



Baloise– integration options fact sheet for distributors

Last updated: 13 July 2020

Contents

1. SNIPPET CODE	3
1.1 DEFINITION:	3
1.2 SERVICE DESCRIPTION:	3
1.3 ADVANTAGES AND DISADVANTAGES:	3
2. LINKING TO API	3
2.1 DEFINITION:	3
2.2 SERVICE DESCRIPTION:	3
2.3 ADVANTAGES AND DISADVANTAGES:	4
3. PLUG-IN	4
3.1 DEFINITION:	4
3.2 SERVICE DESCRIPTION:	4
3.3 ADVANTAGES AND DISADVANTAGES:	4

1. Snippet code

1.1 Definition:

A snippet code is a short code section which is used to display the KASKO widget (front end) on your website. The KASKO widget is an end-to-end solution which is loaded on your website and facilitates the entire insurance transaction.

1.2 Service description:

- KASKO provides a JS embed integration description;
- KASKO generates an individual agent ID for you for the purposes of reporting;
- unlike with integration via REST API, a front-end display aligned with the KASKO front-end framework is included and does not require to be created by your developers;
- the front end will be supplied as a white label solution, i.e. it can be adapted to the company's own colour scheme, logo, etc. (look & feel).

→ *We recommend integration via snippet if you do not plan on developing your own front end and want to use as few IT capacities as possible.*

1.3 Advantages and disadvantages:

- + low costs
- + low integration expense on the part of the insurer and/or the distributor
- + option of front end with own look & feel
- use of own front end not possible.

2. Linking to API

2.1 Definition:

An API (Application Programming Interface) is a set of functions used for communication between different services or systems. There are different versions of APIs; KASKO provides a REST API.

2.2 Service description:

- The product is provided via API, incl. interface description.
- KASKO provides integration support and is always on hand for any queries regarding interface integration.
- KASKO generates an individual agent ID for you for the purposes of reporting;
- There is no KASKO front end included; this would have to be developed and installed by you. The key thing here is that the fields must be mapped between the front end and the API.

→ *Recommended if a front end already exists or is in the course of being developed by the insurer. Alternatively, linking up to an API will also be necessary if integrating within an existing purchase process, or if the product is to be integrated on product aggregation platforms.*

2.3 Advantages and disadvantages:

- + use of own front end
- + integration on product aggregation platforms possible
- + integration in existing purchase processes without anchor in the front end
- greater integration expenses for you.

3. Plug-in

3.1 Definition:

A plug-in is used to link a new feature, in this case the option of purchasing insurance, to an existing system, e.g. an e-commerce shop.

3.2 Service description:

- Use of a plug-in to integrate an additional insurance module in a pre-existing product sale sequence.
- Plug-ins are developed specific to the shop.
- The plug-in is either configured by the insurer itself, or by a relevant third-party provider, e.g. plentyMarketplace.
- The collection of payments can be allocated to the respective partners automatically. As such, no additional payment flows are necessary.

→ *Integration via plug-in is recommended if a compatible insurance module is to be integrated as an add-on to a pre-existing product.*

3.3 Advantages and disadvantages:

- + adoption of client data already entered – one-click purchase
- + direct and targeted integration of the add-on product
- + direct distribution of payment collection without any additional payment flows
- + you will not incur any development expenses.