Templater API

TEMPLATE API DOCS

# API

## POST: api/Token/GenerateToken

### Request Information

***Body Parameters***

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Type | Additional information |
| Username | Username to generate jwt | string | Required |

### Response Information

***Resource Description***

|  |  |  |  |
| --- | --- | --- | --- |
| *HttpResponseMessage* | | | |
| Name | Description | Type | Additional information |
| StatusCode | Success => 200 OK  Fail => 400 Bad Request | HttpStatusCode | None |
| Token | Jwt token | HttpContent | None |

### Example

Request:

POST: api/Token/GenerateToken

=> Support Content-Type application/x-www-form-urlencoded / application/json

***Request Format***

}

"Username": "sample UserName"

{

*application/json, text/json*

Response:

"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1bmlxdWVfbmFtZSI6InRlc3QiLCJuYm YiOjE2MTg4NDQyMDUsImV4cCI6MTYxODg0NTQwNSwiaWF0IjoxNjE4ODQ0MjA1f Q.rQVrSy1O7JHuuwcvrO5UL-0kN2TevQaIvaVhwfEF8go"

P a g e 1 | 9

Templater API

## POST: api/Token/ValidateToken

### Request Information

***Body Parameters***

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Type | Additional information |
| Token | Token Deserialize To User | String | Required |
| User | Username Compare With Token | String | Requried |

***Header Parameters***

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Type | Additional information |
| Authorization | Token jwt | “Authorization” : “Bearer Token” | Required Authorization |

### Response Information

***Resource Description***

|  |  |  |  |
| --- | --- | --- | --- |
| *HttpResponseMessage* | | | |
| Name | Description | Type | Additional information |
| StatusCode | Success => 200 / Fail => 401 | HttpStatusCode | None |

### Example

Request:

GET api/Token/ValidateToken

=> Support Content-Type application/x-www-form-urlencoded / application/json

***Request Format***

}

"Token": "sample Token", "Username": "sample UserName"

{

*application/json, text/json*

Response:

Success => 200 OK

Fail => 401 Unauthorized

P a g e 2 | 9

Templater API

## POST api/AccountAPI/Login

### Request Information

***Body Parameters***

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Type | Additional information |
| Username | Username Login | String | Required |
| Password | Password Login | String | Required |

***Request Formats***

}

"Username": "sample username", "Password": "sample password"

{

*application/json, text/json*

### Response Information

***Resource Description***

|  |  |  |  |
| --- | --- | --- | --- |
| *IHttpActionResult* | | | |
| Name | Description | Type | Additional information |
| StatusCode | Success => 200 / Fail => 400 | HttpStatusCode | None |

**POST: api/FillDocx/Fill Request Information *Body Parameters***

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Type | Additional information |
| NameTemplate | Name of template | String | Required |
| JsonPath | Path of json file (Absolute path) | String | Required |

***Header Parameters***

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Type | Additional information |
| Authorization | Token jwt | “Authorization” : “Bearer Token” | Required Authorization |

### Response Information

***Resource Description***

|  |  |  |  |
| --- | --- | --- | --- |
| *String* | | | |
| Name | Description | Type | Additional information |
| None | Success => String / Fail => Null | String | None |

P a g e 3 | 9

Templater API

### Example

Request:

POST api/FillDocx/Fill

=> Support Content-Type application/x-www-form-urlencoded / application/json

***Request Format***

}

"NameTemplate": "sample template", "JsonPath": "sample Jsonpath"

{

*application/json, text/json*

Response:

Success => 200 OK + File Name (string) Fail => 401 Unauthorized / null

## GET: api/FillDocx/IsExistFile

### Request Information

***Body Parameters***

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Type | Additional information |
| FileName | File Render | String | Required |

***Response Information***

|  |  |  |  |
| --- | --- | --- | --- |
| *HttpResponseMessage* | | | |
| Name | Description | Type | Additional information |
| StatusCode | Success => 200 / Fail => 400 | HttpStatusCode | None |

## POST: api/FillDocx/Download

### Request Information

***Body Parameters***

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Type | Additional information |
| FileName | File Render | String | Required |

***Header Parameters***

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Type | Additional information |
| Authorization | Token jwt | “Authorization” : “Bearer Token” | Required Authorization |

P a g e 4 | 9

Templater API

### Response Information

***Resource Description***

|  |  |  |  |
| --- | --- | --- | --- |
| *HttpResponseMessage* | | | |
| Name | Description | Type | Additional information |
| StatusCode | Success => 200 / Fail => 401 | HttpStatusCode | None |

# JSON FORMAT

## Rules

### Loop Section

* Define your *.json* file => Must be JArray to loop
* Define your *.docx* file => Create section area paragraph want to loop

## Render text

### JProperty

=> All key of property must be different

***Json***

}

“element1” : “value1”, “element2” : “value2”,

…….

{

*Example:*

***Docx***

*Markdown*

[[element1]] [[element2]]

## Render Table

### JArray

#### => Titile key of JArray property must be different

All content in array tag could matches => *render to list table*

P a g e 5 | 9

Templater API

***Json***

}

"Version": [

{

"stt": "1",

"update": "12.5.2021",

"author": "Le Ngoc Son"

},

{

"stt": "2",

"update": "15.5.2021",

"author": "Le Ngoc Son"

}

],

{

*Example:*

***Docx***

*Markdown*

[[Version.stt]] [[Version.update]]] [[Version.author]]

## Render + Loop

### JObject

=> Define section your docx file by second JArray

***Json***

"Vuln": [

{

"Name\_Vuln": "Name Vuln 1", "Name\_Function": [

{

"name\_func": "fun1", "add\_func": "[http://github.com](http://github.com/)"

},

{

"name\_func": "func 1.1", "add\_func": "[http://google.com](http://google.com/)"

{

*Example:*

P a g e 6 | 9

Templater API

}

] …..

},

{

"Name\_Vuln": "Name Vuln 2", "Name\_Function": [

{

"name\_func": "fun1", "add\_func": "[http://github.com](http://github.com/)"

},

{

"name\_func": "func 1.1", "add\_func": "[http://google.com](http://google.com/)"

}

] …….

}

}

***Docx***

*Maskdown:*

[[Vuln.Name\_Vuln]] => Render JArray

[[Name\_Function.name\_func]] => Second JArray + Define section your .docx file [[Name\_function.add\_func]] => Second JArray + Define section your .docx file

## Render image (Base64)

### Json

*Example:*

“image-{element}” : “value”

**Docx**

*Maskdown:*

[[image-{element}]:from-resource:maxSize(15)]

=> maxSize can be resize by change number into maxSize(size)

P a g e 7 | 9

Templater API

## Render Color (only apply for column in table)

**Json**

]

}

“Table”: [

{

…………..

“color-element1” : “critical”

},

{

…………..

“color-element2” : “high”

}

{

*Example*

**Docx**

*Maskdown:*

[[Table.color-{element}]][[Table.Color]:merge-xml]

=> [[{NameTable.Color}]:merge-xml] must be define after if you want fill color

|  |  |
| --- | --- |
| *Format Color* | |
| Risk | Color |
| Critical | Red |
| High | Orange Red |
| Medium | Orange |
| Low | Yellow |

## Render Chart (Property Name Must Be Chart)

### Json

*Example:*

"chart": {

"Critical": 3,

"High": 2,

"Medium": 2,

"Low": 1

}

P a g e 8 | 9

Templater API

### Docx

[[pie.value]]

[[pie.name]]

**Instances**

**Level**

*Maskdown:*



**CHART**

**0%**

[[pie.name]]

P a g e 9 | 9