Statement of Compliance

Västtrafik BoB-APIs

(Server-side)

BoB-Version: v1.3.0

Västtrafik Release, VT BoB-module: 1.1

Disclaimer: The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, expressively or implied.

**Revision History:**

|  |  |  |
| --- | --- | --- |
| Revision | Description | Author |
| PA12 | Version for Review | Mikael Fröberg, Mathias Johansson |
| PA13 | Added GET /ticket/{ticketId} and GET /ticketbundle/{ticketbundleid} as Supported operations. | Mikael Fröberg |
| PA14 | Corrections after review. Updated VT BoB-module to 1.1  In 1.1 support was added for operations:   * GET /ticket/{ticketId} * GET /ticketbundle/{ticketBundleId} |  |
| PA15 | POST /whitelist, corrected to GET /whitelist |  |
| PA16 | Added clarification on line information when making a request to the product API containing a route. |  |
| PA17 | Editorial Updates | Mathias Johansson |
| Rev. A |  | Mikael Fröberg |
| PB1 | RequestId is mandatory in POST /ticket | Mikael Fröberg |

[1. General 5](#_Toc33685574)

[1.1. Compliance 5](#_Toc33685575)

[1.2. Authentication and Authorization 6](#_Toc33685576)

[2. Authentication API 7](#_Toc33685577)

[3. Device API 8](#_Toc33685578)

[3.1. GET /device/kdk 8](#_Toc33685579)

[3.2. POST /device/key 8](#_Toc33685580)

[4. Inspection API 9](#_Toc33685581)

[4.1. POST /inspection/{ticketId} 9](#_Toc33685582)

[4.2. POST /inspection 9](#_Toc33685583)

[5. Product API 10](#_Toc33685584)

[5.1. GET /product 10](#_Toc33685585)

[5.2. POST /product 11](#_Toc33685586)

[5.3. GET /product/{productId} 12](#_Toc33685587)

[5.4. GET /productcat/fare 12](#_Toc33685588)

[5.5. GET /productcat/product 13](#_Toc33685589)

[5.6. GET /productcat/traveller 13](#_Toc33685590)

[5.7. POST /manifest 14](#_Toc33685591)

[5.8. GET /manifest/{manifestId} 15](#_Toc33685592)

[5.9. Response Definitions 16](#_Toc33685593)

[6. Ticket API 17](#_Toc33685594)

[6.1. GET /ticket 17](#_Toc33685595)

[6.2. POST /ticket 17](#_Toc33685596)

[6.3. GET /ticket/{ticketId} 17](#_Toc33685597)

[6.4. GET /ticket/{ticketId}/refundableStatus 19](#_Toc33685598)

[*6.5.* GET /ticket/{ticketId}/refundStatus 19](#_Toc33685599)

[*6.6.* PUT /ticket/{ticketId}/refundStatus 19](#_Toc33685600)

[*6.7.* GET /ticket/{ticketId}/recoverableStatus 20](#_Toc33685601)

[*6.8.* GET /ticket/{ticketId}/recoverStatus 20](#_Toc33685602)

[*6.9.* PUT /ticket/{ticketId}/recoverStatus 20](#_Toc33685603)

[*6.10.* GET /ticket/{ticketId}/hinderedStatus 20](#_Toc33685604)

[*6.11.* PUT /ticket/{ticketId}/hinderedStatus 20](#_Toc33685605)

[*6.12.* PUT /ticket/{ticketId}/revoke 21](#_Toc33685606)

[*6.13.* GET /ticket/{ticketId}/active 21](#_Toc33685607)

[*6.14.* PUT /ticket/{ticketId}/active 21](#_Toc33685608)

[*6.15.* GET /ticket/{ticketId}/event 21](#_Toc33685609)

[*6.16.* GET /ticket/{ticketId}/event/{eventId} 21](#_Toc33685610)

[*6.17.* POST /ticketbundle 22](#_Toc33685611)

[6.18. GET /ticketbundle/{ticketBundleId} 22](#_Toc33685612)

[7. Validation API 23](#_Toc33685613)

[7.1. POST /validation/{ticketId} 23](#_Toc33685614)

[7.2. POST /validation 23](#_Toc33685615)

[7.3. GET /whitelist 23](#_Toc33685616)

[7.4. GET /ticklemacros 23](#_Toc33685617)

[*7.5.* GET /ticklemacros/{serial} 23](#_Toc33685618)

[7.6. GET /blacklist 23](#_Toc33685619)

[7.7. GET /fraudcheck 24](#_Toc33685620)

[8. Traveller API 25](#_Toc33685621)

[9. Booking API 25](#_Toc33685622)

[10. Appendix – Examples 26](#_Toc33685623)

[10.1. Authentication API - JWT 26](#_Toc33685624)

[10.2. Product API – Route Property 26](#_Toc33685625)

­­­­

# General

## Compliance

One of the following compliance statements are made for each part of the BoB-specification:

* ”Supported”: An API, operation or attribute is supported.
* ”Partially Supported”: An API, operation or attribute is partially supported.
* ”Not Supported”: An API, operation or attribute is not supported.

As for error handling, the following applies:

* A missing Mandatory attribute will result in Error handling: HTTP Response Code 400 Bad Request
* An unknown attribute that is received will be ignored and the operation will continue.
* Error codes specified in the BoB API specifications are used, if not otherwise stated
* Error codes related to the above compliance statements are detailed in the compliance definitions below
* Error codes related to authentication, see 1.2 Authentication and Authorization
* Error codes related to authorization, see 1.2 Authentication and Authorization

### Supported

#### API level

A Supported API means all its operations and attributes are supported. Still business rules may apply to operations, see Authorization.

For a Supported API, compliance statements on operations and attributes are not needed and is therefore omitted.

#### Operation level

A Supported operation means all its request and response attributes are supported

For a Supported operation, compliance statements on attributes are not needed and is therefore omitted.

#### Attribute

A supported attribute means it is processed. Still restrictions may apply to the set of values supported – notes in footnotes.

For a Supported attribute, compliance statements on sub-attributes are not needed and are therefore omitted.

### Partially Supported

#### API level

A partially supported API means some of its operations and attributes are supported.

#### Operation level

A partially supported operation means some of its request and/or response attributes are supported.

#### Attribute

A partially supported attribute means that some of its sub-attributes are not supported.

### Not Supported

#### API level

A non-supported API means that none of its operations are supported.

#### Operation level

A non-supported operation means its request and response are not supported.

Error handling: HTTP Response Code 501 Not Implemented

#### Attribute

A non-supported attribute means that the operation will not be accepted, if not otherwise is specifically stated in this document.

Error handling: HTTP Response Code 400 Bad Request

## Authentication and Authorization

### Authentication

This section describes the authentication of JWTs supplied in the context of accessing an operation in Västtrafik´s BoB APIs. For the issuing of JWTs see chapter 2.

All communication with the Västtrafik BoB API must use TLS (Transport Layer Security) version 1.2 or above.

Authentication involves:

* Validating the JWT signature
  + Public authentication keys need to be available in the Samtrafiken Participant Metadata API
* Validating the BoB holder of key claim, bobHok. Holder of key claim is required to be present
* Validating expiration properties of the JWT

If any of the above validation fails, the result is a HTTP Response Code 401 Unauthorized.

### Authorization

BoB specification defines a set of roles. Västtrafik´s BoB implementation makes authorization decisions based on those roles.

In general, the “pos” role (Point of Sale) needs to be preceded by a commercial contract with Västtrafik, and set up internally internally to be able to handle accounting and invoicing. Not doing so results in a failed authorization.

For specific operations, the authorization requirements are defined per operation in the specific API chapters.

A failed authorization results in a HTTP Response Code 403 Forbidden.

# Authentication API

Supported[[1]](#footnote-1)

Notes on issued JWT:

* JWTs always include the claim holder-of-key,”bobHok”.
* Participants authenticating a JWT issued by Västtrafik are recommended to assert the holder-of-key.
* JWTs may include proprietary claims. Those are prefixed with ”vt-” and should be ignored.

For an example JWT, see chapter 10

# Device API

Supported

## GET /device/kdk

Supported

#### Authorization

bobAuthZ: [val, ins]

## POST /device/key

Supported

#### Authorization

Authorization policy: [anonymous]

# Inspection API

Partially Supported

## POST /inspection/{ticketId}

Partially Supported[[2]](#footnote-2) [[3]](#footnote-3)

#### Authorization

bobAuthZ: [ins]

## POST /inspection

Partially Supported[[4]](#footnote-4) [[5]](#footnote-5)

#### Authorization

bobAuthZ: [ins]

# Product API

Partially Supported

## GET /product

Partially Supported

#### Authorization

bobAuthZ: [pos,tvm]

### Request - Partially Supported

* Headers:
  + Accept-Language Not Supported[[6]](#footnote-6)
* Query parameters:
  + productCategoryId Supported
  + fareCategoryId Not Supported
  + travellerCategoryId Supported
  + originLocation Not Supported
  + destinationLocation Not Supported
  + productProperties Not Supported

### Response - Partially Supported

* Headers:
  + Content-Language Supported[[7]](#footnote-7)
  + Expires Supported
* Body:
  + productSetAlternatives Partially Supported
    - products Partially Supported
      * product Partially Supported[[8]](#footnote-8)

## POST /product

Partially Supported

#### Authorization

bobAuthZ: [pos,tvm]

### Request - Partially Supported:

* Headers:
  + Accept –Language Not Supported[[9]](#footnote-9)
* Body:
  + filter Partially Supported
    - route Partially Supported[[10]](#footnote-10)
      * coordinate Not Supported
      * groupId Supported
        + type Supported[[11]](#footnote-11)
    - group Supported[[12]](#footnote-12)
      * groupType Supported[[13]](#footnote-13)
    - area Not Supported
    - fareCategoryIds Not Supported
    - productCategoryIds Supported
    - travellersCategoryId Supported[[14]](#footnote-14)
    - travellersPerCategory Supported[[15]](#footnote-15)
    - productProperties Not Supported
    - adjacencies Not Supported
    - temporal Not Supported

### Response – Partially Supported:

* Headers:
  + Content-Language Supported[[16]](#footnote-16)
  + Expires Supported
* Body:
  + productSetAlternatives Partially Supported
    - products Partially Supported
      * product Partially Supported[[17]](#footnote-17)

## GET /product/{productId}

Partially Supported

#### Authorization

bobAuthZ: [pos,tvm]

### Request - Supported

### Response – Partially Supported

* Headers:
  + Content-Language Supported[[18]](#footnote-18)
  + Expires Supported
* Body:
  + product Partially Supported[[19]](#footnote-19)

## GET /productcat/fare

Partially Supported

#### Authorization

bobAuthZ: [pos]

### Request – Partially Supported

* Headers:
  + Accept-Language Not Supported[[20]](#footnote-20)

### Response - Supported

* Headers:
  + Content-Language Supported[[21]](#footnote-21)
* Body:
  + fareCategories Supported[[22]](#footnote-22)

## GET /productcat/product

Partially Supported

#### Authorization

bobAuthZ: [pos,tvm]

### Request – Partially Supported

* Headers:
  + Accept-Language Not Supported[[23]](#footnote-23)

### Response - Supported

* Headers:
  + Content-Language Supported[[24]](#footnote-24)
* Body:
  + productCategories Supported

## GET /productcat/traveller

Partially Supported

#### Authorization

bobAuthZ: [pos]

### Request – Partially Supported

* Headers:
  + Accept-Language Not Supported[[25]](#footnote-25)

### Response - Supported

* Headers:
  + Content-Language Supported[[26]](#footnote-26)
* Body:
  + travellerCategories Supported

## POST /manifest

Partially Supported

#### Authorization

bobAuthZ: [pos,tvm]

### Request – Partially Supported

* Headers:
  + Accept –Language Not Supported[[27]](#footnote-27)
* Body:
  + productManifestQuery Partially Supported
    - recoverTickets Not Supported
    - discount codes Not Supported
    - productSelections Partially Supported
      * productProperties Not Supported

### Response – Partially Supported:

* Headers:
  + Content-Language Supported[[28]](#footnote-28)
* Body
  + productSetManifest Partially Supported[[29]](#footnote-29)

## GET /manifest/{manifestId}

Partially Supported

#### Authorization

bobAuthZ: [pos,tvm]

### Request – Partially Supported

* Headers:
  + Accept –Language Not Supported[[30]](#footnote-30)
* Parameters:
  + manifestId Supported

### Response – Partially Supported:

* Headers:
  + Content-Language Supported[[31]](#footnote-31)
* Body
  + productSetManifest Partially Supported[[32]](#footnote-32)

## Response Definitions

### Product

* + productId Supported
  + productTitle Supported
  + productDescription Supported
  + validityPeriod Supported
  + productCategoryId Supported
  + fareCategoryId Not Supported
  + travellerCategoryId Not Supported
  + travellersPerCategory Supported
  + fares Supported
  + productExpire Supported
  + productProperties Not Supported
  + spatialValidity Partially Supported
    - routes Not Supported
    - areas Not Supported
    - groups Supported

### ProductSetManifest

* + manifestId Supported
  + productSetTitle Supported[[33]](#footnote-33)
  + productSetDescription Supported[[34]](#footnote-34)
  + manifest Supported
  + manifestExpire Supported
  + distinct Supported[[35]](#footnote-35)
  + bookingRequired Not Supported
  + fares Supported
  + products Partially Supported
    - product Partially Supported[[36]](#footnote-36)
  + discountCodes Not Supported
  + recoverTicketIds Not Supported

# Ticket API

Partially Supported

## GET /ticket

Not Supported

## POST /ticket

Partially Supported

#### Authorization

bobAuthZ: [pos,tvm]

### Request – Partially Supported

* Body
  + manifestMtbTemplate Supported
  + ticketholder NotSupported
  + ticketIssuer NotSupported[[37]](#footnote-37)
  + ticketPublicKey NotSupported[[38]](#footnote-38)
  + recoverTicketId NotSupported
  + issueMtb Supported[[39]](#footnote-39)
  + requestId Supported[[40]](#footnote-40)
  + startOfValidity NotSupported
  + bookingId NotSupported

### Response – Supported

* Headers
  + **Location** Supported
* Body
  + ticketIds Supported
  + settlementId Supported
  + ticketBundle Supported

## GET /ticket/{ticketId}

Partially Supported

#### Authorization

bobAuthZ: [pos,tvm]

### Request – Partially Supported

* Parameters
  + ticketId Supported

### Response – Supported

* Body
  + ticketId Supported
  + issuedAt Supported
  + ticketPayload Supported
  + mtb Supported
  + ticketholder Not Supported
  + activeStatus Supported
  + recoverStatus Supported
    - recovered Supported[[41]](#footnote-41)
    - recoveredAmount Supported[[42]](#footnote-42)
  + refundStatus Supported
  + refundableStatus Supported
  + hinderedStatus Not Supported

## GET /ticket/{ticketId}/refundableStatus

Supported

#### Authorization

bobAuthZ: [pos,tvm]

Point of Sale needs to be PID 14.

Ticket needs to be issued by the same Sales Channel as invoking the refund.

## GET /ticket/{ticketId}/refundStatus

Supported

#### Authorization

bobAuthZ: [pos,tvm]

Point of Sale needs to be PID 14.

Ticket needs to be issued by the same Sales Channel as invoking the refund.

## PUT /ticket/{ticketId}/refundStatus

Supported

#### Authorization

bobAuthZ: [pos,tvm]

Point of Sale needs to be PID 14.

Ticket needs to be issued by the same Sales Channel as invoking the refund.

## GET /ticket/{ticketId}/recoverableStatus

Supported[[43]](#footnote-43)

#### Authorization

bobAuthZ: [pos,tvm]

## GET /ticket/{ticketId}/recoverStatus

Supported[[44]](#footnote-44)

#### Authorization

bobAuthZ: [pos,tvm]

## PUT /ticket/{ticketId}/recoverStatus

Supported[[45]](#footnote-45)

#### Authorization

bobAuthZ: [pos,tvm]

## GET /ticket/{ticketId}/hinderedStatus

Not Supported

## PUT /ticket/{ticketId}/hinderedStatus

Not Supported

## PUT /ticket/{ticketId}/revoke

Supported

#### Authorization

bobAuthZ: [pos,tvm]

Point of Sale needs to be PID 14.

Ticket needs to be issued by the same Point of Sale as invoking the revoke.

## GET /ticket/{ticketId}/active

Not supported

## PUT /ticket/{ticketId}/active

Not supported

## GET /ticket/{ticketId}/event

Supported

#### Authorization

bobAuthZ: [val, ins]

## GET /ticket/{ticketId}/event/{eventId}

#### Authorization

bobAuthZ: [val, ins]

## POST /ticketbundle

Not supported

## GET /ticketbundle/{ticketBundleId}

Partially Supported

#### Authorization

bobAuthZ: [pos,tvm]

### Request – Partially Supported

* Parameters
  + ticketBundleId Supported

### Response – Supported

* Body
  + ticketBundleId Supported
  + ticketIds Supported
  + settlementId Supported
  + ticketPayload Supported
  + manifestId Not Supported
  + mtb Supported

# Validation API

Partially Supported

## POST /validation/{ticketId}

Partially Supported[[46]](#footnote-46), [[47]](#footnote-47)

#### Authorization

bobAuthZ: [val]

## POST /validation

Partially Supported[[48]](#footnote-48), [[49]](#footnote-49)

#### Authorization

bobAuthZ: [val]

## GET /whitelist

Not Supported.

## GET /ticklemacros

Supported

#### Authorization

Authorization policy: [anonymous]

## GET /ticklemacros/{serial}

Supported

#### Authorization

Authorization policy: [anonymous]

## GET /blacklist

Supported[[50]](#footnote-50)

#### Authorization

bobAuthZ: [val]

## GET /fraudcheck

Not supported

# Traveller API

Not Supported

# Booking API

Not Supported

# Appendix – Examples

## Authentication API - JWT

{

"jwtCompact": "eyJhbGciOiJFUzI1NiIsImtpZCI6IjE0OmF1dGgtdGVzdC0wMSJ9.eyJpc3MiOiIxNCIsInN1YiI6InBvczE0IiwiZXhwIjoxNTM4NTY4MzE4LCJib2JIb2siOiI5MTcxZGRmZjk0MTZkYjY2ODlkMjZjYzhhM2YzOGJkZmQ4MDU0YmNmIiwidnRDaGFubmVsSWQiOiIxIiwiYm9iQXV0aFoiOiJwb3MifQ.2b47QYoJOewvP6AHMSEnmTD4FqvvkfIIj1coiQHUOs8Nd50H1\_HoiA-sVoSksf0Hl1JQguCz4Tf8SU-fWbRm4w",

"header": {

"alg": "ES256",

"kid": "14:auth-test-01"

},

"payload": {

"iss": "14",

"sub": "pos14",

"exp": 1538568318,

"bobHok": "9171ddff9416db6689d26cc8a3f38bdfd8054bcf",

"vt-channel-id": "1",

"bobAuthZ": "pos"

}

}

## Product API – Route Property

 "route": [{

            "groupId": { "type": "line", "id": "9015014132103039" },

            "stopId": "9022014003085001"

        },

        {

            "groupId": { "type": "line", "id": "9015014132103039" },

            "stopId": "9022014014311002"

        }

    ]

1. The authentication API is not publicly available. [↑](#footnote-ref-1)
2. Only reporting of ticket events supported. Evaluation of MTB should be performed offline. Online operation where the MTB is evaluated server-side is not supported. [↑](#footnote-ref-2)
3. Attribute tokenTransaction ignored. [↑](#footnote-ref-3)
4. Only reporting of ticket events supported. Evaluation of MTB should be performed offline. Online operation where the MTB is evaluated server-side is not supported. [↑](#footnote-ref-4)
5. Attribute tokenTransaction ignored. [↑](#footnote-ref-5)
6. Accept-Language is ignored [↑](#footnote-ref-6)
7. Content-Language header is always set to “sv-SE” [↑](#footnote-ref-7)
8. Product is partially supported defined in 5.1 Response Definitions [↑](#footnote-ref-8)
9. Accept-Language is ignored [↑](#footnote-ref-9)
10. route is mutually exclusive with group. When supplying route data, it needs to include not only stop ids, but also line ids. The line id should be supplied as a groupId object for each item in the list of stops, see example in 10.2 [↑](#footnote-ref-10)
11. type must be set to “line” [↑](#footnote-ref-11)
12. group is mutually exclusive with route [↑](#footnote-ref-12)
13. groupType must be set to “zone” [↑](#footnote-ref-13)
14. Mutually exclusive with travellersPerCategory. Attribute “tra” should be an unsigned integer. [↑](#footnote-ref-14)
15. Mutually exclusive with travellersCategoryId [↑](#footnote-ref-15)
16. Content-Language header is always set to “sv-SE” [↑](#footnote-ref-16)
17. Product is partially supported defined in 5.1 Response Definitions [↑](#footnote-ref-17)
18. Content-Language header is always set to “sv-SE” [↑](#footnote-ref-18)
19. Product is partially supported defined in 5.1 Response Definitions [↑](#footnote-ref-19)
20. Accept-Language is ignored [↑](#footnote-ref-20)
21. Content-Language header is always set to “sv-SE” [↑](#footnote-ref-21)
22. Always empty list [↑](#footnote-ref-22)
23. Accept-Language is ignored [↑](#footnote-ref-23)
24. Content-Language header is always set to “sv-SE” [↑](#footnote-ref-24)
25. Accept-Language is ignored [↑](#footnote-ref-25)
26. Content-Language header is always set to “sv-SE” [↑](#footnote-ref-26)
27. Accept-Language is ignored [↑](#footnote-ref-27)
28. Content-Language header is always set to “sv-SE” [↑](#footnote-ref-28)
29. productSetManifest is defined in 5.1 Response Definitions [↑](#footnote-ref-29)
30. Accept-Language is ignored [↑](#footnote-ref-30)
31. Content-Language header is always set to “sv-SE” [↑](#footnote-ref-31)
32. productSetManifest is defined in 5.1 Response Definitions [↑](#footnote-ref-32)
33. productSetTitle and productSetDescription contain the same text [↑](#footnote-ref-33)
34. productSetTitle and productSetDescription contain the same text [↑](#footnote-ref-34)
35. Always false [↑](#footnote-ref-35)
36. Product is partially supported defined in 5.1 Response Definitions [↑](#footnote-ref-36)
37. Ignored [↑](#footnote-ref-37)
38. Ignored [↑](#footnote-ref-38)
39. Required to be true [↑](#footnote-ref-39)
40. Mandatory, and required to be unique within the scope of PID and sending BoB-entity. UUID4 is recommended. [↑](#footnote-ref-40)
41. Always false [↑](#footnote-ref-41)
42. As a consequence of recovered being always false, recoveredAmount is never set [↑](#footnote-ref-42)
43. Supported on API level, but from a business perspective Västtrafik tickets are not recoverable. [↑](#footnote-ref-43)
44. Supported on API level, but from a business perspective Västtrafik tickets are not recoverable. [↑](#footnote-ref-44)
45. Supported on API level, but from a business perspective Västtrafik tickets are not recoverable. [↑](#footnote-ref-45)
46. Only reporting of ticket events supported. Evaluation of MTB should be performed offline. Online operation where the MTB is evaluated server side is not supported. [↑](#footnote-ref-46)
47. Attribute tokenTransaction ignored. [↑](#footnote-ref-47)
48. Only reporting of ticket events supported. Evaluation of MTB should be performed offline. Online operation where the MTB is evaluated server side is not supported. [↑](#footnote-ref-48)
49. Attribute tokenTransaction ignored. [↑](#footnote-ref-49)
50. Always empty list [↑](#footnote-ref-50)