

# Digital Forms

## Table of Contents

1. Access .....	4
2. Endpoints .....	4
2.1. Default .....	4
2.1.1. templatesGet .....	4
Description .....	4
Parameters .....	4
Return Type .....	4
Content Type .....	4
Responses .....	4
Samples .....	5
2.1.2. templatesTemplateIdGet .....	5
Description .....	5
Parameters .....	5
Return Type .....	5
Content Type .....	5
Responses .....	5
Samples .....	5
2.2. Favourites .....	5
2.2.1. favouritesGet .....	5
Description .....	5
Parameters .....	5
Return Type .....	6
Content Type .....	6
Responses .....	6
Samples .....	6
2.2.2. favouritesIdDelete .....	6
Description .....	6
Parameters .....	6
Return Type .....	6
Responses .....	6
Samples .....	6
2.2.3. favouritesIdGet .....	7
Description .....	7
Parameters .....	7
Return Type .....	7
Content Type .....	7

Responses .....	7
Samples .....	7
2.2.4. favouritesPost .....	7
Description .....	7
Parameters .....	7
Return Type .....	8
Content Type .....	8
Responses .....	8
Samples .....	8
2.3. Forms .....	8
2.3.1. formsFormIDAttachmentsGet .....	8
Description .....	8
Parameters .....	8
Return Type .....	8
Content Type .....	8
Responses .....	9
Samples .....	9
2.3.2. formsFormIDDelete .....	9
Description .....	9
Parameters .....	9
Return Type .....	9
Responses .....	9
Samples .....	9
2.3.3. formsFormIDDownloadGet .....	9
Description .....	10
Parameters .....	10
Return Type .....	10
Content Type .....	10
Responses .....	10
Samples .....	10
2.3.4. formsFormIDGet .....	10
Description .....	10
Parameters .....	10
Return Type .....	11
Content Type .....	11
Responses .....	11
Samples .....	11
2.3.5. formsFormIDPut .....	11
Description .....	11
Parameters .....	11
Return Type .....	11

Content Type .....	11
Responses .....	12
Samples .....	12
2.3.6. formsGet .....	12
Description .....	12
Parameters .....	12
Return Type .....	12
Content Type .....	12
Responses .....	12
Samples .....	12
2.3.7. formsPost .....	12
Description .....	12
Parameters .....	13
Return Type .....	13
Content Type .....	13
Responses .....	13
Samples .....	13
2.4. Groups .....	13
2.4.1. groupsGet .....	13
Description .....	13
Parameters .....	13
Return Type .....	13
Content Type .....	13
Responses .....	14
Samples .....	14
2.5. UserData .....	14
2.5.1. userDataGet .....	14
Description .....	14
Parameters .....	14
Return Type .....	14
Content Type .....	14
Responses .....	14
Samples .....	14
2.5.2. userDataPost .....	14
Description .....	14
Parameters .....	15
Return Type .....	15
Content Type .....	15
Responses .....	15
Samples .....	15
3. Models .....	15

3.1. <i>Attachment</i> .....	15
3.2. <i>Favourite</i> .....	16
3.3. <i>Form</i> .....	16
3.4. <i>FormElement</i> .....	16
3.5. <i>FormSection</i> .....	17
3.6. <i>TemplateGroup</i> .....	17
3.7. <i>UserData</i> .....	17

## Abstract

### *API for the Digital Forms Application*

# 1. Access

- **Bearer** Authentication

# 2. Endpoints

## 2.1. Default

### 2.1.1. templatesGet

GET /templates

Get all Templats

#### Description

Returns all templates

#### Parameters

#### Return Type

array[[Form](#)]

#### Content Type

- application/json

#### Responses

Table 1. HTTP Response Codes

Code	Message	Datatype
200	successful operation	List[ <a href="#">Form</a> ]

## Samples

### 2.1.2. templatesTemplateIdGet

GET /templates/{templateId}

Find Template by ID

#### Description

Returns a single Template

#### Parameters

##### Path Parameters

Name	Description	Required	Default	Pattern
templateId	ID of Template to return	X	null	

#### Return Type

[Form](#)

#### Content Type

- application/json

#### Responses

Table 2. HTTP Response Codes

Code	Message	Datatype
200	successful operation	<a href="#">Form</a>
404	Template not found	<<>>

## Samples

## 2.2. Favourites

### 2.2.1. favouritesGet

GET /favourites

Get all Favourites

#### Description

Returns all favourites

#### Parameters

## Return Type

array[[Favourite](#)]

## Content Type

- application/json

## Responses

Table 3. HTTP Response Codes

Code	Message	Datatype
200	successful operation	List[ <a href="#">Favourite</a> ]

## Samples

### 2.2.2. favouritesIdDelete

DELETE /favourites/{id}

delete a Favourite by ID

## Description

delete a Favourite by ID for the requesting User

## Parameters

### Path Parameters

Name	Description	Required	Default	Pattern
id	ID of favourite to delete	X	null	

## Return Type

-

## Responses

Table 4. HTTP Response Codes

Code	Message	Datatype
200	successful operation	<<>>
404	Can't find ID	<<>>

## Samples

### 2.2.3. favouritesIdGet

GET /favourites/{id}

Find Favourite by Template ID

#### Description

Returns a single Favourite

#### Parameters

##### Path Parameters

Name	Description	Required	Default	Pattern
id	ID of a form template, that might be a Favorite	X	null	

#### Return Type

*Favourite*

#### Content Type

- application/json

#### Responses

Table 5. HTTP Response Codes

Code	Message	Datatype
200	successful operation	<i>Favourite</i>
404	Template not found	<<>>

#### Samples

### 2.2.4. favouritesPost

POST /favourites

Save a new Favourite

#### Description

Save a new Favourite for the requesting User

#### Parameters

##### Body Parameter

Name	Description	Required	Default	Pattern
Favourite	Save a new Favourite for the requesting User <i>Favourite</i>	-		

## Return Type

*Favourite*

## Content Type

- application/json

## Responses

Table 6. HTTP Response Codes

Code	Message	Datatype
201	successful Creation with ID	<i>Favourite</i>
400	Invalid Data	<<>>

## Samples

# 2.3. Forms

## 2.3.1. formsFormIDAttchmentsGet

GET /forms/{formID}/attchments

Get All Requird attchments

## Description

Returns All Requird attchments for the Form

## Parameters

### Path Parameters

Name	Description	Required	Default	Pattern
formID	ID of form	X	null	

## Return Type

array[*Attachment*]

## Content Type

- application/json



## Responses

Table 7. HTTP Response Codes

Code	Message	Datatype
200	successful operation	List[ <a href="#">Attachment</a> ]
404	Form not found	<<>>

## Samples

### 2.3.2. formsFormIDDelete

DELETE /forms/{formID}

delete a Form by ID

## Description

delete a Form by ID for the requesting User

## Parameters

### Path Parameters

Name	Description	Required	Default	Pattern
formID	ID of form to return	X	null	

## Return Type

-

## Responses

Table 8. HTTP Response Codes

Code	Message	Datatype
200	successful operation	<<>>
404	Can't find ID	<<>>

## Samples

### 2.3.3. formsFormIDDownloadGet

GET /forms/{formID}/download

Download as a PDF file

## Description

Returns a PDF file of the Form.

## Parameters

### Path Parameters

Name	Description	Required	Default	Pattern
formID	ID of form to return	X	null	

## Return Type

[\[File\]](#)

## Content Type

- application/pdf

## Responses

Table 9. HTTP Response Codes

Code	Message	Datatype
200	A PDF file.	<a href="#">[File]</a>
400	Bad Request.	<<>>
500	Exception During PDF Generation	<<>>
501	The Provided Form does not Support the PDF Export	<<>>

## Samples

### 2.3.4. formsFormIDGet

**GET** /forms/{formID}

Get the formdata by ID

## Description

Get the Data for a Form by ID for the requesting User

## Parameters

### Path Parameters

Name	Description	Required	Default	Pattern
formID	ID of form to return	X	null	

## Return Type

*Form*

## Content Type

- application/json

## Responses

Table 10. HTTP Response Codes

Code	Message	Datatype
200	successful operation	<i>Form</i>
404	Form not found	<<>>

## Samples

### 2.3.5. formsFormIDPut

PUT /forms/{formID}

Update a Form by ID

## Description

Update a Form by ID for the requesting User

## Parameters

### Path Parameters

Name	Description	Required	Default	Pattern
formID	ID of form to return	X	null	

### Body Parameter

Name	Description	Required	Default	Pattern
Form	Update a Form by ID for the requesting User <i>Form</i>	-		

## Return Type

*Form*

## Content Type

- application/json

## Responses

Table 11. HTTP Response Codes

Code	Message	Datatype
200	successful operation	<a href="#">Form</a>
400	Invalid Data	<<>>

## Samples

### 2.3.6. formsGet

GET /forms

Get all Forms

#### Description

Returns all Forms for the requesting User

#### Parameters

#### Return Type

array[[Form](#)]

#### Content Type

- application/json

## Responses

Table 12. HTTP Response Codes

Code	Message	Datatype
200	successful operation	List[ <a href="#">Form</a> ]

## Samples

### 2.3.7. formsPost

POST /forms

Save a new Form

#### Description

Save a new Form for the requesting User

## Parameters

### Body Parameter

Name	Description	Required	Default	Pattern
Form	Save a new Form for the requesting User <a href="#">Form</a>	-		

### Return Type

[Form](#)

### Content Type

- application/json

### Responses

Table 13. HTTP Response Codes

Code	Message	Datatype
201	successful Creation with ID	<a href="#">Form</a>
400	Invalid Data	<<>>

### Samples

## 2.4. Groups

### 2.4.1. groupsGet

GET /groups

Get Template groups

#### Description

Returns all Template Groups

#### Parameters

#### Return Type

array[[TemplateGroup](#)]

#### Content Type

- application/json

## Responses

Table 14. HTTP Response Codes

Code	Message	Datatype
200	successful operation	List[ <a href="#">TemplateGroup</a> ]

## Samples

# 2.5. UserData

## 2.5.1. userDataGet

GET /userData

Get Autofill userdata of a user

## Description

Returns the Autofill userdata of the user

## Parameters

## Return Type

[UserData](#)

## Content Type

- application/json

## Responses

Table 15. HTTP Response Codes

Code	Message	Datatype
200	successful operation	<a href="#">UserData</a>

## Samples

## 2.5.2. userDataPost

POST /userData

Save new Autofill Userdata

## Description

Save new Autofill Userdata

## Parameters

### Body Parameter

Name	Description	Required	Default	Pattern
UserData	Save new Userdata <i>UserData</i>	-		

### Return Type

*UserData*

### Content Type

- application/json

### Responses

Table 16. HTTP Response Codes

Code	Message	Datatype
201	successful Creation with ID	<i>UserData</i>
400	Invalid Data	<<>>

### Samples

## 3. Models

### 3.1. Attachment

Field Name	Required	Type	Description	Format
id		String		
description		String	describeds the Attachment	
required		String	When is it Required / When should it be shown	<i>Enum:</i> always, user, conditional,
conditionRef		String	Only needed if required==conditional. Refrences an FormElement id	
conditionRefVal		String	Used for a value check with conditionRef	
help		String	Additional help text regarding the Attachment	

## 3.2. Favourite

Field Name	Required	Type	Description	Format
id		String		
owner		String	User this Favourite belongs to	
formId		String	ID of Template	

## 3.3. Form

Field Name	Required	Type	Description	Format
id		String	the id	
owner		String	User this Form belongs to	
parent		String	id of the parent Template	
template		Boolean	Is this a Template?	
titel		String	titel of the Form	
description		String	description of the Form	
category		String		Enum: Antrag, Abrechnung,
form		List of <i>FormSection</i>	Content of the Form	
attachments		List of <i>Attachment</i>	All Attachments	

## 3.4. FormElement

Field Name	Required	Type	Description	Format
id		String	unique ID within Form	
Description		String	Description of expected content	
help		String	Help text	



Field Name	Required	Type	Description	Format
type		String	Type of this input	<i>Enum:</i> text, address, iban, date, money, TextMultiLine, bool,
value		String	saved Value	

### 3.5. *FormSection*

Field Name	Required	Type	Description	Format
order		Integer	Position of this Section	int32
section		String	Name of this Section	
items		List of <i>FormElement</i>	List of Elements in this Section	

### 3.6. *TemplateGroup*

Field Name	Required	Type	Description	Format
id		String		
titel		String	Titel of a Antrag Abrechnung Group	
description		String	Description of a Antrag Abrechnung Group	
antragId		String	Form id of the Antrag	
rechnungen		List of <i>[string]</i>	List of one or more Form id of the Abrechnung	
reasons		List of <i>[string]</i>	Only used when more then one rechnungen exists. Used to help the user decided what variant shall be used	

### 3.7. *UserData*

Field Name	Required	Type	Description	Format
id		String		
userId		String		
email		String		
name		String		
firstName		String		
adress		String		
IBAN		String		
CreditInstitute		String		