

# DocDocGoose - Legal Document Automation

## Introduction

Documentation for a legal document automation web application.

***Important to note, none of the sample templates provided are legally accurate in anyway, do NOT use them.***

## Overview

The web application has the following entities:

- **Templates** - These are text with "gaps" (variables) which can be filled by user input.
- **Documents** - These are text which are the result of filling in the variable "gaps" of a given template.

## Error Codes

Successful requests should return a 200 status code, failed requests should return a 400 status.

---

## Official Templates API

The official templates are a publicly available set of templates of legal documents which anyone with an account has access to use.

---

**GET** <http://127.0.0.1:8090/officialTemplates>

<http://127.0.0.1:8090/officialTemplates>

Returns a JSON object of all official templates.

## Example Request

`http://127.0.0.1:8090/officialTemplates`

```
var xhr = new XMLHttpRequest();
xhr.withCredentials = true;

xhr.addEventListener("readystatechange", function() {
  if(this.readyState === 4) {
    console.log(this.responseText);
  }
});
```

[View More](#)

## GET `http://127.0.0.1:8090/officialTemplates/:templateName`

`http://127.0.0.1:8090/officialTemplates/:templateName`

Return the JSON object for the official template of the given name.

### PATH VARIABLES

**templateName**

NDA

The name of the template being requested (other official templates are "Test" and "Will").

## Example Request

`http://127.0.0.1:8090/officialTemplates/:templateName`

```
var xhr = new XMLHttpRequest();
xhr.withCredentials = true;

xhr.addEventListener("readystatechange", function() {
  if(this.readyState === 4) {
    console.log(this.responseText);
  }
});
```

[View More](#)

## POST `http://127.0.0.1:8090/officialTemplates`

```
http://127.0.0.1:8090/officialTemplates
```

Create a new official template of the form as shown in the JSON object in the body.

BODY raw

```
{
  "name": "Postman",
  "type": "will",
  "class": "official",
  "dateModified": "",
  "vars": {
    "effectiveDate": "",
    "p1Name": "",
    "p1Address": "",
    "p2Name": "",
```

[View More](#)

Example Request

```
http://127.0.0.1:8090/officialTemplates
```

```
var data = "{
\n  \"name\": \"Postman\",
\n  \"type\": \"will\",
\n  \"class\": \"official\",
\n  \"dateModified\": \"\",
\n  \"vars\": {
\n    \"effectiveDate\": \"\",
\n    \"p1Name\": \"\",
\n    \"p1Address\": \"\",
\n    \"p2Name\": \"\",
```

[View More](#)

**DEL** `http://127.0.0.1:8090/officialTemplates/deleteTemplate/:templateName`

```
http://127.0.0.1:8090/officialTemplates/deleteTemplate/:templateName
```

Return the JSON object for the official template of the given name.

PATH VARIABLES

**templateName**

Postman

## Example Request

```
http://127.0.0.1:8090/officialTemplates/deleteTemplate/:templateName
```

```
var xhr = new XMLHttpRequest();
xhr.withCredentials = true;

xhr.addEventListener("readystatechange", function() {
  if(this.readyState === 4) {
    console.log(this.responseText);
  }
});
```

[View More](#)

# Documents API

An API to handle the requests for the documents of an individual user.

## POST

`http://127.0.0.1:8090/documents/createDocument/:documentName/:templateName`

```
http://127.0.0.1:8090/documents/createDocument/:documentName/:templateName
```

Create a new document of a given name, from the specified template.

### PATH VARIABLES

**documentName**

MyNDA

**templateName**

NDA

## Example Request

```
http://127.0.0.1:8090/documents/createDocument/:documentName/:templateName
```

```
var xhr = new XMLHttpRequest();
xhr.withCredentials = true;

xhr.addEventListener("readystatechange", function() {
  if(this.readyState === 4) {
    console.log(this.responseText);
  }
});
```

[View More](#)

## POST http://127.0.0.1:8090/documents/updateDocument/:documentName

http://127.0.0.1:8090/documents/updateDocument/:documentName

Update a specified document with the given JSON object.

### PATH VARIABLES

#### **documentName**

MyNDA

Name of the document to be updated.

### BODY raw

```
{
  "name": "Will",
  "type": "will",
  "class": "official",
  "dateModified": "",
  "vars": {
    "effectiveDate": "Today",
    "p1Name": "Nino",
    "p1Address": "1st Street",
    "p2Name": "X1z",
```

[View More](#)

### Example Request

http://127.0.0.1:8090/documents/updateDocument/:documentName

```
var data = "{
  \n  \"name\": \"Will\",
  \n  \"type\": \"will\",
  \n  \"class\": \"official\",
  \n  \"dateModified\": \"\",
  \n  \"vars\": {
    \n    \"effectiveDate\": \"Today\",
    \n    \"p1Name\": \"Nino\",
    \n    \"p1Address\": \"1st Street\",
    \n    \"p2Name\": \"X1z\",
```

[View More](#)

## GET http://127.0.0.1:8090/documents

http://127.0.0.1:8090/documents

Returns a JSON object of all documents.

### Example Request

http://127.0.0.1:8090/documents

```
var xhr = new XMLHttpRequest();
xhr.withCredentials = true;

xhr.addEventListener("readystatechange", function() {
  if(this.readyState === 4) {
    console.log(this.responseText);
  }
});
```

[View More](#)

## GET http://127.0.0.1:8090/documents/:documentName

http://127.0.0.1:8090/documents/:documentName

Returns a JSON object of the specified document.

### PATH VARIABLES

#### **documentName**

MyNDA

The name of the document which you want to return.

## Example Request

```
http://127.0.0.1:8090/documents/:documentName
```

```
var xhr = new XMLHttpRequest();
xhr.withCredentials = true;

xhr.addEventListener("readystatechange", function() {
  if(this.readyState === 4) {
    console.log(this.responseText);
  }
});
```

[View More](#)**DEL** `http://127.0.0.1:8090/documents/deleteDocument/:documentName`

```
http://127.0.0.1:8090/documents/deleteDocument/:documentName
```

Delete the specified document.

## PATH VARIABLES

**documentName**

MyNDA

Name of the document to be deleted.

## Example Request

```
http://127.0.0.1:8090/documents/deleteDocument/:documentName
```

```
var xhr = new XMLHttpRequest();
xhr.withCredentials = true;

xhr.addEventListener("readystatechange", function() {
  if(this.readyState === 4) {
    console.log(this.responseText);
  }
});
```

[View More](#)