

# Features in BzCtrl.

These features can be used across multiple Pages and Workspaces to simplify daily operations and improve efficiency.

Features such as **ApiForm** and **SnapForm** also enable integration with external users and systems, allowing data to be exchanged and processes to connect seamlessly across platforms.

Additional information and feature updates can also be found on the official website: [BzCtrl](#).

- [External System Integration - ApiForms](#)
- [External System Integration - Incidents & Tasks](#)
- [ApiForm instruction - VTTI specific](#)

# External System Integration

## - ApiForms

BzCtrl. provides the ability to call predefined Forms through API communication using the **ApiForms** feature. This allows external systems to create and link Form instances directly from within their own operational processes.

The main value of ApiForms is that a user can launch a specific Form directly from the source system, while the created Form instance remains permanently accessible through a stored URL within that source system.

## How ApiForms Work

Each ApiForm provides a unique API endpoint that allows an external system to create a new Form instance in BzCtrl.

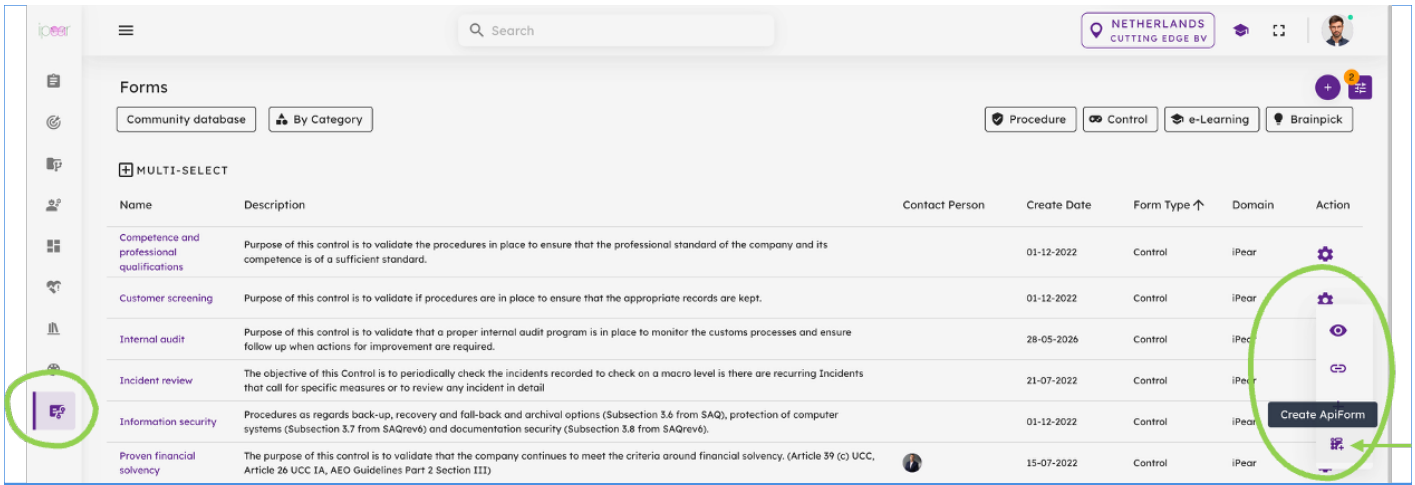
The request includes:

- Predefined Form reference (via API Endpoint)
- Tags and identifiers (for example an Order Number, Shipment Number, or Case ID).

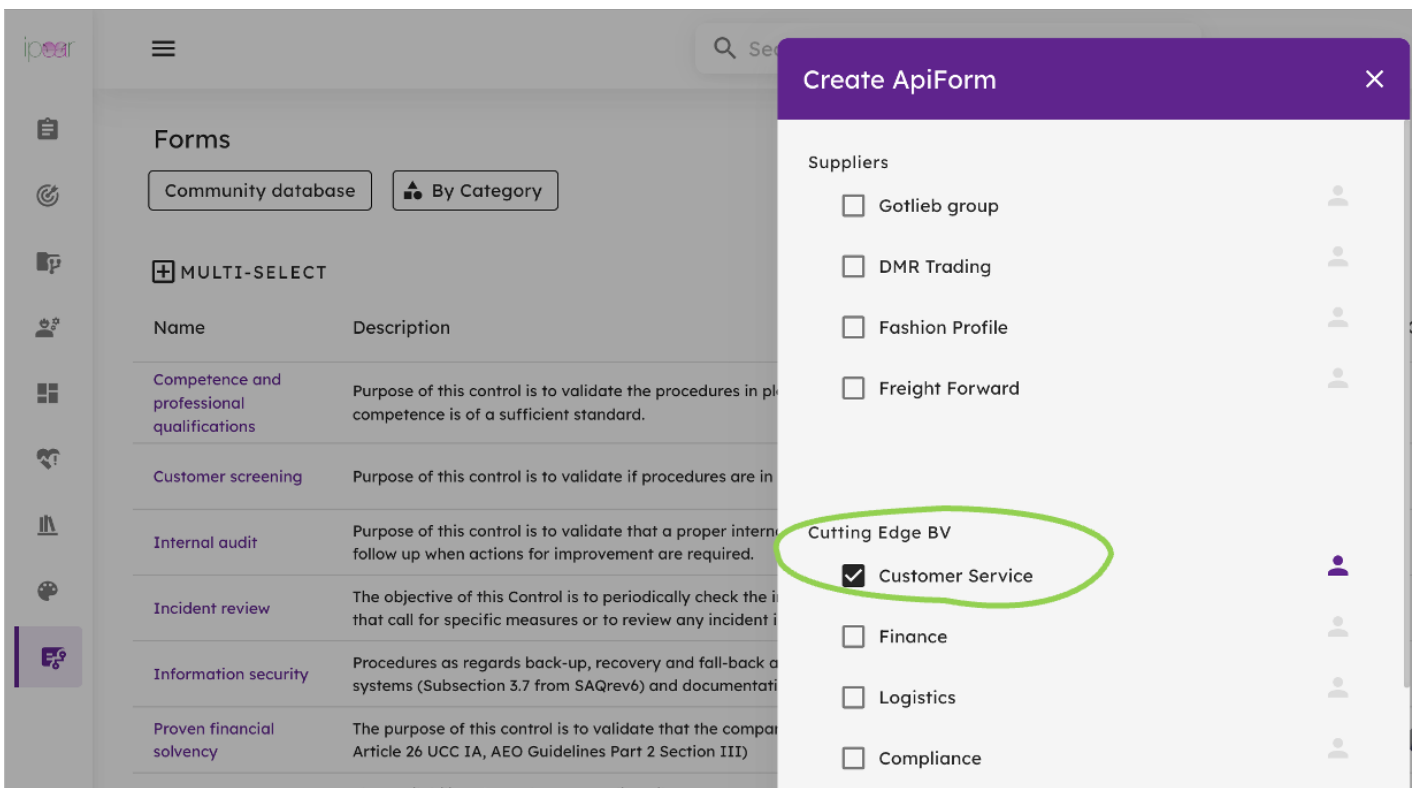
Once the Form instance is created, BzCtrl. returns a response containing the direct URL to the Form instance. This URL can then be stored in the source system, allowing users to always reopen the corresponding Form directly from their operational process.

## User actions

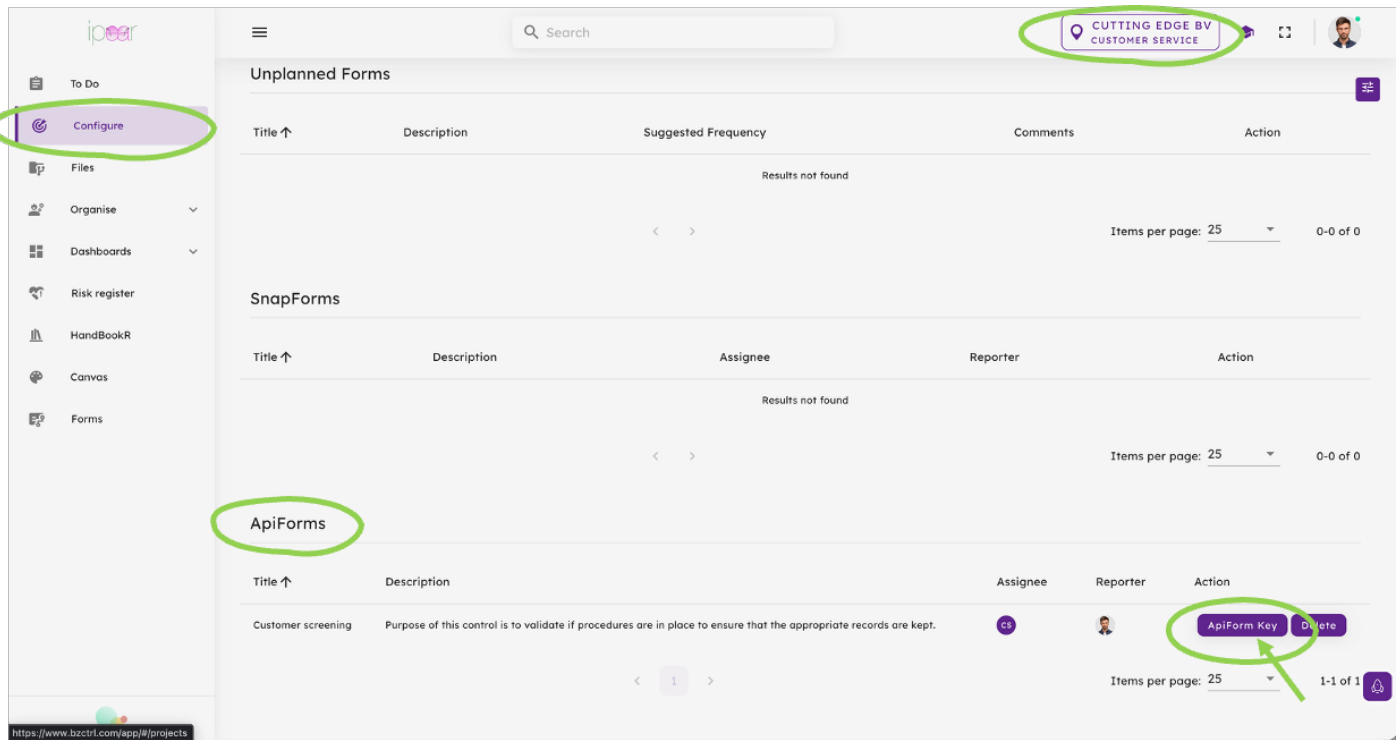
**Step 1** - From the Forms Page the User can create an ApiForm via the Action Button for a specific Form;



**Step 2** - Select the Workspace where the Form Instances should be created and stored;



**Step 3** - From the Configure page the Form is now available as ApiForm. Please make sure the correct Workspace is selected (right top). From the ApiForms Table select the ApiForm Key Button to get the details to share with your developers.



# Input to Developers External system

## Step 1 - Authentication

Before an ApiForm can be called, the external application must authenticate with BzCtrl. Authentication is performed by sending a **POST** request to the appropriate authentication endpoint using credentials provided for API communication.

### Endpoints

#### Acceptance Environment (ACC)

<https://acc.bzctrl.com/bzctrl-core-api/oauth/token>

#### Production Environment (PROD)

<https://bzctrl.com/bzctrl-core-api/oauth/token>

### Request Method

POST

### Request Body (x-www-form-urlencoded)

```
grant_type=password  
username=example@example.com  
password=Ex@mpl3!
```

## Sample Response

```
{
  "access_token": "xxxxxxxxxx",
  "refresh_token": "xxxxxxxxxx",
  "token_type": "Bearer",
  "expires_in": 3600,
  "scope": "*",
  "jti": "xxxxxxxxxx"
}
```

## Result


Upon successful authentication, BzCtrl. returns an **access token** that must be included in subsequent API requests. The token serves as proof of authentication and authorizes communication with the BzCtrl. API. The token remains valid for the period specified in the [expires\\_in](#) field.

## Step 2 - Api Form Key


Here you see the API Endpoint url to share with your developers, which use will result in the creation of a Form Instance of the respective Form on your selected Workspace. In the Body of the message Tags can be included, which is usually used for adding Order ID's to be recorded in the Form Instance.


 API Endpoint

```
https://www.bzctrl.com/bzctrl-core-api/api/v1/task/add-exposed-task/eyJwcm9qZWNO0SWQI0jQ2NCwiY29tcGFueUlkiJoxMDEsImNvbnRyb2xJZCI6MTQ1NCwiZW5jb2RlVXJsIjoIbmRkODVhODgtZGY5Ny00NDkzLWE2OWMtZjJmMmJjYjAzZDY1IiwiaWY3JiYXRvcklkIjo0MjEsImFzc2lnbmVlVXNlckdyb3VwSWQI0jEzMWid29ya1NwYWNIImFtZSI6Ikd1c3RvbWVvIFNlcnZpY2UiLCJyZXBvcnRlclVzZXJJZCI6NDIxLClJyZXBvcnRlclVzZXJOYW1IiwoiSmFtZXMGmQmlzaG9wIn0=
```

 Copy URL

POST request to update tags for a specific task

 Request Body Example

 You can update the tags value with your own tags as needed.

## JSON Request

 Copy

```
{
  "tags": [
    "urgent",
    "bug-fix",
    "frontend"
  ]
}
```

 Sample Response

## JSON Response

 Copy

```
{
  "title": "Fix navigation bug",
  "taskType": "CONTROL",
  "description": "Resolve issue with mobile navigation menu",
  "taskKey": "FED-98",
  "status": "TODO",
  "priority": null,
  "dueDate": null,
  "reminderDate": null,
  "completionDate": null,
  "taskLink": "https://example.com/#/todo/fed-98",
  "priorityLevel": 0,
  "assigneeUserGroupId": null,
  "creatorUserId": 1,
  "deleted": false,
  "isPrivate": false,
  "showResult": false,
  "tags": [
    "urgent",
    "bug-fix",
    "frontend"
  ],
  "assignee": "John Doe",
  "reporter": "Jane Smith",
  "taskAttributes": [],
  "watchers": [],
  "todoList": [],
  "categoryList": [],
  "subCategoryList": [],
  "company": "TechCorp Inc.",
  "project": "Website Redesign"
}
```

 Copy

After creation the Response message includes a Tasklink that can be stored to allow the user access to the Form Instance from the External system at any stage.

# External System Integration

## - Incidents & Tasks

BzCtrl. provides a feature that allows external systems to create **Incidents** and **Tasks** directly through REST API communication. This enables operational systems to register issues or actions directly within their own workflows while ensuring follow-up and tracking take place in BzCtrl.

The main value of this feature is that it allows Incidents and Tasks to be created directly from specific screens or processes within an external operational system, without requiring users to manually enter data into BzCtrl.

## Integration Flow

The integration process consists of the following steps:

The request contains the information required to create the Incident (or Task) in BzCtrl.

### Step 1 - Authentication

Before an ApiForm can be called, the external application must authenticate with BzCtrl. Authentication is performed by sending a **POST** request to the appropriate authentication endpoint using credentials provided for API communication.

#### Endpoints

##### Acceptance Environment (ACC)

<https://acc.bzctrl.com/bzctrl-core-api/oauth/token>

##### Production Environment (PROD)

<https://bzctrl.com/bzctrl-core-api/oauth/token>

#### Request Method

POST

#### Request Body (x-www-form-urlencoded)

```
grant_type=password
username=example@example.com
password=Ex@mpl3!
```

## Sample Response

```
{
  "access_token": "xxxxxxxxxx",
  "refresh_token": "xxxxxxxxxx",
  "token_type": "Bearer",
  "expires_in": 3600,
  "scope": "*",
  "jti": "xxxxxxxxxx"
}
```

## Result

Upon successful authentication, BzCtrl. returns an **access token** that must be included in subsequent API requests. The token serves as proof of authentication and authorizes communication with the BzCtrl. API. The token remains valid for the period specified in the expires\_in field.

## Step 2 - Create Incident (or Task)

After receiving the access token from the OAuth API, use the Add Task API to create a new Task or Incident.

### EndPoint

ACC: <https://acc.bzctrl.com/bzctrl-core-api/api/v1/task/add-task>

PROD: <https://www.bzctrl.com/bzctrl-core-api/api/v1/task/add-task>

**Note: The access token received from the OAuth Token API must be included in the Authorization header using the Bearer scheme.**

**Request Method:** [POST](#)

## Required Headers

```
Content-Type: application/json
Authorization: Bearer {access_token}
```

## Sample Request Body

```
{
  "todoType": "INCIDENT",
  "domainWorkSpace": "EXA-CON-BUS-WOR",
  "title": "Potential Customs claim",
  "description": "Operation started without proper approvals",
  "financialImpact": "123577",
  "impact": "HIGH",
  "likelihood": "MEDIUM",
  "status": "IN PROGRESS",
  "priority": "HIGH",
  "assignee": "Example@example.com",
  "reporter": "Example@example.com",
  "category": [ "Customs" ],
  "subCategory": [ "IT" ],
  "lrn": "725462",
  "orderID": "ORDER 777",
  "reminderDate": "2024-09-27"
}
```

## Sample Response

```
{
  "title": "Potential Customs claim",
  "taskType": "INCIDENT",
  "description": "Operation started without proper approvals",
  "taskLink": "https://www.bzctrl.com/app/#/todo/EXA-CON-BUS-WOR-03",
  "tags": [ "ORDER 777" ]
}
```

**This URL can be stored in the external system for direct access and traceability. Or you can open this URL in the browser by the JavaScript function(`window.open("URL")`).**

From a UI/UX perspective the following should be considered (\* Mandatory, *Italic* - User input, CAPITOL - System generated Values)

Todo Type	Fields	Input	Comments
incident, task	todoType*	INCIDENT	<b>Customer</b> - option to change to Task (or in case only used for Incidents default value)

incident, task	domain*	EXA-CON-BUS-WOR	Default value to include by <b>Customer</b> , relates to the Domain
incident, task	workspace*	WOR	Default value to include by <b>Customer</b> , relates to the Workspace under the Domain
incident, task	title*	<i>Free text field</i>	<b>Customer</b> , allow user to provide Title/Subject
incident, task	description*	<i>Free text field</i>	<b>Customer</b> , allow user to provide Description
task	result	<i>Free text field</i>	<b>Customer</b> , allow user to provide Results (optional)
incident	financialImpact	<i>Numbers field</i>	<b>Customer</b> - For User it should be clear this is a EUR amount. Only Number is provided. BzCtrl. - This should be included in the Field "Potential financial impact of the incident" set as EUR.
incident	impact	<i>High, Medium, Low</i>	<b>Customer</b> , allow user to select out of these three options (optional)
incident	likelihood	<i>High, Medium, Low</i>	<b>Customer</b> , allow user to select out of these three options (optional)
incident, task	status*	<i>In Progress</i>	Options BzCtrl. are To do, In Progress, Done and Expired. <b>Customer</b> - default when sending on In Progress, with option to change by User to Done. BzCtrl. - If status - Done, set due date to date transaction.
incident, task	priority	<i>High, Medium, Low</i>	<b>Customer</b> , allow user to select out of these three options (optional)
incident, task	assignee*	<u>User@example.com</u>	<b>Customer</b> , Same as Reporter. E-mail address of the User.
incident, task	reporter*	<u>User@example.com</u>	<b>Customer</b> , E-mail address of the User.
incident, task	category		n/a

incident, task	orderID*	ServiceOrder Number	<b>Customer</b> - BzCrtrl. - Set as Tag
incident, task	LRN	LRN	<b>Customer</b> - BzCrtrl. - Set as Tag (if available handy reference)
incident, task	reminderDate	Calendar selection	n/a

# ApiForm instruction - VTTI specific

BzCtrl. provides the ability to call predefined Forms through API communication using the **ApiForms** feature. This allows external systems to create and link Form instances directly from within their own operational processes.

The main value of ApiForms is that a user can launch a specific Form directly from the source system, while the created Form instance remains permanently accessible through a stored URL within that source system.

## How ApiForms Work

Each ApiForm provides a unique API endpoint that allows an external system to create a new Form instance in BzCtrl.

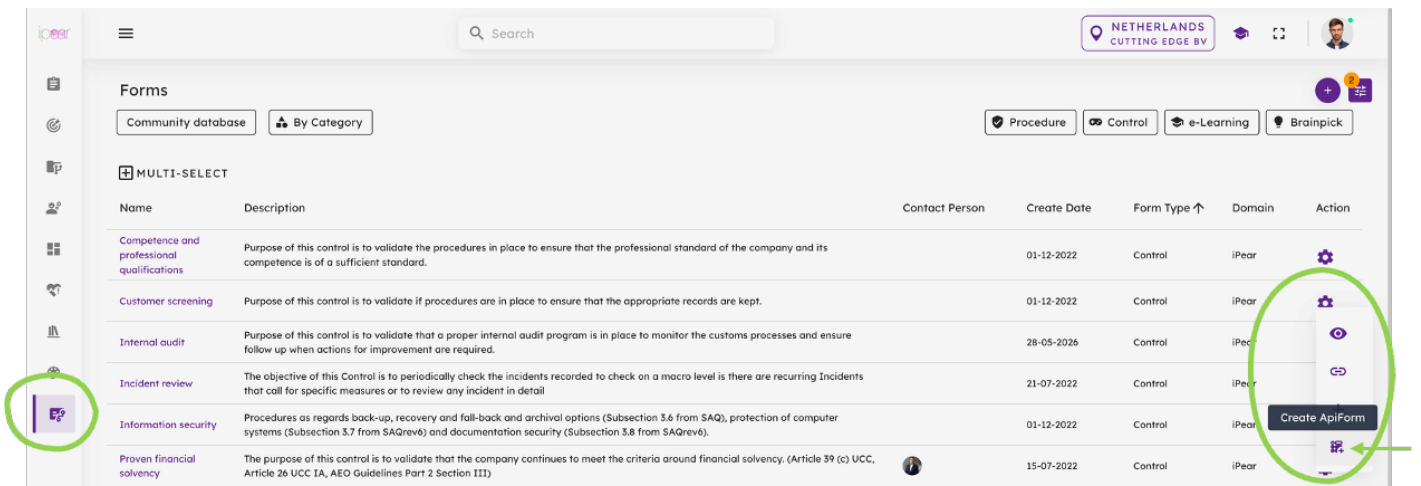
The request includes:

- Predefined Form reference (via API Endpoint)
- Tags and identifiers (for example an Order Number, Shipment Number, or Case ID).

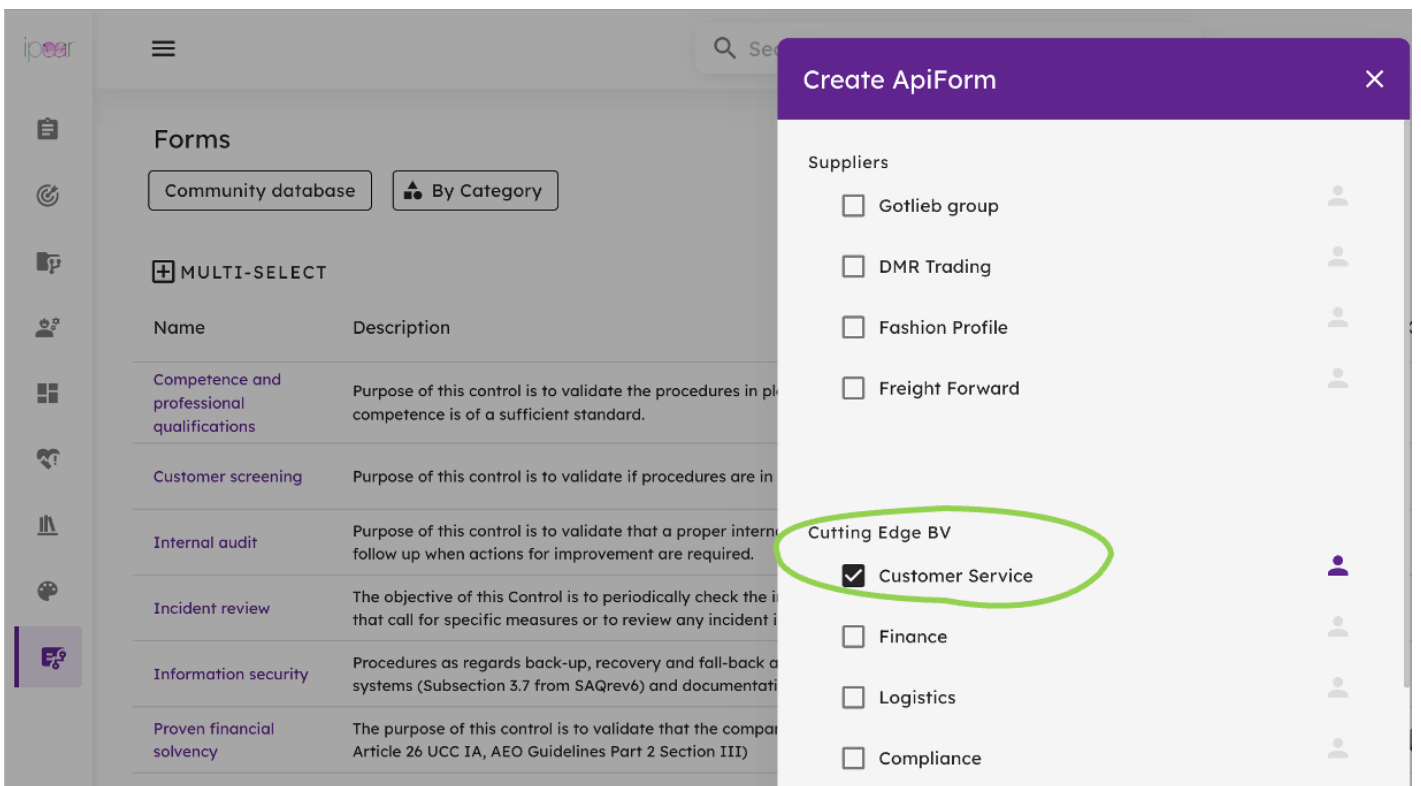
Once the Form instance is created, BzCtrl. returns a response containing the direct URL to the Form instance. This URL can then be stored in the source system, allowing users to always reopen the corresponding Form directly from their operational process.

## User actions

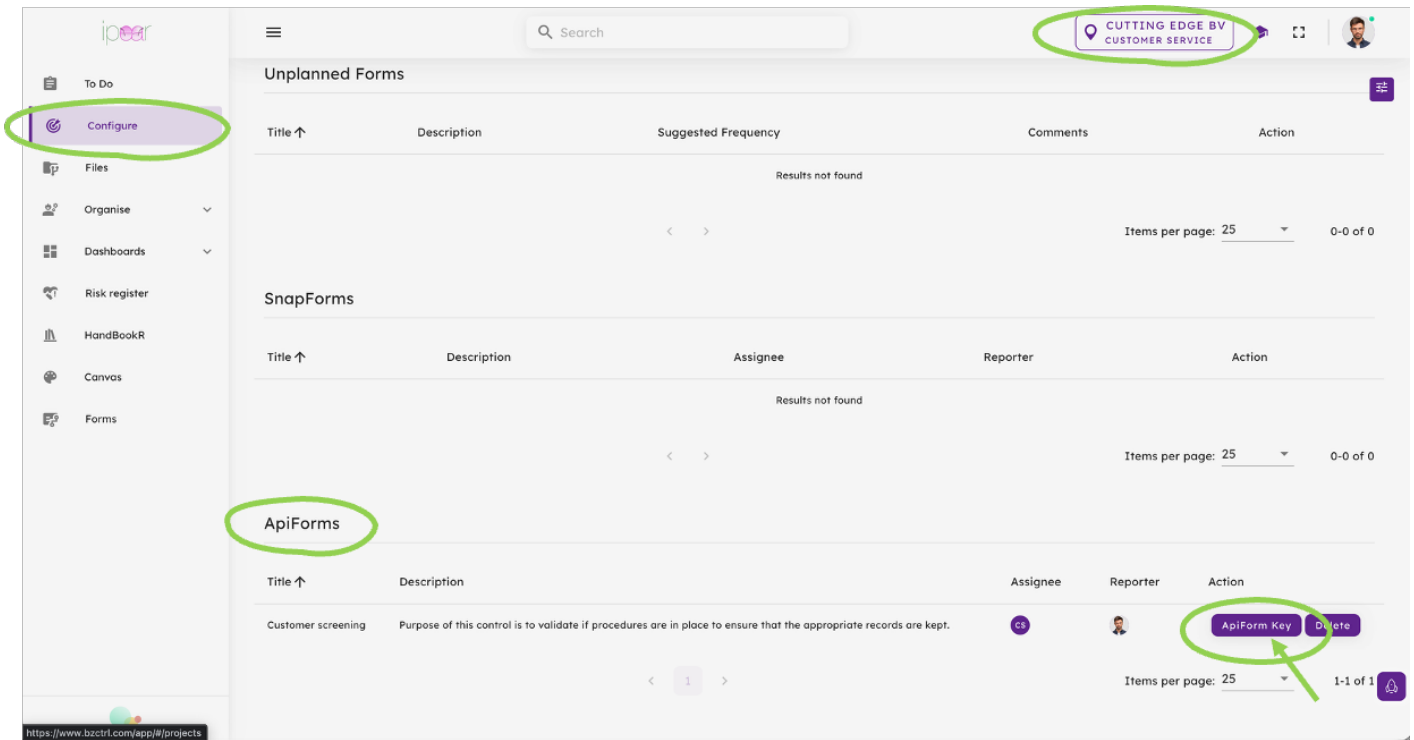
**Step 1** - From the Forms Page the User can create an ApiForm via the Action Button for a specific Form;



**Step 2** - Select the Workspace where the Form Instances should be created and stored;



**Step 3** - From the Configure page the Form is now available as ApiForm. Please make sure the correct Workspace is selected (right top). From the ApiForms Table select the ApiForm Key Button to get the details to share with your developers.



# Input to Developers External system

## Step 1 - Authentication

Before an ApiForm can be called, the external application must authenticate with BzCtrl. Authentication is performed by sending a **POST** request to the appropriate authentication endpoint using credentials provided for API communication.

## Endpoints

### Acceptance Environment (ACC)

<https://acc.bzctrl.com/bzctrl-core-api/oauth/token>

### Production Environment (PROD)

<https://bzctrl.com/bzctrl-core-api/oauth/token>

## Request Method

POST

## Request Body (x-www-form-urlencoded)

```
grant_type=password
username=cms@api.com
password=Nologicincustoms#24
```

## Sample Response

```
{
  "access_token": "xxxxxxxxxx",
  "refresh_token": "xxxxxxxxxx",
  "token_type": "Bearer",
  "expires_in": 3600,
  "scope": "*",
  "jti": "xxxxxxxxxx"
}
```

## Result

Upon successful authentication, BzCtrl. returns an **access token** that must be included in subsequent API requests. The token serves as proof of authentication and authorizes communication with the BzCtrl. API. The token remains valid for the period specified in the expires\_in field.

## Step 2 - Api Form Key

Here you see the API Endpoint url to share with your developers, which use will result in the creation of a Form Instance of the respective Form on your selected Workspace. In the Body of the message Tags can be included, which is usually used for adding Order ID's to be recorded in the Form Instance.

### ApiForm Key - Acceptance Environment (ACC)

<https://acc.bzctrl.com/bzctrl-core-api/api/v1/task/add-exposed-task/eyJwcm9qZW50SWQiOjUwMywiY29tcGFueUlkljozMzEsImNvb3Ryb2xjZCI6MTQxNSwiZW5jb2RlVXJsljoiZDMzOGExNmYtZGZzMS00OTQzLTliMzUtMzQ4YmYzMzc1ZTE1IiwieY3JlYXRvcklkIjo2NjMslndvcmtTcGFjZU5hbWUiOiJkdXN0b21zliwicmVwb3J0ZXJvc2VySWQiOjY2MywicmVwb3J0ZXJvc2VyTmFtZSI6IjI0bXkgR2VlcnRzIn0=>

### ApiForm Key - Production Environment (PROD)

<https://www.bzctrl.com/bzctrl-core-api/api/v1/task/add-exposed-task/eyJwcm9qZW50SWQiOjI4MywiY29tcGFueUlkljoxOTksImNvb3Ryb2xjZCI6MTQ1NSwiZW5jb2RlVXJsljoiNWJhM2QxMGYtNTgzMy00N2VjLWE5NzktMDk5NTM5MTNjNTM3IiwieY3JlYXRvcklkIjoxMywiY29ya1NwYWNIImFtZSI6IjI0bXkgR2VlcnRzIn0=>



After creation the Response message includes a Tasklink that can be stored to allow the user access to the Form Instance from the External system at any stage.