

---

# SAP Fieldglass

## Approval API Requirements Document v1

April 17, 2017

---

## Table of Contents

Description .....	3
Authentication and Authorization .....	3
Using OAuth 2.0 to Access SAP Fieldglass APIs .....	3
Resource URL .....	5
Retrieve All Work Items for Approval .....	5
Response Body Elements .....	5
Retrieve All Work Items for Approval by Module .....	8
Response Body Elements .....	8
Retrieve Details for Specific Work Item Requiring Approval .....	10
Response Body Elements .....	11
Approve Work Item .....	13
Approval Post Body Elements .....	14
Get Rejection Reasons .....	14
Response Body Elements .....	14
Reject Work Item .....	16
Rejection Post Body Elements .....	16
HTTP Codes .....	17
Successful Requests .....	17
HTTP Success Codes .....	17
Failed Requests .....	17
HTTP Error Codes .....	17
Appendix A: Module ID Reference .....	18

## Description

The SAP Fieldglass approvals resource allows clients to approve or reject SAP Fieldglass work items. Clients retrieve the work items pending an approval action from the authenticated user and return the workflow action to be performed (either approve or reject).

The list and drill down details provided by the approvals resource give clients:

- the ability to present work items out for approval to their users with the level of detail needed to make an informed approval decision, and
- the mechanism to return the approval or rejection decision

## Authentication and Authorization

### Using OAuth 2.0 to Access SAP Fieldglass APIs

SAP Fieldglass APIs use the OAuth 2.0 protocol for authentication and authorization. SAP Fieldglass supports common OAuth 2.0 scenarios, such as those for web server, installed, and client-side applications.

To begin, obtain OAuth 2.0 client credentials from your SAP Fieldglass representative. Then your client application requests an access token from the SAP Fieldglass Authorization Server, extracts a token from the response, and sends the token to the SAP Fieldglass API that you want to access.

All applications follow a basic pattern when accessing a SAP Fieldglass API using OAuth 2.0. At a high level, you follow three steps:

1. Obtain OAuth 2.0 credentials from SAP Fieldglass.

Your SAP Fieldglass representative must provide you with two key pieces of information:

- URL: `https://<SAP Fieldglass environment URL>/api/oauth2/v2.0`
- Client-specific application key.

2. Obtain an access token from the SAP Fieldglass Authorization Server.

Before your application can access private data using an SAP Fieldglass API, it must obtain an access token that grants access to that API. A single access token can grant varying degrees of access to multiple APIs.

This is done by making the following `x-www-form-urlencoded` HTTP POST request, specific to username/password credentials. The following sample is relevant information on utilizing SAML to obtain the access token:

#### Sample Request for token with user/password credentials

```
POST /api/oauth2/v2.0/token
Authorization: Basic {Base64Encoded(user:credential)}
Content-Type: application/x-www-form-urlencoded
X-ApplicationKey: {server provided key}

grant_type=client_credentials&response_type=token
```

### Sample Response

```
{"access_token" : "WDXlKj3TTOn3rpg9GHnZpbKmvj1=",  
"token_type" : "Bearer","expires_in" : 7200}
```

### OAuth 2.0 SAML Bearer Assertion Flow

The OAuth 2.0 SAML bearer assertion flow is similar to any other token flow within OAuth. The SAML assertion is POSTed to the OAuth token endpoint, which in turn processes the assertion, and issues an `access_token` based upon prior approval of the application. However, the client does not need a `client_secret` to be passed to the token endpoint.

### sample request for token with SAML

```
POST  
/api/oauth2/v2.0/token?grant_type=client_credentials&response_type=token&SAMLRe  
sponse={Base64Encoded SAML Assertion}  
Authorization: Basic {Base64Encoded(user:credential)}  
Content-Type: application/x-www-form-urlencoded  
X-ApplicationKey: {server provided key}
```

### sample response

```
{"access_token" : "WDXlKj3TTOn3rpg9GHnZpbKmvj1=",  
"token_type" : "Bearer","expires_in" : 7200}
```

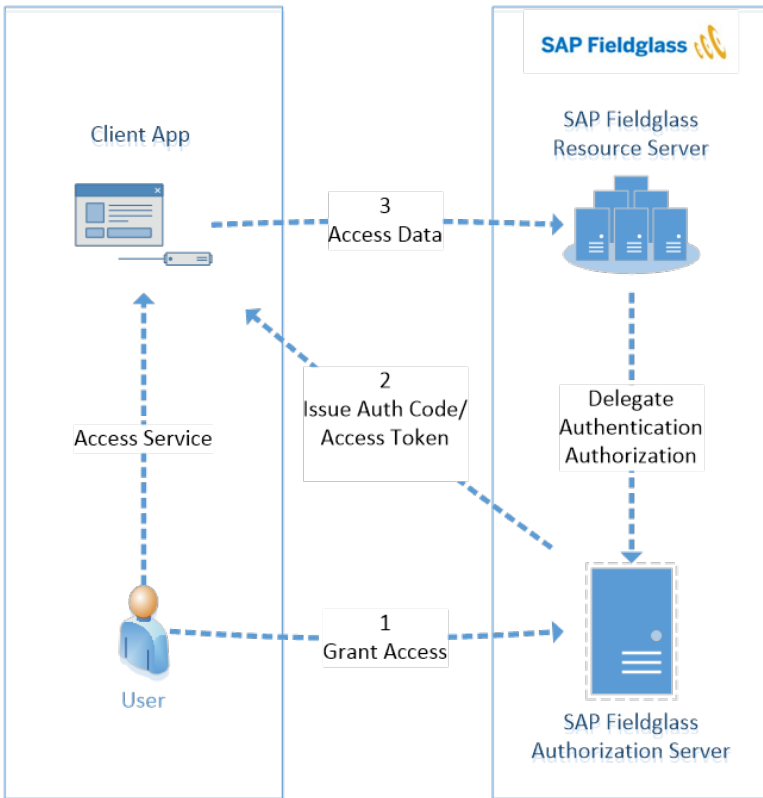
### 3. Send the access token to an SAP Fieldglass API.

After an application obtains an access token, it sends the token to a SAP Fieldglass API in an HTTP authorization header. Access tokens are valid only for the set of operations and resources described in the scope of the token request.

For example, if an access token is issued for the SAP Fieldglass Approvals API, it does not grant access to the SAP Fieldglass Timesheet API. You can, however, send that access token to the SAP Fieldglass Approvals API multiple times for similar operations.

Access tokens have limited lifetimes. If your application needs access to a SAP Fieldglass API beyond the lifetime of a single access token, you must acquire a new token.

The following diagram provides a graphical view of the authorization and authentication flow:



## Resource URL

<https://www.fieldglass.net/api/v1/approvals/>

## Retrieve All Work Items for Approval

The following request statement returns a list of work items for approval:

<b>GET</b>	<code>/api/v1/approvals/</code>
------------	---------------------------------

## Response Body Elements

The following table describes the data elements contained in the response. In addition:

- Other elements may be provided depending on the type of object. For example, job posting will return additional creator and owner details.
- Fields marked with an asterisk are considered “core” values and should be used for population of a list view to be presented to the end user.

Element Name	Data Type	Description	Scope
ModuleID	string	Numeric module reference ID.	All
*ModuleName	string	Name of module, also type of object to be approved.	All
ID	string	Globally unique database-level identifier of object. This is the ID used for the drill down call. Not visible to end users.	All
*ref	string	Object ID, visible to end users.	All
*uom	string	3-character currency code.	All
*name	string	Name of object as defined by user.	All
*amount	string	Amount to be approved.	All
refVersion	numeric	Revision number of object to be approved.	All
iconType	string	Description of object type: worker, procurement, payment.	All
validObject	boolean	Always returns "false".	All
Status	string	Ok if response is OK.	All
Details	string	All ok if response is OK.	All
startDate	Date (mm/dd/yyyy/)	Recorded start date of object to be approved. Logical value is module dependent.	Module dependent

#### Sample API call

```
GET /api/v1/approvals/
authorization: Bearer WDXlKj3TTO3rpg9GHnZpbKmvj1=
x-ApplicationKey: 5c91f4fdb0c6ee9992ff476f89bf6cf25e589350
content-type: application/x-www-form-urlencoded
```

#### Sample Response

```
{
  "HEADER": {
    "AppVersion": "FieldglassAPIServer/1.0",
    "OSUAStr": "FieldglassAPIServer/1.0",
    "NumRecs": 1,
    "Status": "ok",
    "Details": "all ok"
  },
  "PAYLOAD": [
    {
      "ModuleID": "40",
```

```
"ModuleName": "Job Posting",
"ID": "z16042611572049751234567",
"Attributes": {
  "ref": "ABCDJP00000001",
  "uom": "USD",
  "name": "API developer",
  "startDate": "05/01/2016",
  "amount": "100,000.00",
  "refVersion": 0,
  "iconType": "procurement",
  "validObject": false,
  "createDate": "04/01/2016",
  "creatorName": "User ABCD",
  "uniqueName": "user_abcd@test.com",
  "emailId": "user_abcd@test.com",
  "status": "Pending Approval"
},
"Status": "ok",
"Details": "all ok"
}
```

```
]
```

## Retrieve All Work Items for Approval by Module

The following request statement returns details for work items requiring approval action, filtered down to a specific module:

```
GET /api/v1/approvals/module_{moduleid}
```

Parameter Name	Parameter Type	Data Type	Description
moduleID	moduleID	string	Module ID for drill-down purposes presented in the work_item call.

## Response Body Elements

The following table describes the data elements contained in the response. In addition:

- Many other elements will be provided depending on the type of object. For example, job posting will return additional creator and owner details, along with start/end dates, job description, and so on. The additional details are presented in the “others” section following the core attributes.
- Fields marked with an asterisk are considered core values and should be used for population of a list view to be presented to the end user. The rest of the data should be considered extrinsics/extension elements and can be shown to the end user per the customer’s business requirements.

Element Name	Data Type	Description	Scope
ModuleID	string	Numeric module reference ID.	All
*ModuleName	string	Name of module, also type of object to be approved.	All
ID	string	Globally unique database-level identifier of object. This is the ID used for the drill down call. Not visible to end users.	All
*ref	string	Object ID, visible to end users.	All
*uom	string	3-character currency code.	All
*name	string	Name of object as defined by user.	All
*amount	string	Amount to be approved.	All
refVersion	numeric	Revision number of object to be approved.	All
iconType	string	Description of object type: worker, procurement, payment.	All
validObject	boolean	Always returns “false”.	All
Status	string	Ok if response is OK.	All



Element Name	Data Type	Description	Scope
Details	string	All ok if response is OK.	All
startDate	Date (mm/dd/yyyy/)	Recorded start date of object to be approved. Logical value is module dependent.	Module dependent
others	various	Additional data captured in SAP Fieldglass for the object. Full list of data is module and configuration-dependent, and should be consumed/presented as extrinsics.	Module dependent

#### Sample API call

```
GET /api/v1/approvals/module_40
authorization: Bearer WDXlKj3TTOOn3rpg9GHnZpbKmvj1=
x-ApplicationKey: 5c91f4fdb0c6ee9992ff476f89bf6cf25e589350
content-type: application/x-www-form-urlencoded
```

#### Sample Response

```
{
  "HEADER": {
    "AppVersion": "FieldglassAPIServer/1.0",
    "OSUAstr": "FieldglassAPIServer/1.0",
    "NumRecs": 1,
    "Status": "ok",
    "Details": "all ok"
  },
  "PAYLOAD": [
    {
      "ModuleID": "40",
      "ModuleName": "Job Posting",
      "ID": "z16042611572049752170918",
      "Attributes": {
        "ref": "XXXXJP00037008",
        "uom": "INR",
        "name": "QA-Engineer-Software Test I(India Monthly)",
        "startDate": "01/05/2016",
        "amount": "685,243.65",
        "others": {
          "[GROUP_ID]label.details": "Details",
          "Job Posting Number of Positions": "1",
          "Job Posting Start Date": "01/05/2016",
          "Description": "Description",
          "Primary Work Site": "Test",
          "Maximum Submissions Per Supplier": "3",
          "Buyer": "XXXX",
          "Cost Center List": "Cost Center 1",
          "Job Posting Title": "Test Engineer",
          "Job Posting End Date": "31/07/2016",
          "Job Posting Distribution Level": "1",
          "Job Posting Duration (Days)": "65",
          "Primary Location": "Test",
          "Job Posting Service Type": "Temp",
          "Logged User Name": "Test, John",

```

```

    "[GROUP_ID]label.accounting": "Accounting",
    "Rates": "Calculated Rate",
    "Job Posting Average Rate": "228,414.55 INR",
    "Cost Allocation %": "Cost Center 1 100.000%",
    "Estimated Spend": "685,243.65 INR",
    "Target Bill Rate": "228,414.55",
    "Prior PO Number": "N/A",
    "Prior PO Amount: (if not applicable, enter zero).": "0.00",
    "Prior FG Transaction ID": "N/A",
    "Is this transaction a Renewal?": "(No Value)",
    "Rebate Exception": "No",
    "Exception Required": "(No Value)",
    "Is this worker from a reclassified SOW?": "(No Value)",
    "Previous SOW Transaction ID": "(No Value)",
    "GBW escalation required?": "(No Value)",
    "[GROUP_ID]label.postingInformation": "Posting Information",
    "Job Posting Creator": "John Test",
    "Job Posting Owner": "John Test",
    "Reason for Job Posting/Use of Temporary Worker": "Increased or peak workload
requires additional resourcing",
    "Will an interview be required?": "No",
    "Work Hours": "8",
    "Job Posting Category": "TEMPORARY PERSONNEL",
    "Job Title": "Test Title"
  },
  "refVersion": 0,
  "refNameKey": "label.jobPostingId",
  "refName": "Job Posting ID",
  "iconType": "procurement",
  "validObject": true,
  "createDate": "26/04/2016",
  "creatorName": "John Test",
  "uniqueName": "asrajash@cisco.com",
  "emailId": "fg_qatest8@sap.com",
  "status": "Pending Approval"
},
"Status": "ok",
"Details": "all ok"
}
]
}

```

## Retrieve Details for Specific Work Item Requiring Approval

The following request statement returns details for a specific work item requiring approval action:

```
GET /api/v1/approvals/module_{moduleid}/{workItemId}
```

Parameter Name	Parameter Type	Data Type	Description
moduleid	moduleID	string	Module ID for drill-down purposes presented in the work_item call.
workItemId	objectID	string	Specific object ID for drill-down purposes presented in the work_item call.

## Response Body Elements

The response will contain the following data elements. In addition:

- Many additional elements will be provided depending on the type of object. For example, job posting will return additional creator and owner details, along with start/end dates, job description, and so on. The additional details are presented in the “others” section following the core attributes.
- Fields marked with an asterisk are considered “core” values and should be used for population of a list view to be presented to the end user. The rest of the data should be considered extrinsics/extension elements and can be shown to the end user per the customer’s business requirements.

Element Name	Data Type	Description	Scope
ModuleID	string	Numeric module reference ID.	All
*ModuleName	string	Name of module, also type of object to be approved.	All
ID	string	Globally unique database-level identifier of object. This is the ID used for the drill down call. Not visible to end users.	All
*ref	string	Object ID, visible to end users.	All
*uom	string	3-character currency code.	All
*name	string	Name of object as defined by user.	All
*amount	string	Amount to be approved.	All
refVersion	numeric	Revision number of object to be approved.	All
iconType	string	Description of object type: worker, procurement, payment.	All
validObject	boolean	Always returns “false”.	All
Status	string	Ok if response is OK.	All
Details	string	All ok if response is OK.	All
startDate	Date (mm/dd/yyyy)	Recorded start date of object to be approved. Logical value is module dependent.	Module dependent
others	various	Additional data captured in SAP Fieldglass for the object. Full list of data is module and configuration-dependent and should be consumed/presented as extrinsics.	Module dependent

### Sample API call

```
GET /api/v1/approvals/module_40/z16052019464862442387902
```

```
authorization: Bearer WDXlKj3TTOOn3rpg9GHnZpbKmvjl=
x-ApplicationKey: 5c91f4fdb0c6ee9992ff476f89bf6cf25e589350
content-type: application/x-www-form-urlencoded
```

## Sample Response

```
{
  "HEADER": {
    "AppVersion": "FieldglassAPIServer/1.0",
    "OSUAStr": "FieldglassAPIServer/1.0",
    "NumRecs": 1,
    "Status": "ok",
    "Details": "all ok"
  },
  "PAYLOAD": [
    {
      "ModuleID": "40",
      "ModuleName": "Job Posting",
      "ID": "z16042611572049752170918",
      "Attributes": {
        "ref": "XXXXJP00037008",
        "uom": "INR",
        "name": "Test Title",
        "startDate": "01/05/2016",
        "amount": "685,243.65",
        "others": {
          "[GROUP_ID]label.details": "Details",
          "Job Posting Number of Positions": "1",
          "Job Posting Start Date": "01/05/2016",
          "Description": "Job Posting Description",
          "Primary Work Site": "Chicago",
          "Maximum Submissions Per Supplier": "3",
          "Buyer": "XXXX",
          "Cost Center List": "Test Cost Center",
          "Job Posting Title": "Test Title",
          "Job Posting End Date": "31/07/2016",
          "Job Posting Distribution Level": "1",
          "Job Posting Duration (Days)": "65",
          "Primary Location": "Test",
          "Job Posting Service Type": "Temp",
          "Logged User Name": "Test",
          "[GROUP_ID]label.accounting": "Accounting",
          "Rates": "ST/ HR INR",
          "Job Posting Average Rate": "228,414.55 INR",
          "Cost Allocation %": "Cost Center 1 100.000%",
          "Estimated Spend": "685,243.65 INR",
          "Target Bill Rate": "228,414.55",
          "Is this transaction a Renewal?": "(No Value)",
          "Exception Required": "(No Value)",
          "Job Posting Creator": "John Test",
          "Job Posting Owner": "John Test",
          "Reason for Job Posting/Use of Temporary Worker": "Increased or peak workload requires additional resourcing",
          "Will an interview be required?": "No",
          "Work Hours": "8",
          "Job Posting Category": "TEMPORARY PERSONNEL",
```

### Sample Response

```
{
  "HRMS Job Title": "Testing Job Title"
},
{
  "refVersion": 0,
  "refNameKey": "label.jobPostingId",
  "refName": "Job Posting ID",
  "iconType": "procurement",
  "validObject": true,
  "createDate": "26/04/2016",
  "creatorName": "John Test",
  "uniqueName": "qa_test8@sap.com",
  "emailId": "fg_qatest8@sap.com",
  "status": "Pending Approval"
},
{
  "Status": "ok",
  "Details": "all ok"
}
]
```

## Approve Work Item

The following method posts decision to approve work item to SAP Fieldglass:

```
POST /api/v1/approvals/module_{moduleid}/{workItemId}/action/approve
```

Parameter Name	Parameter Type	Data Type	Description
moduleID	moduleID	string	Numeric module reference ID from the object to be approved.
workItemID	Object ID	string	Specific object ID for drill-down purposes presented in the work_item call.

## Approval Post Body Elements

Element Name	Data Type	Description	Scope
ModuleID	string	Numeric module reference ID from the object to be approved.	All
Comments	string	User-entered comments.	All

### Sample Approval Post

```
POST /api/v1/approvals/module_40/z012704071613232110373902/action/approve
authorization: Bearer WDXlKj3TTOn3rpg9GHnZpbKmvjl=
x-ApplicationKey: 5c91f4fdb0c6ee9992ff476f89bf6cf25e589350
content-type: application/x-www-form-urlencoded

comments=Item looks good, I approve.
```

## Get Rejection Reasons

The following request statement returns all configured work item rejection reasons by module.

```
GET /api/v1/approvals/reject_reasons/module_{moduleId}
```

## Response Body Elements

The response will contain the following data elements. :

- Note that rejection reasons configured in SAP Fieldglass are specific to the module that the item being acted upon falls under. Even if the text description of the rejection reason across modules is the same, the ID returned on the rejection post will be module specific.

Element Name	Data Type	Description	Scope
ModuleID	string	Numeric module reference ID.	All
*ModuleName	string	Name of module.	All
ID	string	Globally unique database-level identifier of reject reason. This is the ID that must be returned in the subsequent rejection post.	All
attributes/others/description	various	User-visible rejection reason.	All

## Sample API call

```
GET /api/v1/approvals/reject_reasons/module_40
authorization: Bearer WDXlKj3TTON3rpg9GHnZpbKmvjl=
x-ApplicationKey: 5c91f4fdb0c6ee9992ff476f89bf6cf25e589350
content-type: application/x-www-form-urlencoded
```

## Sample Response

```
{
  "HEADER": {
    "AppVersion": "1.0",
    "OSUAStr": "FieldglassAPIServer/1.0",
    "NumRecs": 2,
    "Status": "ok",
    "Details": "all ok"
  },
  "PAYLOAD": [
    {
      "ModuleID": "40",
      "ModuleName": "Job Posting",
      "ID": "z14090820350427782472901",
      "Attributes": {
        "others": {
          "description": "Full Job Description Required"
        },
        "refVersion": 0,
        "validObject": false
      },
      "Status": "ok",
      "Details": "pass"
    },
    {
      "ModuleID": "730",
      "ModuleName": "Fee",
      "ID": "z12061115370330719460a6d",
      "Attributes": {
        "others": {
          "description": "Claim in Dispute"
        },
        "refVersion": 0,
        "validObject": false
      },
      "Status": "ok",
      "Details": "pass"
    }
  ]
}
```

## Reject Work Item

The following method posts decision to reject the work item to SAP Fieldglass:

```
POST /api/v1/approvals/module_{moduleId}/{workItemId}/action/reject
```

Parameter Name	Parameter Type	Data Type	Description
moduleId	moduleId	string	Numeric module reference ID from the object to be approved.
workItemId	Object ID	string	Specific object ID for drill-down purposes presented in the work_item call.

## Rejection Post Body Elements

The body of the approval post contains the module ID reference as well as end user-entered optional comments.

Element Name	Data Type	Description	Scope
ModuleID	string	Numeric module reference ID from the object to be approved.	All
Comments	string	User-entered comments.	All
Rejected	string	Globally unique ID of the rejection reason selected by the user.	All

### Sample Rejection Post

```
POST /api/v1/approvals/module_40/z16052021440472596879916/action/reject
authorization: Bearer WDXlKj3TTOOn3rpg9GHnZpbKmvjl=
x-ApplicationKey: 5c91f4fdb0c6ee9992ff476f89bf6cf25e589350
content-type: application/x-www-form-urlencoded
comments=no good, I reject this&rejectId=z14090820350427782472901
```



## HTTP Codes

### Successful Requests

The web services return a HTTP 2xx response code when the request was successful.

#### HTTP Success Codes

Success Code	Message	Description	Scope
200	OK	The request was received successfully.	All

### Failed Requests

The web service returns a 4xx or 5xx HTTP response code if the request had any errors.

#### HTTP Error Codes

Success Code	Message	Description	Scope
400	Bad Request	This response is returned if any of the following conditions are true:  * The specified URL is invalid. * The request is not formatted correctly. * The request is missing a required field.	All
401	Unauthorized	User/password or application key is not verified.	Token request
403	Forbidden	Token is not verified.	Resource Request
404	Not Found	The service being requested does not exist.	All
500	Internal Server Error	Server not available.	All

## Appendix A: Module ID Reference

The following modules are supported for the work item API:

Module ID	Module Code	Module Name	Mobile Approval Supported
40	JP	Job Posting	Yes
70	TS	Timesheet	Yes
143		Prequalify Workforce	No
180	IN	Invoice	Yes
270	WO	Work Order	Yes
271		Work Order Revision	Yes
272		SOW Worker	No
370	CI	Consolidated Invoice	Yes
411	MI	Miscellaneous Invoice	Yes
412	CD	Credit/Debit Memo	Yes
430	ES	Expense Sheet	Yes
540	TQ	SOW	Yes
550		SOW Revision Response	Yes
560	PC	SOW Credit/Debit Memo	Yes
570	PI	SOW Invoice	Yes
590	PE	Event	Yes
600	TR	SOW Response	Yes
610		SOW Revision Response	Yes
640	PS	Schedule	Yes
730	TU	Fee	Yes
890	RX	RFX	Yes
1190	PJ	Project	No
1690	PA	Decision Form	No
1900	SQ	Supplier Qualification	No

Module ID	Module Code	Module Name	Mobile Approval Supported
2100	PW	Profile Worker	No
3080		Work Order Change Request	No

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. Please see <http://www.sap.com/corporate-en/legal/copyright/index.epx#trademark> for additional trademark information and notices. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors.

National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP SE or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP SE or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice.

The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, which speak only as of their dates, and they should not be relied upon in making purchasing decisions.