

PUBLIC
2023-06-20

Developer Guide for SAP Digital Vehicle Hub

© 2023 SAP SE or an SAP affiliate company. All rights reserved.

THE BEST RUN SAP

Content

1 Overview. 3

2 Introduction. 4

3 Accessing APIs 8

3.1 Creating a Service Instance. 8

3.2 Create a Service Key. 9

3.3 Authentication. 10

4 API Information. 13

4.1 REST APIs. 13

Create Vehicle Service. 13

Update Vehicle Service. 14

Read Vehicle Service. 15

Search Vehicle Service. 17

Delete Vehicle Service. 20

4.2 OData APIs. 20

1 Overview

This development guide complements the APIs of SAP Digital Vehicle Hub, published in the SAP Business Accelerator Hub for SAP Digital Vehicle Hub. It provides usage details, and convenient examples for developers who wish to consume these APIs.

2 Introduction

SAP Digital Vehicle Hub offers the following services for a vehicle. These services will be published as APIs for external consumption.

The APIs are categorized into:

1. REST APIs
   REST APIs are used for maintaining vehicles. For example, to create, update, read and search complete vehicle information. It also includes both configuration and transactional information.
   These APIs are intended to be used in A2A scenarios where vehicle information needs to replicated in SAP Digital Vehicle Hub from different applications, for example, from SAP S/4HANA Vehicle Management System.For more information, see here.

REST API	COMMENT
Create Vehicle	To create a vehicle in SAP Digital Vehicle Hub
Update Vehicle	To update a vehicle in SAP Digital Vehicle Hub
Read Vehicle	To get details of a vehicle in SAP Digital Vehicle Hub
Search Vehicle	To get list of vehicles in SAP Digital Vehicle Hub, based on a specified search criterion.
Delete Vehicle	To delete a vehicle in SAP Digital Vehicle Hub

2. OData APIs
   OData APIs are fine granular APIs used to to maintain vehicle associations such as transactional information of a vehicle.
   These APIs are intended to be used in A2X scenarios where transactional information of a vehicle must be maintained directly from a UI. For example, a process UI to maintain the service information of a vehicle.
   For more information, see here.

OData API	COMMENT
Create Procurement	To create procurement entry for a vehicle in SAP Digital Vehicle Hub
Update Procurement	To update procurement entry for a vehicle in SAP Digital Vehicle Hub
Get Procurement	To get procurement entries for a vehicle in SAP Digital Vehicle Hub
Create Sales	To create sales entry for a vehicle in SAP Digital Vehicle Hub
Update Sales	To update sales entry for a vehicle in SAP Digital Vehicle Hub
Get Sales	To get sales entries for a vehicle in SAP Digital Vehicle Hub

OData API	COMMENT
Create Usage	To create usage entry for a vehicle in SAP Digital Vehicle Hub
Update Usage	To update usage entry for a vehicle in SAP Digital Vehicle Hub
Get Usage	To get usage entries for a vehicle in SAP Digital Vehicle Hub
Create Insurance Contract	To create insurance contract entry for a vehicle in SAP Digital Vehicle Hub
Update Insurance Contract	To update insurance contract entry for a vehicle in SAP Digital Vehicle Hub
Get Insurance Contract	To get insurance contract entries for a vehicle in SAP Digital Vehicle Hub
Create Registration	To create registration entry for a vehicle in SAP Digital Vehicle Hub
Update Registration	To update registration entry for a vehicle in SAP Digital Vehicle Hub
Get Registration	To get registration entries for a vehicle in SAP Digital Vehicle Hub
Create Invoice	To create invoice entry for a vehicle in SAP Digital Vehicle Hub
Update Invoice	To update invoice entry for a vehicle in SAP Digital Vehicle Hub
Get Invoice	To get invoice entries for a vehicle in SAP Digital Vehicle Hub
Create Warranty	To create warranty entry for a vehicle in SAP Digital Vehicle Hub
Update Warranty	To update warranty entry for a vehicle in SAP Digital Vehicle Hub
Get Warranty	To get warranty entries for a vehicle in SAP Digital Vehicle Hub
Create Service Contract	To create service contract entry for a vehicle in SAP Digital Vehicle Hub
Update Service Contract	To update service contract entry for a vehicle in SAP Digital Vehicle Hub
Get Service Contract	To get service contract entries for a vehicle in SAP Digital Vehicle Hub
Create Service History	To create service history entry for a vehicle in SAP Digital Vehicle Hub
Update Service History	To update service history entry for a vehicle in SAP Digital Vehicle Hub
Get Service History	To get service history entries for a vehicle in SAP Digital Vehicle Hub

OData API	COMMENT
Create Planned Maintenance Interval	To create planned maintenance intreval entry for a vehicle in SAP Digital Vehicle Hub
Update Planned Maintenance Interval	To update planned maintenance intreval entry for a vehicle in SAP Digital Vehicle Hub
Get Planned Maintenance Interval	To get planned maintenance intreval entries for a vehicle in SAP Digital Vehicle Hub
Create Material Document	To create a material document entry for a vehicle in SAP Digital Vehicle Hub.
Update Material Document	To update a material document entry for a vehicle in SAP Digital Vehicle Hub.
Get Material Document	To get a material document entries for a vehicle in SAP Digital Vehicle Hub.
Create Deliveries	To create an entry for deliveries for a vehicle in SAP Digital Vehicle Hub.
Update Deliveries	To update an entry for deliveries for a vehicle in SAP Digital Vehicle Hub.
Get Deliveries	To get delivery entries for a vehicle in SAP Digital Vehicle Hub.
Create Related Procurement	To create a related procurement entry for a vehicle in SAP Digital Vehicle Hub.
Update Related Procurement	To update a related procurement entry for a vehicle in SAP Digital Vehicle Hub.
Get Related Procurement	To get related procurement entries for a vehicle in SAP Digital Vehicle Hub.
Create Related Object	To create a related object for a vehicle in SAP Digital Vehicle Hub
Update Related Object	To update a related object for a vehicle in SAP Digital Vehicle Hub
Get Related Object	To get a related object entries for a vehicle in SAP Digital Vehicle Hub
Create Financial Document	To create a financial document entry for a vehicle in SAP Digital Vehicle Hub.
Update Financial Document	To update a financial document entry for a vehicle in SAP Digital Vehicle Hub.
Get Financial Document	To get a financial document entry for a vehicle in SAP Digital Vehicle Hub.
Create Source System Status	To create a source system status entry for a vehicle in SAP Digital Vehicle Hub.
Update Source System Status	To update a source system status entry for a vehicle in SAP Digital Vehicle Hub.
Get Source System Status	To get a source system status entry for a vehicle in SAP Digital Vehicle Hub.

OData API	COMMENT
Create extension entity 1,extension entity 2,extension entity 3,extension entity 4,extension entity 5	To create an extension entity for a vehicle in SAP Digital Vehicle Hub.
Update extension entity 1,extension entity 2,extension entity 3,extension entity 4,extension entity 5	To update an extension entity for a vehicle in SAP Digital Vehicle Hub.
Get extension entity 1,extension entity 2,extension entity 3,extension entity 4,extension entity 5	To get an extension entity for a vehicle in SAP Digital Vehicle Hub

For more information regarding the APIs for SAP Digital Vehicle Hub, see SAP Business Accelerator Hub for SAP Digital Vehicle Hub.

3 Accessing APIs

Navigate to *Subaccount* > *Spaces* > *Service Instances (select relevant service instance)* > *Service Keys* and select the relevant service key to find the application URL from the service key instance.

3.1 Creating a Service Instance

Create spaces in your Cloud Foundry organization using the SAP Business Technology Platform cockpit.

Prerequisites

You have a Cloud Foundry organization in your Cloud Foundry subaccount, and you have the *Org Manager* role in that organization. For more information, see Create Orgs.

Procedure

1. In the subaccount where you have subscribed to the application SAP Digital Vehicle Hub, choose *Spaces* from the navigation area.
2. 2. Open the space in which you want to create a service instance. If you do not have a space yet, create one as described in Create Spaces.
3. In the navigation area, choose *Service Marketplace*.
4. Choose the SAP Digital Vehicle Hub tile.
5. Choose *Instances* from the navigation pane.
6. Choose *New Instance*.
7. Choose the service plan *default* or *authorization-controlled* from the dropdown list.
8. Choose *Next*.
9. Specify a JSON file or specify parameters in the JSON format, then choose *Next*. This step is optional.
10. If you have already deployed an application that you want to bind to the new service instance, select it from the list. Choose *Next*. This step is optional.
11. Enter a name for your instance and choose *Finish*.

3.2 Create a Service Key

To generate the credentials, you must communicate directly with the service instance. To do this, you need to create a service key.

Prerequisite

Create a service instance. For more information, see Creating Service Instances.

Procedure

1. Navigate to the space in which you have created a service instance for which you want to create a service key.
2. Choose *Service Marketplace* from the navigation pane.
3. Choose the SAP Digital Vehicle Hub tile.
4. 15. Choose *Instances* from the navigation pane and select the instance you created previously.
5. 16. Choose *Service Keys* from the navigation pane
6. Choose *Create Service Key*.
7. Enter a name for the service key. You do not need to enter configuration parameters.
8. Save your changes.

Results

After creating the service key, you can view the parameters of the key.

Client Credentials	Parameters
Client ID	uaa.clientid
Client Secret	uaa.clientsecret
Token URL	uaa.url + '/oauth/token'
Application URLs	REST APIs: endpoints.service-url OData APIs: endpoints.odata-service-url

3.3 Authentication

All API endpoints are secured via the OAuth 2.0 Authorization Framework. The bearer/access tokens are used to grant access to the APIs. The bearer token is sent with each request in the authorization header parameter. The name bearer precedes the authorization token that is sent.

Example

authorization: Bearer ezyJh...

The token is granted from the tenant authorization using the client credential or password flow. The type flow is controlled by specifying the header parameter *grant_type* and the value *client_credentials* or password.

The caller needs the granting authority URL, the client ID, and the client secret to retrieve a token.

The response contains the access token (JWT) in the body as *access_token*. The validity of the parameter *expires_in* is limited. After the validity expires, a new token must be retrieved.

> **i Note**
>
> The password grant flow is supported if you are using the SAP ID Service to maintain the list of named users.

Get an Application Access Token

Use the credentials from the service key you created and a REST client of your choice to retrieve a JWT token.

You can obtain this token with the following HTTP request:

- Navigate to the service key that you created either while binding the service instance to your application or creating a service key.
- Make a POST request to fetch the access token using the parameters in the service key.

The parameters for the POST request are:

HTTP Method	POST
URL	<service key.uaa.url>/oauth/token For example, https://myidentityzone.authentication.eu10.hana.ondemand.com

HTTP Method	POST
Authorization	Basic authentication, <service key.uaa.clientid>and <service key.uaa.clientsecret> OR client_id and client_secret can be provided as url encoded form parameters.
Request Body	**For Client Credential Authentication** the client credentials (form-urlencoded) in the case of client credentials grant. For example, grant_type=client_credentials client_id=<service key.uaa.clientid> client_secret=<service key.uaa.clientsecret> response_type=token **For Password Authentication** client and user credentials (form-urlencoded in the case of password grant. For example, grant_type=password client_id=<service key.uaa.clientid> client_secret=<service key.uaa.clientsecret> username=<email ID> password=<password>
Response Code	200 OK

Response Body	Sample Code
	{
	"access_token":"2YotnFZFEjr1zCsicMWpAA",
	"token_type":"bearer",
	"expires_in":43199
	“scope”: “dvh……”
	"jti":"32ea018f008040c28e2d65aa5f141b05"
	}

Example

The following example shows the credentials needed in a service binding for a hosted service:



After you send the request, you receive a JSON response containing the access token. The following is a sample response to a request for the service:

Sample Code

```
{
    "access_token": "<long string>",
    "token_type": "bearer",
    "expires_in": "<time it expired in>",
    "scope": "uaa.*********",
    "jti": "12d345678a9b01c2af345aaedef67890"
}
```

⚠ Caution

The access token grants service access to anyone who is aware of it. Publishing this information on your side would constitute a considerable security risk.

4 API Information

All service endpoints will follow this convention: **`https://<application_url>/<service_endpoint_suffix>`**. For application URL, refer to Create a Service Key [page 9].

Header	Required	Description
Authorization	Yes	Bearer access token
content-type	Yes, for POST/PUT/ PATCH methods	application/json

The following chapters provide details of the REST and OData APIs.

4.1 REST APIs

REST APIs are used for maintaining vehicles. For example, to create, update, read and search complete vehicle information. It also includes both configuration and transactional information.

These APIs are intended to be used in A2A scenarios where vehicle information needs to replicated in SAP Digital Vehicle Hub from different applications, for example, from SAP S/4HANA Vehicle Management System.For more information, see here.

4.1.1 Create Vehicle Service

Details	Value
service_endpoint_suffix	/dvh/api/v1/vehicle
HTTP Method	POST
Behavior/Mode	Synchronous/Single
Request	Refer to SAP Business Accelerator Hub for SAP Digital Vehicle Hub for examples on request payload. See here.

Details	Value
Successful Execution	HTTP 201, Created Response: Sample Code `{ "vehicleIdentifyingElements": { "externalID": "test-vehicle", "id": "9AC2805C3AF340F48FB4CC2FA5CB2A72", "vin": "12345678901234567" } }`
Erroneous Execution	HTTP 400, Bad Request. For example, malformed request payload or error during data processing/data validation HTTP 401, 403. When the user is not authorized and forbidden from accessing the API HTTP 500, Internal Server Error

4.1.2 Update Vehicle Service

Details	Value
service_endpoint_suffix	/dvh/api/v1/vehicle
HTTP Method	PATCH
Behavior/Mode	Synchronous/Single
Query Parameter	Either ID or external ID or VIN
Request	Refer to SAP Business Accelerator Hub for SAP Digital Vehicle Hub for examples on request payload. See here.
Successful Execution	HTTP 204, No Content
Erroneous Execution	HTTP 404 Not Found. For example, when specified vehicle ID does not exist. HTTP 400, Bad Request. For example, malformed request payload or error during data processing/data validation. HTTP 401, 403. When the user is not authorized and forbidden from accessing the API. HTTP 500, Internal Server Error

4.1.3 Read Vehicle Service

Details	Value
service_endpoint_suffix	/dvh/api/v1/vehicle
HTTP Method	GET
HTTP Header	Accept-Language: <language>. For example, Accept-Language: en
Behavior/Mode	Synchronous/Single

Details	Value
Query Parameter	Either externalID or ID or VIN, modelSpecificationRequiredIndicator, vehicleSpecificationRequiredIndicator, componentRequiredIndicator, locationRequiredIndicator, documentRequiredIndicator, insuranceContractRequiredIndicator, invoiceRequiredIndicator, plannedMaintenanceIntervalRequiredIndicator, procurementRequiredIndicator, registrationRequiredIndicator, salesRequiredIndicator, serviceContractRequiredIndicator, serviceHistoryRequiredIndicator, usageRequiredIndicator, warrantyRequiredIndicator, sourceSystemStatusRequiredIndicator relatedProcurementRequiredIndicator materialDocumentRequiredIndicator deliveriesRequiredIndicator relatedObjectRequiredIndicator financialDocumentRequiredIndicator financialDocumentRequiredIndicator extensionEntity1RequiredIndicator extensionEntity2RequiredIndicator extensionEntity3RequiredIndicator extensionEntity4RequiredIndicator extensionEntity5RequiredIndicator sparePartRequiredIndicator

Details	Value
	Note The **true/false** indicator represents **true** if the corresponding details are returned in the response, and **false** if the details are not returned in the response.
Successful Execution	HTTP 200, with vehicle information in the response Refer to SAP Business Accelerator Hub for SAP Digital Vehicle Hub for examples on response. See here.
Erroneous Execution	HTTP 404 Not Found. For example, when the specified vehicle ID does not exist. HTTP 400, Bad Request. For example, malformed request payload or error during data processing/data validation HTTP 500, Internal Server Error

4.1.4 Search Vehicle Service

Details	Value
service_endpoint_suffix	/dvh/api/v1/vehicle/search
HTTP Method	PUT
Behavior/Mode	Synchronous/Bulk
Request	Refer to SAP Business Accelerator Hub for SAP Digital Vehicle Hub for examples on payload request. See here.
Successful Execution	HTTP 200, with list of vehicles Refer to SAP Business Accelerator Hub for SAP Digital Vehicle Hub for examples on response. See here.
No Result in Response	HTTP 200, with no vehicles HTTP 500, Internal Server Error

Constructing Search Request

Sample Code

```
"registration": {
            "licensePlateNumber": "KA"
```

```
    },
    "modelExternalID": [
        "9FD6075A0B5740538E350755002C91FE",
        "208E870E7E8448EABAB45F21FDA63DEF"
    ],
    "attribute": [
        {
                "id": "DVH_INTERIOR_COLOR",
                "operator": "IN",
                "value": ["Black","Red"]
        },
        {
                "id": "DVH_ODOMETER_READING",
                "operator": "IN",
                "value": ["4235"]
        }
    ]
```

Basic Filter Structure

Sample Code

```
{
"id": "DVH_INTERIOR_COLOR",
"operator": "IN",
"value": ["Black","Red"]
},
"registration": {
"licensePlateNumber": ["KA","DL"]
}
```

Filter conditions support the following for type attributes, modelAttributes and filter of type date and number:

Operators

- “EQ" - Equals
- "GT" - Greater than
- "GE" - Greater than or Equal to
- "LT" - Less than
- "LE" - Less than or Equal to
- "IN" - Present in a list/array
- “BT” - Between a range (range is defined by setting value1 & value2 fields)

For string fields without code lists, such as licensePlateNumber, the operator is fixed to “IN” with fuzzy searching.

For string fields with code lists, such as type code in any entity, the operator is fixed to “IN” without fuzzy searching.

Combining multiple filters

Multiple or nested filters are formed when more than one filter is defined in a level connected by a logical operation (Logical AND).

Example

The licensePlateNumber (KA) and modelExternalID (9FD6075A0B5740538E350755002C91FE) are connected by AND operation.

Sample Code

```
"registration": {
"licensePlateNumber": ["KA"]
},
"modelExternalID": ["9FD6075A0B5740538E350755002C91FE"]
```

Pagination

Pagination can be achieved by simultaneously setting the top and skip request options with every request.

> **Note**
>
> The default value of top and skip will be 20 and 0.
>
> The supported maximum value of top + skip is 1000.

Pagination using SearchAfter

For pagination beyond 1000 objects, use *searchAfter*.

This value will be part of every response and must be sent in the subsequent call to get the next batch of records.

> **Note**
>
> The skip parameter will not be applied if searchAfter is being used.

Sample Code

```
{
"top": 20,
"skip": 0,
"attributes": [],
“searchAfter”: “”
}
```

Sample Code

```
{
“count”: 10,
“value”: [],
“searchAfter”: “4BC02F4450CE4321B3112937346944C2”
}
```

Limitations

1. Filter on attributes of type *picture*, *enum*, attributes with code lists of type *date*, numeric, and numeric flexible with multiselect options along with unit of measure are not supported.
2. Multilanguage string attribute will not be displayed as per your logon language.
3. Search on attributes with a unit of measure is only supported when the actual value is maintained. The automatic value conversion to your preferred metric even when the value is maintained in a different metric is not supported. For example, if you have saved 10 Kilograms as a value and try to search for 22 Pounds, you will not get the results, even though the value is the same in different metrics.

4. The search results are *near real-time*, there might be a slight delay between when the changes are made and when they are available in the search results.
5. Partial text search on string attributes is not supported for attributes in *modelAttributes* and *attributes* section.
6. Case insensitive search on string attributes is not supported for attributes in *modelAttributes* and *attributes* section.
7. Search on attributes belonging to both model or vehicle configuration is supported but the payload should only contain the attribute in only one of those sections.
8. If filters are provided for any of the transactional entities or for few header fields, then it should not match more than 5000 vehicles.
9. Multi value search is not supported for numeric and date fields.
10. If the API is used with top and skip, then top and skip cannot be more than 1000. To get more than 1000 vehicles, use *searchAfter*.
11. Sorting will only be supported on *changedOn* when using top and skip. Srting cannot be supported when using the *searchAfter* field.
12. The exported excel will not show the vehicles in sorted order.
13. In case of an invitee account, the shared vehicles will not be displayed in the View Vehicles and Modify Vehicles apps.

4.1.5 Delete Vehicle Service

Details	Value
service_endpoint_suffix	/dvh/api/v1/vehicle
HTTP Method	DELETE
Behavior/Mode	Synchronous/Single
Request	Refer to SAP Business Accelerator Hub documentation for SAP Digital Vehicle Hub for examples on request payload. See here.
Successful Execution	HTTP 204, No Content
Erroneous Execution	HTTP 400, Bad Request. For example, malformed request payload or error during data processing/data validation. HTTP 401, 403. When the user is not authorized and forbidden from accessing the API. HTTP 500, Internal Server Error.

4.2 OData APIs

OData APIs are fine granular APIs used to to maintain vehicle associations such as transactional information of a vehicle.

These APIs are intended to be used in A2X scenarios where transactional information of a vehicle must be maintained directly from a UI. For example, a process UI to maintain the service information of a vehicle. For more information, see here.

Create

Details	Value
service_endpoint_suffix	/odata/v4/vehicleUsage/usage
	/odata/v4/vehicleRegistration/registration
	/odata/v4/vehicleSales/sales
	/odata/v4/vehicleProcurement/procurement
	/odata/v4/vehicleWarranty/warranty
	/odata/v4/vehicleInvoice/invoice
	/odata/v4/vehicleServiceContract/serviceContract
	/odata/v4/vehicleServiceHistory/serviceHistory
	/odata/v4/vehiclePlannedMainteanceInterval/plannedMaintenanceInterval
	/odata/v4/vehicleInsuranceContract/insuranceContract
	/odata/v4/vehicleSourceSystemStatus/sourceSystemStatus
	/odata/v4/vehicleRelatedProcurement/relatedProcurement
	/odata/v4/vehicleMaterialDocument/materialDocument
	/odata/v4/vehicleDeliveries/deliveries
	/odata/v4/vehicleRelatedObject/relatedObject
	/odata/v4/vehicleFinancialDocument/financialDocument
	/odata/v4/vehicleExtensionEntity/extensionEntity1, /odata/v4/vehicleExtensionEntity/extensionEntity2, /odata/v4/vehicleExtensionEntity/extensionEntity3, /odata/v4/vehicleExtensionEntity/extensionEntity4, /odata/v4/vehicleExtensionEntity/extensionEntity5
	/odata/v4/vehicleSparePart/sparePart
HTTP Method	POST
Behavior/Mode	Synchronous/Single. Supports deep insert
Request	Refer to SAP Business Accelerator Hub for SAP Digital Vehicle Hub for examples on payload request.See here.
Successful Execution	HTTP 200, with usage information in the response. Refer to SAP Business Accelerator Hub for SAP Digital Vehicle Hub for examples on response payload. See here.

Details	Value
Erroneous Execution	HTTP 400 Bad Request. For example, malformed request payload or error during data processing/data validation HTTP 401. User not authorized to access the API. HTTP 500 Internal Server Error.

Update

Details	Value
service_endpoint_suffix	/odata/v4/vehicleUsage/usage
	/odata/v4/registrationService/registration
	/odata/v4/vehicleSales/sales
	/odata/v4/vehicleProcurement/procurement
	/odata/v4/vehicleWarranty/warranty
	/odata/v4/vehicleInvoice/invoice
	/odata/v4/vehicleServiceContract/serviceContract
	/odata/v4/vehicleServiceHistory/serviceHistory
	/odata/v4/vehiclePlannedMainteanceInterval/plannedMaintenanceInterval
	/odata/v4/vehicleInsuranceContract/insuranceContract
	/odata/v4/vehicleSourceSystemStatus/sourceSystemStatus
	/odata/v4/vehicleRelatedProcurement/relatedProcurement
	/odata/v4/vehicleMaterialDocument/materialDocument
	/odata/v4/vehicleDeliveries/deliveries
	/odata/v4/vehicleRelatedObject/relatedObject
	/odata/v4/vehicleFinancialDocument/financialDocument
	/odata/v4/vehicleExtensionEntity/extensionEntity1, /odata/v4/vehicleExtensionEntity/extensionEntity2, /odata/v4/vehicleExtensionEntity/extensionEntity3, /odata/v4/vehicleExtensionEntity/extensionEntity4, /odata/v4/vehicleExtensionEntity/extensionEntity5
	/odata/v4/vehicleSparePart/sparePart
URL Key Parameter	insuranceContract: vehicleID, policyNumber

Details	Value
	relatedObject: vehicleID, number, typeCode, validFrom, sourceSystemStatus
	invoice: vehicleID, number, itemNumber
	plannedMaintenanceInterval: vehicleID, typeCode
	procurement: vehicleID, documentNumber, itemNumber
	registration: vehicleID, licensePlateNumber, startDate
	sales: vehicleID, documentNumber, itemNumber
	serviceContract: vehicleID, contractID
	serviceHistory: vehicleID, completionDate, serviceOrder-Number
	sourceSystemStatus: vehicleID, code, statusID, dateTime
	usage: vehicleID, contractNumber
	warranty: vehicleID, number
	relatedProcurement: vehicleID, documentNumber, docu-mentItemNumber
	materialDocument: vehicleID, number, itemNumber, year
	deliveries: vehicleID, documentNumber, documentItem-Number
	financialDocument: vehicleID, typeCode, number, itemNum-ber,companyCode, fiscalYear
	extensionEntities(1-5): vehicleID, documentNumber
	sparepart: vehicleID, partID, id
HTTP Method	PUT
Behavior/Mode	Synchronous/Single and supports deep update
Request	Refer to SAP Business Accelerator Hub for SAP Digital Vehicle Hub for examples on payload request. See here.
Successful Execution	HTTP 200, with usage information in the response. Refer to SAP Business Accelerator Hub for SAP Digital Vehicle Hub for examples on response payload. See here.
Erroneous Execution	HTTP 404 Not Found. When specified keys are invalid HTTP 400 Bad Request. For example, malformed request payload or error during data processing/data validation. HTTP 401. User not authorized to access the API. HTTP 500 Internal Server Error.

Get

Details	Value
service_endpoint_suffix	/odata/v4/vehicleUsage/usage
	/odata/v4/vehicleRegistration/registration
	/odata/v4/vehicleSales/sales
	/odata/v4/vehicleProcurement/procurement
	/odata/v4/vehicleWarranty/warranty
	/odata/v4/vehicleInvoice/invoice
	/odata/v4/vehicleServiceContract/serviceContract
	/odata/v4/vehicleServiceHistory/serviceHistory
	/odata/v4/vehiclePlannedMainteanceInterval/plannedMaintenanceInterval
	/odata/v4/vehicleInsuranceContract/insuranceContract
	/odata/v4/vehicleSourceSystemStatus/sourceSystemStatus
	/odata/v4/vehicleRelatedProcurement/relatedProcurement
	/odata/v4/vehicleMaterialDocument/materialDocument
	/odata/v4/vehicleDeliveries/deliveries
	/odata/v4/vehicleRelatedObject/relatedObject
	/odata/v4/vehicleFinancialDocument/financialDocument
	/odata/v4/vehicleExtensionEntity/extensionEntity1, /odata/v4/vehicleExtensionEntity/extensionEntity2, /odata/v4/vehicleExtensionEntity/extensionEntity3, /odata/v4/vehicleExtensionEntity/extensionEntity4, /odata/v4/vehicleExtensionEntity/extensionEntity5
	/odata/v4/vehicleSparePart/sparePart
Key Parameter URL for GET details for a specific entity	insuranceContract: vehicleID, policyNumber
	invoice: vehicleID, number, itemNumber
	plannedMaintenanceInterval: vehicleID, typeCode
	procurement: vehicleID, documentNumber, itemNumber
	registration: vehicleID, licensePlateNumber, startDate
	sales: vehicleID, documentNumber, itemNumber
	serviceContract: vehicleID, contractID
	serviceHistory: vehicleID, completionDate, serviceOrderNumber
	sourceSystemStatus: vehicleID, code, statusID, dateTime
	usage: vehicleID, contractNumber

Details	Value
	warranty: vehicleID, number
	relatedProcurement: vehicleID, documentNumber, documentItemNumber
	materialDocument: vehicleID, number, itemNumber, year
	deliveries: vehicleID, documentNumber, documentItemNumber
	relatedObject: vehicleID, number, typeCode, validFrom, sourceSystemStatus
	financialDocument: vehicleID, typeCode, number, itemNumber,companyCode, fiscalYear
	extensionEntities(1-5): vehicleID, documentNumber
	sparepart: vehicleID, partID, id
Expand entities	insuranceContract: document, attribute (value)
	invoice: document, attribute (value), businessTransactionDocumentReference
	plannedMaintenanceInterval: attribute (value)
	procurement: document, attribute (value), businessTransactionDocumentReference
	registration: document, attribute (value)
	sales: document, attribute (value), businessTransactionDocumentReference
	serviceContract: document, attribute (value), businessTransactionDocumentReference
	serviceHistory: document, attribute (value), part, businessTransactionDocumentReference
	usage: document, attribute (value), additionalDriver, cardProvider
	warranty: document, attribute (value)
	relatedProcurement: document, attribute (value), businessTransactionDocumentReference
	materialDocument: document, attribute (value), businessTransactionDocumentReference
	deliveries: document, attribute (value), businessTransactionDocumentReference
	relatedObject: document, attribute (value), businessTransactionDocumentReference
	financialDocument: document, attribute (value), businessTransactionDocumentReference
	extensionEntities(1-5): document, attribute (value), businessTransactionDocumentReference

Details	Value
	sparepart: document, attribute (value)
HTTP Method	GET
Behavior	Synchronous. Supports GET details for a specific instance of an entity Supports GET list of instances for an entity
Request	Refer to SAP Business Accelerator Hub for SAP Digital Vehicle Hub for examples on payload request. See here.
Successful Execution	HTTP 200, with usage information in the response. Refer to SAP Business Accelerator Hub for SAP Digital Vehicle Hub for examples on response payload. See here.
Erroneous Execution	HTTP 404 Not Found. When specified keys are invalid HTTP 400 Bad Request. For example, malformed request payload or error during data processing/data validation. HTTP 401. User not authorized to access the API. HTTP 500 Internal Server Error.

> **i Note**
>
> It is recommended that you use vehicleID as filter in sparePart ODATA GET API.

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.
About the icons:

- Links with the icon : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
  - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
  - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon : You are leaving the documentation for that particular SAP product or service and are entering an SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.
The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.

www.sap.com/contactsap

© 2023 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. The information contained herein may be changed without prior notice.

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 or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP 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.

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. All other product and service names mentioned are the trademarks of their respective companies.

Please see https://www.sap.com/about/legal/trademark.html for additional trademark information and notices.

THE BEST RUN SAP