

# Countermeasure REST API V1

Staufen ValueStreamer GmbH

# Content

## **1. Introduction**

- 1.1 REST-API & basic technical knowledge
- 1.2 Authorization and user assignment for changes via the API
- 1.3 Countermeasure management overview

## **2. Necessary metadata for querying, creating and updating countermeasures**

- 2.1 Metadata: Use cases and API endpoints
- 2.2 Meta data REST endpoint details and examples

## **3. Countermeasure use cases and REST-API endpoints**

- 3.1 Use case and REST-API endpoint mapping
- 3.2 Fields of a countermeasure and where to get them (PDCA)
- 3.3 Fields of a countermeasure and where to insert or update them (PDCA)
- 3.4 Fields of a countermeasure and where to get them (A3)
- 3.5 REST endpoints details and examples

# 1. Introduction

# 1.1 REST-API & basic technical knowledge

- The API allows clients to fetch or insert entries via **REST** and **JSON documents**
- The API specification contains all endpoints and JSON formats in **.yaml** files as **OpenAPI 3** format
- We recommend to view the file via **Swagger UI / Editor**
- All access to the API is done via **HTTPS**
- The API uses appropriate **HTTP methods** for each action.
  - GET: For retrieving one or more entries, descriptions or meta data
  - POST: For creating an entry
  - PUT: For updating an entry or a description
  - DELETE: For deleting an entry
- The API returns corresponding **HTTP status codes** for each request, see Swagger file. E.g.
  - 2xx for success: 200 OK, 204 Deleted
  - 4xx for client errors: 400 Bad request, 401 Unauthorized, 404 Not found
  - 5xx for server errors: 500 Internal server error
- The API is reachable via <https://api-<tenant>.valuestreamer.de/api/> where <tenant> must be replaced with your tenant
- E.g.: <https://api-demo.valuestreamer.de/api/exchange/countermeasures>

Countermeasure <small>Read endpoints for both A3 and PDCA countermeasures</small>			⌵
GET	/exchange/countermeasures	Retrieve a collection of countermeasures (A3 + PDCA) by team	⌵ 🔒
GET	/exchange/countermeasures/{id}	Retrieve a specific countermeasure (A3 + PDCA)	⌵ 🔒
GET	/exchange/countermeasures/{id}/status-history	Get the status history of the countermeasure (A3 + PDCA)	⌵ 🔒
PDCA Countermeasure <small>Create, update and delete endpoints for PDCA countermeasures</small>			⌵
POST	/exchange/countermeasures/pdca	Create a countermeasure (PDCA only)	⌵ 🔒
PUT	/exchange/countermeasures/pdca/{id}	Update a countermeasure (PDCA only)	⌵ 🔒
DELETE	/exchange/countermeasures/pdca/{id}	Delete a countermeasure (PDCA only)	⌵ 🔒
PDCA Countermeasure: Description <small>Read and update the description of PDCA countermeasures</small>			⌵
GET	/exchange/countermeasures/pdca/{id}/description	Get the description (PDCA only)	⌵ 🔒
PUT	/exchange/countermeasures/pdca/{id}/description	Update the description (PDCA only)	⌵ 🔒

## 1.2 Authorization and user assignment

### Authorization

1. Authorization on the API endpoints works via basic auth per request
2. ValueStreamer Support provides username + password to the customer
3. When sending requests to the API, the basic auth credentials must be passed base64 encoded in the authorization header like this:  
Authorization: Basic 123456

### Assigned user for changes via the API

1. The ValueStreamer Support creates a ValueStreamer Application user named "API-USER"
2. Whenever a countermeasure is created, modified or deleted via the API, this API-USER is written to the changelogs (createdBy, modifiedBy, ...)

#### HTTP Example:

```
GET /api/exchange/countermeasures?team=e70ebfdd-760b-4f50-b69f-6564acb4a367&status=ALL&method=ALL&include=ALL&page=1
Host: api-demo.valuestreamer.de
Accept: application/json
Authorization: Basic
a3BpZXhjaGFuZ2U6NW1VZE1HbWpxOTNvdE9WTdDldXY=
```

#### Postman example:

The screenshot shows the Postman interface for a GET request. The URL bar contains: `{{url}}/api/exchange/deviations?status=ALL&level=ALL&team=e70ebfdd-760b-4f50-b69f-6564acb4a367&pa...`. The 'Authorization' tab is selected, showing 'Basic Au...' in the dropdown. The 'Username' field contains `{{basicUser}}` and the 'Password' field contains `{{basicPassword}}`. A 'Show Password' checkbox is checked. The interface also shows tabs for Params, Headers (8), Body, Pre-request Script, Tests, and Settings.

#### Additional information

Created by API USER	Created on 19.10.2021	Reshared on -
Changed by System Admin	Changed on 20.10.2021	Archived on -

# 1.3 Countermeasure overview

- The deviation management is available for every team by default
- To analyze and solve a deviation, countermeasures can be created and assigned to a team and a responsible user
- Countermeasures are either of type A3 or PDCA
  - **A3 countermeasures**
    - The A3 report follow a structured problem-solving and continuous-improvement approach with various workflow steps, input fields and rules
    - The API supports the reading of various fields of the A3 process via GET request
    - The API does not support creating, updating or deleting A3 reports. This is only possible via user interface
  - **PDCA countermeasures**
    - PDCA measures can be read as well as created, updated and deleted via the API
    - The progress of a PDCA countermeasure can be set to PLAN, DO, CHECK, ACT
    - When closing a PDCA countermeasure by setting the progress to ACT, the user can decide whether to also close the corresponding deviation or not, when the deviation has no other open countermeasures
    - Related to the PDCA countermeasure, various tasks can be created as implementation plan. The list of tasks is also returned via API
- A countermeasure can be cascaded which means that the deviation belongs to "Team A", but the countermeasure is assigned to "Team B" as responsible team
- Each countermeasure has a status which is either green, yellow or red

Countermeasures

35 0 34 11 0 7 1

Active A3-Reports Overdue Cascaded For information Closed Global Archive (A3)

Top issue Go & See Feedback Delete

Deviation	Countermeasure	Priority/Team	Responsible	Method	End (plan)	Progress/Linked with
Abweichung	Maßnahme	Project Management ...	Christina Kuhn	PDCA	22.02.20...	100%
Machine Failure	Call maintenance	Project Management ...	Christina Kuhn	PDCA	23.02.20...	100%
Test	Maßnahme	Project Management ...	Vincent Zimmerm...	PDCA	11.03.20...	100%
Beispiel	Material über Logistik bescha...	Project Management ...	Christina Kuhn	PDCA	26.02.20...	100%
Drilling machine broken	Example Counter Measure	Project Management ...	Christina Kuhn	PDCA	26.02.20...	100%
Maschinendefekt Z74	Beispiel Maßnahme	Project Management ...	Christina Kuhn	PDCA	05.03.20...	100%
Deviation example	Cascaded Measure from MPS	Assembly	Marco Kappel	PDCA	05.03.20...	100%

Countermeasures

Prüfung mit Inszenierung 24.04.2022 Marco Kappel

Immediate countermeasure (PDCA)

Task: Assembly Responsible: Marco Kappel End date: 20.04.2022

Priority

For information at

Description

Implementation plan (Tasks)

Fast entry

Actions	Title	Responsible	End date (target)	Progress
✓	Q-Audit		16.03.2021	Archived

PDCA

Countermeasures

Follow-up Actions 28.03.2022 Christina Kuhn

A3

Background Problem Statement Goal Statement Root Cause Implementation Follow-up

Basic information

Title of the problem: Not enough material

Responsible: Christina Kuhn Support: Ralph Walter Priority

Team members

Christina Kuhn Ralph Walter

Planning

Step	Name of the step	Start (actual)	End (plan)	End (plan new)	End (actual)
1	Background and relevance	07.03.2022			07.03.2022
2	Problem Statement	07.03.2022			07.03.2022
3	Goals Statement	07.03.2022			07.03.2022

## 2. Necessary meta data for querying, creating and updating countermeasures

## 2.1 Meta data: Use cases and API endpoints

- The following table describes how to get all the data you need for interactions with the API

#	Use case	Use case description	HTTP method	API URL
1	<b>Team: Get the list of teams</b>	Returns all teams with id and name. The id can be used to fetch the team details.	GET	/exchange/teams
2	<b>Team: Get a team's details (by team id)</b>	Returns the details of a team by its team id: <ul style="list-style-type: none"><li>▪ Team's name and short name</li><li>▪ List of team members</li></ul>	GET	/exchange/teams/{teamid}
3	<b>Get the list of all users</b>	Returns all existing users with id and name	GET	/exchange/users

For more details, please have a look into the OpenAPI files.



## 2.2 #1 GET /exchange/teams

### Get the list of teams

- The endpoint returns a lists of all teams existing on the tenant
- Accept: application/json

**Request:**  
GET /api/exchange/teams HTTP/1.1  
Host: api-demo.valuestreamer.de  
Accept: application/json  
Authorization: Basic a3BpZXhjaGFuZ2U6NW1VZE1HbWpxOTNvdE9WTDdldXY=

**Response:**

```
[
  {
    "id": "5bcaa97a-d0c1-11ea-87d0-0242ac130003",
    "name": "Linie 1",
    "shortName": "Linie 1",
    "parentTeam": "5bcaa8b2-d0c1-11ea-87d0-0242ac130003"
  },
  {
    "id": "5bcaaa38-d0c1-11ea-87d0-0242ac130003",
    "name": "Linie 2",
    "shortName": "Linie 2",
    "parentTeam": "5bcaa8b2-d0c1-11ea-87d0-0242ac130003"
  },
  {
    "id": "d72fb249-af46-43d8-a116-afd5b32815ef",
    "name": "Log-X",
    "shortName": "Log-X",
    "parentTeam": "14321ce0-ba75-4336-ab26-1b13ebc33114"
  },
  {
    "id": "5bca9930-d0c1-11ea-87d0-0242ac130003",
    "name": "Management",
    "shortName": "Management",
    "parentTeam": "8ba7de30-41b8-4fb3-8d19-dd6c9e92ead7"
  }
]
```

## 2.2 #2 GET /exchange/teams/{id}

### Get team details

- The endpoint returns the details of a team, like the teams members
- Accept: application/json

**Request:**  
GET /api/exchange/teams HTTP/1.1  
Host: api-demo.valuestreamer.de  
Accept: application/json  
Authorization: Basic a3BpZXhjaGFuZ2U6NW1VZE1HbWpxOTNvdE9wTDdldXY=

**Response:**

```
{
  "id": "5bcaa97a-d0c1-11ea-87d0-0242ac130003",
  "name": "Linie 1",
  "shortName": "Linie 1",
  "parentTeam": "5bcaa8b2-d0c1-11ea-87d0-0242ac130003",
  "members": [
    {
      "id": "e3373620-d0c2-11ea-87d0-0242ac130003",
      "firstName": "Nino",
      "lastName": "Baresi",
      "email": "Nino.Baresi@valuestreamer-demo.de"
    },
    {
      "id": "e3375560-d0c2-11ea-87d0-0242ac130003",
      "firstName": "Jürgen",
      "lastName": "Seiler",
      "email": "Jürgen.Seiler@valuestreamer-demo.de"
    }
  ],
  "processBoards": [
    {
      "id": "78f748e5-0ad5-4f1a-951e-7bcc7cc94d97",
      "name": "log-x"
    },
    {
      "id": "d40f30ae-f509-4126-bd71-73da5df21967",
      "name": "KVP Projekte"
    }
  ]
}
```

## 2.2 #3 GET /exchange/users

### Get the list of users

- The endpoint returns all users present on the tenant
- Accept: application/json

**Request:**  
GET /api/exchange/users HTTP/1.1  
Host: api-demo.valuestreamer.de  
Accept: application/json  
Authorization: Basic a3BpZXhjaGFuZ2U6NW1VZE1HbWpxOTNvdE9WTdDldXY=

**Response:**

```
[
  {
    "id": "e3372d60-d0c2-11ea-87d0-0242ac130003",
    "firstName": "Daniel",
    "lastName": "Ackermann",
    "email": "Daniel.Ackermann@valuestreamer-demo.de"
  },
  {
    "id": "e3372fcc-d0c2-11ea-87d0-0242ac130003",
    "firstName": "Martina",
    "lastName": "Amsel",
    "email": "Martina.Amsel@valuestreamer-demo.de"
  },
  {
    "id": "e3373314-d0c2-11ea-87d0-0242ac130003",
    "firstName": "Sarah",
    "lastName": "Andersen",
    "email": "Sarah.Andersen@valuestreamer-demo.de"
  },
  ...
]
```

# 3. Countermeasure management use cases and REST-API endpoints

## 3.1 Use case and REST-API endpoint mapping

#	Use case	A3	PDC A	HTTP Method	API URL
1	Get a collection of countermeasures according to the query parameters	X	X	GET	/exchange/countermeasures?{query-parameters}
2	Get the given countermeasure and related fields	X	X	GET	/exchange/countermeasures/{id}
3	Get the status history of the given countermeasure	X	X	GET	/exchange/countermeasures/{id}/status-history
4	Create a new PDCA countermeasure		X	POST	/exchange/countermeasures/pdca
5	Update the given PDCA countermeasure		X	PUT	/exchange/countermeasures/pdca/{id}
6	Delete the given PDCA countermeasure		X	DELETE	/exchange/countermeasures/pdca/{id}
7	Get the description of the given PDCA countermeasure		X	GET	/exchange/countermeasures/pdca/{id}/description
8	Update the description of the given PDCA countermeasure		X	PUT	/exchange/countermeasures/pdca/{id}/description

For more details, please have a look into the OpenAPI Files.

## 3.2 Fields of a PDCA countermeasure and where to get them as response

Machine downtime

Close Apply changes

Deviation Countermeasures

Countermeasures

Prüfung mit Instandhaltung 20.04.2022 Marco Kappel

Immediate countermeasure (PDCA)

Title (Description of the solution) Prüfung mit Instandhaltung

Team Assembly Responsible Marco Kappel End (plan) 20.04.2022

Priority

For information at

Description

10/08/2021 02:02 PM (Christina Kuhn)

Implementation plan (Tasks)

Fast entry

Actions	Title	Responsible	End date (target)	Progress
✓	Q-Audit		16.03.2021	Archived

Additional information

Created by Dominik Fischer Created on 21.09.2020 Origin Adherence to schedules (

Changed by Christina Kuhn Changed on 22.03.2022 Archived on -

#	Field	Format	JSON field name	List of endpoints:			
				1. GET /countermeasures?{query_parameters}	2. GET /countermeasures/{id}	3. GET /countermeasures/{id}/status-history	4. GET /countermeasures/{id}/pdca/description
				1	2	3	4
	id	UUID	id	✓	✓		
	Deviation ID	UUID	deviationId	✓	✓		
1	Title	String	title	✓	✓		
2	Team	UUID	responsibleTeam	✓	✓		
3	Responsible	UUID	responsiblePerson	✓	✓		
4	End (plan)	Date	endDatePlan		✓		
5	status	Int	status	✓	✓	✓	
	statusMessage	String	statusMessage		✓	✓	
6	Progress	String	progress	✓	✓		
7	Priority	Boolean	priority	✓	✓		
8	Created By	UUID	createdBy		✓		
9	Created On	Date	createdOn		✓		
10	Origin	UUID	origin		✓		
11	Last changed by	UUID	changedBy		✓		
12	Last changed on	Date String	changedOn		✓		
13	Archived on	Date string	closedOn		✓		
14	Description	String					✓
15	List of tasks	Object	pdcaAttributes.implementationPlan		✓		

# 3.3 Fields of a PDCA countermeasure and where to insert or update them

Close

Machine downtime

Apply changes

Deviation

Countermeasures

Countermeasures

Prüfung mit Instandhaltung  
20.04.2022 Marco Kappel

Immediate countermeasure (PDCA)

Title (Description of the solution)

Prüfung mit Instandhaltung

Team

Assembly

Responsible

Marco Kappel

End (plan)

20.04.2022

Progress

Priority

7

For information at

Team

Description

10/08/2021 02:02 PM (Christina Kuhn)

Implementation plan (Tasks)

Fast entry

Actions

Title

Responsible

End date (target)

Progress

✓

Q-Audit

16.03.2021

Archived

Additional information

Created by

Dominik Fischer

Created on

21.09.2020

Origin

Adherence to schedules (

Changed by

Christina Kuhn

Changed on

22.03.2022

Archived on

-

#	Field	Format	JSON field name	List of endpoints:		
				1. POST /countermeasures/pdca	2. PUT /countermeasures/pdca/{id}	3. PUT /countermeasures/pdca/{id}/description
				1	2	3
	id	UUID	id			
	Deviation ID	UUID	deviationId	✓		
1	Title	String	title	✓	✓	
2	Team	UUID	responsibleTeam	✓		
3	Responsible	UUID	responsiblePerson	✓	✓	
4	End (plan)	Date	endDatePlan	✓	✓	
5	status	Int	status	✓	✓	
	statusMessage	String	statusMessage	✓	✓	
6	Progress	String	progress	✓	✓	
7	Priority	Boolean	priority	✓	✓	
8	Created By	UUID	createdBy			
9	Created On	Date	createdOn			
10	Origin	UUID	origin			
11	Last changed by	UUID	changedBy			
12	Last changed on	Date	changedOn			
13	Archived on	Date	closedOn			
14	Description	String				✓
15	List of tasks	Object	pdcaAttributes.implementationPlan			

## 3.4 Fields of a A3 countermeasure and where to get them as response

Countermeasures +

Follow-up Actions
28.03.2022 Christina Kuhn

Back Next

Background Problem Statement Goal Statement Cause-and-Effect Root-Cause Implementation Plan Follow-up Actions

Basic information ?

Title of the problem  
Not enough material 1

Team Line 1 2 Responsible Christina Kuhn 3 Sponsor Ralph Walter 5 Priority 7

Team members

Christina Kuhn RW Ralph Walter

Planning 14

Step	Name of the step	Start (actual)	End (plan)	End (plan new)	End (actual)
1	Background and relevance	07.03.2022			07.03.2022
2	Problem Statement	07.03.2022			07.03.2022
3	Goals Statement	07.03.2022			07.03.2022
4	Cause-and-Effect Analysis	07.03.2022			07.03.2022
5	Root-Cause Analysis	07.03.2022			07.03.2022
6	Derive the implementation plan	07.03.2022			07.03.2022
7	Follow-up Actions	07.03.2022	28.03.2022 4		07.03.2022

Additional information

Created by Dominik Fischer 8 Created on 21.09.2020 9 Origin Adherence to schedules ( 10

Changed by Christina Kuhn 11 Changed on 22.03.2022 12 Archived on - 13

#	Field	Format	JSON field name	List of endpoints:		
				1. GET /countermeasures/{query_parameters} 2. GET /countermeasures/{id} 3. GET /countermeasures/{id}/status-history		
				1	2	3
	id	UUID	id	✓	✓	
	Deviation ID	UUID	deviationId	✓	✓	
1	Title	String	title	✓	✓	
2	Team	UUID	responsibleTeam	✓	✓	
3	Responsible	UUID	responsiblePerson	✓	✓	
4	End (plan)	Date	endDatePlan		✓	
	Planning dates (start / end)	Object	a3Attributes		✓	
5	Sponsor	UUID	sponsor		✓	
6	Status	Int	status	✓	✓	✓
	statusMessage	String	statusMessage		✓	✓
7	Priority	Boolean	priority	✓	✓	
8	Created By	UUID	createdBy		✓	
9	Created On	Date	createdOn		✓	
10	Origin	UUID	origin		✓	
11	Last changed by	UUID	changedBy		✓	
12	Last changed on	Date	changedOn		✓	
13	Archived on	Date	closedOn		✓	
14	List of planning dates	Object	a3Attributes		✓	



## 3.5 #1 GET /exchange/countermeasures?{query-parameters}

### Retrieve a collection of countermeasures according to the query parameters

- The endpoint returns all countermeasures for a team that match the given query parameters
  - Filter attributes as req. parameters:
    - **&team=** → id of the team
    - **&status=**
      - ACTIVE: Only includes countermeasures that are not done (progress neq. ACT)
      - CLOSED: Only includes countermeasures that are done (progress eq. ACT)
      - ALL: Includes both, ACTIVE + CLOSED countermeasures
    - **&method=**
      - ALL: Includes both A3 and PDCA countermeasures
      - PDCA: Only includes PDCA countermeasures
      - A3: Only includes A3 countermeasures
    - **&include=**
      - RESPONSIBLE: Only includes countermeasures for which the team is responsible
      - CASCADED: Only includes cascaded countermeasures that have been created in the team, but assigned to another team as responsible team
      - ALL: Includes both responsible and cascaded countermeasures
    - **&page=1** → Page number, max. 200 entries per page
  - Accept: application/json
- Bold = required / mandatory**

```
Request:
GET /api/exchange/countermeasures?team=e70ebfdd-760b-4f50-b69f-6564acb4a367&status=ALL&method=ALL&include=ALL&page=1 HTTP/1.1
Host: api-demo.valuestreamer.de
Accept: application/json
Authorization: Basic a3BpZXhjaGFuZ2U6NW1VZE1HbWpxOTNvdE9WTDDdLdXY=
```

```
Response:
{
  "page": {
    "page": 1,
    "elementsOnPage": 75,
    "elementsPerPage": 200,
    "totalPages": 1,
    "totalElements": 75
  },
  "entries": [
    {
      "id": "95c0a07e-b4a6-48f0-a702-be44620b46bc",
      "closed": false,
      "deviationId": "9770be21-ba1a-4b45-a21c-92fcd62a4e3",
      "method": "PDCA",
      "priority": true,
      "progress": "PLAN",
      "responsiblePerson": "ffa6fda3-e10f-4e86-bd3a-6314e7e28ee6",
      "responsibleTeam": "e70ebfdd-760b-4f50-b69f-6564acb4a367",
      "status": 1,
      "title": "Order cookies"
    }, ...
  ]
}
```

## 3.5 #2 GET /exchange/countermeasures/{id}

### Retrieve the given countermeasure and each of its fields

- The endpoint returns all fields and additional information of the given countermeasure
- For the full list of properties, please have a look at the OpenAPI specification
- When the countermeasure **method** is **PDCA**:
  - the attribute **pdcaAttributes** contains the list of tasks in the implementationPlan array
  - The **description** richtext field of the PDCA countermeasure can be retrieved with endpoint #8
  - The attribute **a3Attributes** is **null**
- When the countermeasure **method** is **A3**:
  - the attribute **a3Attributes** contains the sponsor (user id) and all planning related steps and dates
  - The attribute **pdcaAttributes** is **null**
- Accept: application/json

```
Request:
GET /api/exchange/countermeasures/61d93096-3235-4b6d-8fef-1ca2935e2ee6
HTTP/1.1
Host: api-demo.valuestreamer.de
Accept: application/json
Authorization: Basic a3BpZXhjaGFuZ2U6NW1VZE1HbWpxOTNvdE9WTDdldXY=
```

```
Response:
{
  "id": "95c0a07e-b4a6-48f0-a702-be44620b46bc",
  "changedBy": "785a94d2-97d8-43b3-aa55-9e7141571ee7",
  "changedOn": "2022-05-06",
  "closed": false,
  "closedOn": null,
  "createdBy": "785a94d2-97d8-43b3-aa55-9e7141571ee7",
  "createdOn": "2022-05-06",
  "deviationId": "9770be21-ba1a-4b45-a21c-92fcd62a4e3",
  "deviationTitle": "api-test-1",
  "endDatePlan": "2022-05-06",
  "method": "PDCA",
  "origin": "Accounting",
  "priority": true,
  "progress": "PLAN",
  "responsiblePerson": "ffa6fda3-e10f-4e86-bd3a-6314e7e28ee6",
  "responsibleTeam": "e70ebfdd-760b-4f50-b69f-6564acb4a367",
  "status": 1,
  "statusMessage": "ok",
  "title": "Order cookies",
  "pdcaAttributes": {
    "implementationPlan": [{
      "id": "5becab2b-e915-424b-a333-e27235814040",
      "endDatePlan": "2022-05-10",
      "progress": "CREATED",
      "responsiblePerson": "71c8e26d-6760-4c3f-abdf-6606b8872052",
      "responsibleTeam": "e70ebfdd-760b-4f50-b69f-6564acb4a367",
      "title": "Make a choice and order"
    }]
  },
  "a3Attributes": {
    "sponsor": "00140d5b-6b52-4ebb-b983-a7ca2b442857",
    "planning": {
      "background": {
        "step": 1,
        "name": "Background and relevance",
        "startDateActual": "2022-05-11",
        "endDateActual": null,
        "endDatePlan": null,
        "endDatePlanNew": null
      }
    }
  }
}
```

## 3.5 #3 GET /exchange/countermeasures/{id}/status-history

### Retrieve the status history

- Returns a list of all status history entries of the given countermeasure
- Mapping of the colors and status
  - 1: green
  - 2: yellow
  - 3: red
- Accept: application/json

```
Request:
GET /api/exchange/countermeasures/61d93096-3235-4b6d-8fef-
1ca2935e2ee6/status-history HTTP/1.1
Host: api-demo.valuestreamer.de
Accept: application/json
Authorization: Basic a3BpZXhjaGFuZ2U6NW1VZE1HbWpxOTNvdE9wTDdldXY=
```

```
Response:
[
  {
    "comment": "inventory",
    "changedBy": "785a94d2-97d8-43b3-aa55-9e7141571ee7",
    "changedOn": "2022-05-06",
    "status": 2
  }
]
```

## 3.5 Closing deviations and countermeasures

The following information is important for the countermeasure POST and PUT endpoints and the closing of countermeasures by setting their progress to ACT.

- A countermeasure always belongs to a deviation. A deviation can be closed when all related countermeasures are closed
- The API supports to close a countermeasure and a deviation either in separate requests (via countermeasure API and deviation API) or in a single request via countermeasure API. Closing both in a single transaction is possible by submitting the countermeasures progress as ACT and the property closeDeviation as TRUE. However, there are some rules to follow when closing deviations, depending on the deviation type
- A deviation is either a single deviation, a combined parent deviation or a child deviation related to a combined parent
  - **Single deviation**
    - Single deviations are not combined resp. related to another deviation
    - Single deviations can be closed directly, when all related countermeasures are closed
  - **Combined parent deviation**
    - A combined parent deviation is created based on multiple single deviations, which then become child deviations of the combined parent deviation
    - Countermeasures can be assigned either to the child deviations or the combined parent deviation
    - To close a combined parent deviation, both the combined parent deviation and all child deviations must not have open countermeasures
    - When a combined parent deviation is closed, its child deviations are also closed
  - **Child deviation of a combined parent deviation**
    - A combined child deviation is related to a parent deviation
    - A child deviation can't be closed directly. A child deviation is only closed, when the parent deviation is closed
    - When the API client tries to close a countermeasure related to a child deviation, the API will not only try to close the related child deviation, but also the combined parent deviation and all of its child deviations
- When the client tries to close a countermeasure and a deviation in a single request by sending closeDeviation:true, but the related single, parent or child deviations have open countermeasures, the server will **reject** the full POST/PUT request for the countermeasure. Therefore, use this function wisely to avoid problems or confusion with rejected requests due to violations of the above-described logic. Alternatively, deviations can be also closed via UI or the deviation API.

## 3.5 #4 POST /exchange/countermeasures/pdca

### Create a new PDCA countermeasure

- The endpoint creates a new PDCA countermeasure
- Property: **deviationId**: A countermeasure is linked to a deviation by the deviationId. The deviation must not be closed, otherwise the server responds with HTTP 409
- Property: **endDatePlan**: Planned end date as yyyy-mm-dd
- Property: **priority**: True or false
- Property: **progress**: Possible values are PLAN, DO, CHECK or ACT
- Property: **closeDeviation**: Indicator to close the related deviation, when the countermeasure is closed (ACT)
  - Value: true/false
  - ☒ When the countermeasures progress is set to ACT and when closeDeviation is FALSE, the request will close the countermeasure, but the deviation remains open. The server respond with HTTP 200 OK
  - ☒ When the countermeasures progress is set to ACT and when closeDeviation is TRUE, the request will close the countermeasure and the corresponding deviation, when the deviation or the related parent deviation and its childs have no other open countermeasures. The server respond with HTTP 200 OK
  - ☒ When the countermeasures progress is set to ACT and when closeDeviation is TRUE, but the deviation or the related parent deviation and its childs have open countermeasures, the request is rejected. Neither the countermeasure nor the deviation will be closed and the server will respond with an error / HTTP 422
- Property **responsiblePerson** and **responsibleTeam**: Related UUIDs of the user / team
- Property **status**: Possible values: 1 (green), 2 (yellow), 3 (red)
- Property **statusMessage**
- Property: **title**
- Accept: application/json – Content-Type: application/json

**Bold = required / mandatory**

**Request:**  
POST /api/exchange/countermeasures/pdca HTTP/1.1  
Host: api-demo.valuestreamer.de  
Accept: application/json  
Authorization: Basic a3BpZXhjaGFuZ2U6NW1VZE1HbWpxOTNvdE9WTDdldXY=

```
{
  "deviationId": "9770be21-ba1a-4b45-a21c-92fcd62a4e3",
  "endDatePlan": "2022-05-06",
  "priority": true,
  "progress": "PLAN",
  "closeDeviation": false,
  "responsiblePerson": "ffa6fda3-e10f-4e86-bd3a-6314e7e28ee6",
  "responsibleTeam": "e70ebfdd-760b-4f50-b69f-6564acb4a367",
  "status": 1,
  "statusMessage": ""
}
```

**Response:**

```
{
  "title": "Order cookies",
  "id": "95c0a07e-b4a6-48f0-a702-be44620b46bc",
  "changedBy": "785a94d2-97d8-43b3-aa55-9e7141571ee7",
  "changedOn": "2022-05-06",
  "closed": false,
  "closedOn": null,
  "createdBy": "785a94d2-97d8-43b3-aa55-9e7141571ee7",
  "createdOn": "2022-05-06",
  "deviationId": "9770be21-ba1a-4b45-a21c-92fcd62a4e3",
  "deviationTitle": "api-test-1",
  "endDatePlan": "2022-05-06",
  "method": "PDCA",
  "origin": "Accounting",
  "priority": true,
  "progress": "PLAN",
  "responsiblePerson": null,
  "responsibleTeam": "e70ebfdd-760b-4f50-b69f-6564acb4a367",
  "status": 1,
  "statusMessage": "",
  "title": "Order cookies",
  "pdcaAttributes": {
    "implementationPlan": []
  },
  "a3Attributes": null
}
```

## 3.5 #5 PUT /exchange/countermeasures/pdca/{id}

### Update the given countermeasure

- The request updates the given countermeasure (PDCA)
- Countermeasures, which are already closed (progress eq. ACT), can't be updated anymore
- All fields are mandatory, except closeDeviation
- The endpoint creates a new PDCA countermeasure
- Property: **endDatePlan**: Planned end date as yyyy-mm-dd
- Property: **priority**: true or false
- Property: **progress**: Possible values are PLAN, DO, CHECK or ACT
- Property: **closeDeviation**: Indicator to close the related deviation, when the countermeasure is closed (ACT)
  - ☒ When the countermeasures progress is set to ACT and when closeDeviation is FALSE, the request will close the countermeasure, but the deviation remains open. The server respond with HTTP 200 OK
  - ☒ When the countermeasures progress is set to ACT and when closeDeviation is TRUE, the request will close the countermeasure and the corresponding deviation, when the deviation or the related parent deviation and its childs have no other open countermeasures. The server respond with HTTP 200 OK
  - ☐ When the countermeasures progress is set to ACT and when closeDeviation is TRUE, but the deviation or the related parent deviation and its childs have open countermeasures, the request is rejected. Neither the countermeasure nor the deviation will be closed and the server will respond with an error / HTTP 422
- Property **responsiblePerson**: Related UUID of the user
- Property **status**: Possible values: 1 (green), 2 (yellow), 3 (red)
- Property **statusMessage**
- Property: **title**
- Accept: application/json – Content-Type: application/json

```
Request:
PUT /api/exchange/countermeasures/pdca/d80673f9-5011-4e86-88e2-
278c01ac6dfb HTTP/1.1
Host: api-demo.valuestreamer.de
Accept: application/json
Authorization: Basic a3BpZXhjaGFuZ2U6NW1VZE1HbWpxOTNvdE9WTDdldXY=
{
  "endDatePlan": "2022-05-06",
  "priority": true,
  "progress": "DO",
  "closeDeviation": true,
  "responsiblePerson": "ffa6fda3-e10f-4e86-bd3a-6314e7e28ee6",
  "status": 1,
  "statusMessage": "ok",
  "title": "Order cookies"
```

```
Response:
{
  "id": "95c0a07e-b4a6-48f0-a702-be44620b46bc",
  "changedBy": "785a94d2-97d8-43b3-aa55-9e7141571ee7",
  "changedOn": "2022-05-06",
  "closed": false,
  "closedOn": null,
  "createdBy": "785a94d2-97d8-43b3-aa55-9e7141571ee7",
  "createdOn": "2022-05-06",
  "deviationId": "9770be21-ba1a-4b45-a21c-92fcd62a4e3",
  "deviationTitle": "api-test-1",
  "endDatePlan": "2022-05-06",
  "method": "PDCA",
  "origin": "Accounting",
  "priority": true,
  "progress": "DO",
  "responsiblePerson": "ffa6fda3-e10f-4e86-bd3a-6314e7e28ee6",
  "responsibleTeam": "e70ebfdd-760b-4f50-b69f-6564acb4a367",
  "status": 1,
  "statusMessage": "",
  "title": "Order cookies",
  "pdcaAttributes": {
    "implementationPlan": []
  },
  "a3Attributes": null
}
```

## 3.5 #6 DELETE /exchange/countermeasures/pdca/{id}

### Delete a countermeasure

- The endpoint deletes the given PDCA countermeasure
- When a countermeasure is deleted, entries of the following types related to the countermeasure are also deleted:
  - Go + see
  - Top issue
  - Feedback
- Tasks that are linked to the countermeasure are not deleted, but the tasks relation to the countermeasure is removed and the task title is modified as „PARENT DELETED: <tasktitle>“
- Accept: application/json

**Request:**  
DELETE /exchange/countermeasures/pdca/54e0f2ab-9c22-402f-a915-f63b4c4b7071  
HTTP/1.1  
**Host:** api-demo.valuestreamer.de  
**Accept:** application/json  
**Authorization:** Basic a3BpZXhjaGFuZ2U6NW1VZE1HbWpxOTNvdE9WTDdldXY=

**Response:**

## 3.5 #7 GET /exchange/countermeasures/pdca/{id}/description

### Get the description of the given PDCA countermeasure

- The endpoint returns the description of the given PDCA countermeasure as HTML code
- **Accept: text/html**

**Request:**  
GET /api/exchange/countermeasures/pdca/81beaa9b-b510-4924-af84-0fe651de8fcb/description HTTP/1.1  
**Host:** api-demo.valuestreamer.de  
**Accept:** text/html  
**Authorization:** Basic a3BpZXhjaGFuZ2U6NW1VZE1HbWpxOTNvdE9WTDdldXY=

**Response:**  
<p>Your text could stand here</p>



## 3.5 #8 PUT /exchange/countermeasures/pdca/{id}/description

### Update the description of the given PDCA countermeasure

- The endpoint updates the description of the given PDCA countermeasure
- Valid HTML code must be sent in the request body
- **Accept: text/html – Content-Type: text/html**

**Request:**  
PUT /api/exchange/countermeasure/pdca/54e0f2ab-9c22-402f-a915-f63b4c4b7071/description HTTP/1.1  
**Host:** api-demo.valuestreamer.de  
**Accept:** 'text/html'  
**Authorization:** Basic a3BpZXhjaGFuZ2U6NW1VZE1HbWpxOTNvdE9WTdDldXY=  
  
'<p>Get some cookies!</p>'

**Response:**  
'<p>Get some cookies!</p>'

valueSstreamer

ECHTE ZUSAMMENARBEIT