BILINFO SHARED SERVICES

Third party integration document

Abstract

Bilinfo Shared Services encompass Dealer Lookup and User Lookup Services that may be used in building a Bilinfo integration. This document describes the prerequisites and details of integrating into Bilinfo Shared Services.

BilInfo

bilinfo@bilinfo.dk

Version history

Version	Date	Authors	Comments
0.1.0	10/09/2018	Jacob Sønderskov (jsonderskov@ebay.com)	First draft version based on the *Bilinfo integration using the Case Plugin Architecture (2.4.0)* documentation.
1.0.0	12/09/2018	Jacob Sønderskov	First release version.
1.0.1	13/09/2018	Jacob Sønderskov	Change cover page and add abstract.

Contents

A. Disclaimer

Information presented here might be altered by eBay from time to time. Inconsistencies across the document are to be expected and they will be addressed in updates.

Any update will be specified in Version history.

Data and system integrity

Abuse of the system is forbidden in any regard. If you find a security issue or exploitation outside the original intent of the system, you are expected to report the exploit or bug to the Bilinfo team.

B. Versioning and Deprecation Policy

Versioning in Bilinfo Services is essential to achieving our vision behind Partner integrations in Bilinfo. Using the versioning principles described below will allow for your Bilinfo integrations to remain stable and fully functional as the Bilinfo business continues to evolve and mature.

New Versions of the Bilinfo Services

The versioning principles employed in Bilinfo Services largely follow that of the *Semantic Versioning Specification*[1]. The Semantic Versioning Specification, in short, specifies a version increment based on the backwards compatibility of the API or Web Service. A summary of the specification can be seen in the following Listing B-1:

> Given a version number `MAJOR.MINOR.PATCH`, increment the:
>
> 1. `MAJOR` version when you make incompatible API changes,
> 2. `MINOR` version when you add functionality in a backwards-compatible manner, and
> 3. `PATCH` version when you make backwards-compatible bug fixes.
>
> Additional labels for pre-release and build metadata are available as extensions to the `MAJOR.MINOR.PATCH` format.

Listing B-1: Semantic Versioning Specification 2.0.0 summary

The types of changes that are minor version changes and backward compatible are:

- Adding a new method (`GET`, `POST` etc.) to an API
- Adding a new property to the method response payload
- Adding a new *non-personal data*[2] property to an `iframe` communication

The types of changes that are major version changes and not backward compatible are:

- Removing existing method (`GET`, `POST` etc.) from an API
- Renaming existing method path
- Changing request body or query string for existing method
- Changing method response structure and/or property names
- Removing a property from an `iframe` communication
- Renaming a property from an `iframe` communication
- Renaming a message in an `iframe` communication
- Adding a new *personal data*[2] property to an `iframe` communication

[1] https://semver.org

[2] Personal data as defined in Regulation (EU) 2016/679 of 27 April 2016 (GDPR) and the Danish Data Protection Act.

In general, new major versions of Bilinfo Services will only be introduced, when existing interfaces does not allow for further evolution and improving of our Partner integrations without modification. Due to the need for Partner action, major versions are used as a last resort and are as such very rare. Minor version updates will require no Partner action.

Updating your Bilinfo integration

Updating your Bilinfo integration to support a new major version is non-optional as the existing integration paradigm is fundamentally changed. It is as such not possible to opt out without risking major problems with your Bilinfo integration. Minor versions, however, are fully optional, but may contain new fields, which may enrich the experience and value of your Bilinfo integration.

To assist Partners in upgrading their Bilinfo integration with minimal efforts, each major version will be associated with a *migration chapters* added to this document. Minor version changes are specified primarily in the Version History and is subject to the reader to adhere to the changes.

Deprecation Policy and Supported Versions

Bilinfo Services will support older versions for a grace period appropriate to the contractual obligations. After that time, integrations based on older versions may no longer work or experience severe operational issues.

1. Introduction

1.1 Purpose and Scope

This document will present the technical specifications for the Bilinfo Shared Services, which integrating parties may use to access additional information from the Bilinfo.net context.

1.2 References

Documents relevant to the reading of this document are listed here. Links and other external resources accessible via the internet are referenced via footnotes relative to the term or technology. You should have access to every document mentioned in this list. If that is not the case, contact Bilinfo.

Table 1-1: Document references

Document name	Description	Link
Bilinfo Auth Services (Integration)	Describes how to integrate into the Bilinfo Auth Services, SSO and OAuth2.0, needed to respectively authenticate and authorize access to Bilinfo.net.	TBA

1.3 Definitions and acronyms

The definitions and acronyms defined in Table 1-2 cover frequently used concepts, terms and acronyms used throughout this document. It is suggested that the reader acquaints him- or herself with the key concepts and refer to this list, when in doubt.

Table 1-2: Definitions and acronyms

Term/acronym	Definition
Bilinfo Services	Includes – but not limited to – Bilinfo Auth Services, Bilinfo Shared Services, Bilinfo Finance Offer On Platform integrations and Bilinfo Case Plugin integrations.
Partner	Synonym for the integrating party.

2. Prerequisites

This chapter specifies the prerequisites needed to communicate with the Bilinfo Shared Services.

2.1 Security

All Bilinfo Services use SSL/TLS (HTTPS) for transport layer security.

2.2 Bilinfo Auth Services

To consume the Bilinfo Shared Services, you must integrate to Bilinfo Authorization Service specified in the Bilinfo Auth Services document referenced in section 1.2 References.

In short, you must specify the Authorization `scope`'s for the services you wish to consume. In return you will be provided with a `client_id` and `client_secret` to be used when acquiring Bearer tokens.

3. Bilinfo Shared Services

Bilinfo provides a number of shared services to integrating party. This chapter will cover each available endpoint in terms of

- Base paths
- Endpoints

The Bilinfo service are RESTful web services and use JSON payload responses.

3.1 Dealer Lookup Service

The Dealer Lookup Service may be used to map a dealer CVR number to Bilinfo.net relative information. This information may be used to maintain mapping towards the Bilinfo.net system.

3.1.1 Base paths

Table 3-1 shows the base paths for the available Dealer Lookup Service environments.

Table 3-1: Dealer lookup service base paths

Environment	Path
QA	`https://qa01-dealerlookupservice.ecgh.dk/api`
Production	`https://dealerlookupservice.ecgh.dk/api`

3.1.2 Endpoints

The following section iterates through the endpoints available in the Dealer Lookup Service API. Each endpoint has a specific path, which must be appended to the base path, and must be requested using a scope Bearer token as specified in Section 2.1.

Dealers CVR

The Dealers CVR endpoint is used to acquire dealer information via their respective CVR[3] number.

- Path: `/dealers/cvr/{cvr}`
- Verb: `GET`
- Authorization scope: `https://dealerlookupservice.ecgh.dk/auth/read`

Response

The service may return a list of dealers if more than one outlet is associated with the same CVR number. The service returns http status code 404, if no dealers were found with the given CVR number.

Table 3-2 shows the possible responses provided by the Dealers CVR endpoint with Table 3-3 specifying the associated data type(s).

[3] Det Centrale Virksomhedsregister (en. The Central Company Register)

Table 3-2: Dealers CVR endpoint response specification

Status code	Description
200	An array with elements of type Dealer
404	When no dealer with the given cvr could be found
500	Internal server error while processing the request

Dealer

Table 3-3 shows the model specification for the Dealer type

Table 3-3: Dealer type data specification

Field	Type	Description
Id	string	Bilinfo.net guid, e.g. 42304136-fdca-df11-9e98-0025b3e6b7d8
Cvr	string	CVR number
DealerId	string	Bilinfo classic ID, e.g. 100 (BilInfo ApS)
Email	string	
Name	string	
Address	string	
Phone	string	
ZipCode	number	

Example

An example request for the Dealers CVR endpoint can be seen in Listing 3-1.

```
GET https://<server>.ecgh.dk/api/dealers/cvr/20618175
Host: gw.bilinfo.net
Authorization: Bearer eyJ0eXA…PL4UPsfp

[
    {
        "Id": "42304136-fdca-df11-9e98-0025b3e6b7d8",
        "CrmNumber": "233304",
        "Cvr": "20618175",
        "DealerId": "100",
        "Version": 0,
        "Email": "niels@bilinfo.dk",
        "Name": "BilInfo ApS",
        "Address": "Axel Kiers Vej 11",
        "Phone": "96194400",
        "ZipCode": 8270,
        "CustomerType": null
    },
    …
]
```

Listing 3-1: Dealer CVR endpoint example

3.2 User Lookup Service

The User Lookup Service allows the integrating party to retrieve information about the users/sales persons of a dealerships. This service may be used for mapping specific sales persons access.

3.2.1 Base paths

Table 3-4 shows the base paths to which the endpoint paths must be appended in order to query the User Lookup Service.

Table 3-4: User service base paths

Environment	Path
QA	`https://qa01-userlookupservice.ecgh.dk/api`
Production	`https://userlookupservice.ecgh.dk/api`

3.2.2 Endpoints

The following section iterates through the endpoints available in the User Lookup Service API. Each endpoint has a specific path, which must be appended to the base path, and must be requested using a scope Bearer token as specified in Section 2.1.

Users

The Users endpoint is used to acquire user information via a specified dealer id.

- Path: `/users/dealerid/{dealerid}`
- Verb: `GET`
- Authorization scope: `https://userlookupservice.ecgh.dk/auth/read`

The `dealerid` is supplied from one of two places.

1. From the SSO token supplied by the Host.
2. From the Dealer Lookup Service response

Response

Table 3-5 shows the possible responses provided by the Users endpoint with Table 3-6 specifying the associated data type(s).

Table 3-5: Users endpoint response specification

Status code	Description
200	An array with elements of type User
400	Malformed dealer identifier
404	When no users are found on the requested dealer, or if the requested dealerid does not exist
500	Internal server error while processing the request

User

Table 3-6 shows the model specification for a User type.

Table 3-6: User type specification

Field	Type	Description
Id	string	Bilinfo user id
Name	string	
Email	string	
Address	string	
ZipCode	number	
Cellphone	string	

Example

An example request for the Dealers CVR endpoint can be seen in Listing 3-2.

```
GET https://<server>.ecgh.dk/api/users/dealerid/42304136-FDCA-DF11-9E98-
0025B3E6B000
Host: gw.bilinfo.net
Authorization: Bearer eyJ0eXA…PL4UPsfp

[
    {
        "Id": "13819",
        "Name": "Test user ",
        "Email": "test@test.dk",
        "Address": "Test 123",
        "ZipCode": "8000",
        "Cellphone": "12344321"
    },
    …
]
```

Listing 3-2: Dealer CVR endpoint example