add support for x-www-form-urlencoded content
- Improved JS for admin form
- Auto-add content-type header
- Added better HTTP request/response records.