MDP - Web hook platform proposal

Оглавление

[1.](#_heading=h.30j0zll) Introduction 2

[2.](#_heading=h.2et92p0) Gateway API 3

[2.1.](#_heading=h.4d34og8) Create web hook 3

[2.2.](#_heading=h.1ksv4uv) Retrieves a list of web hooks 3

[2.3.](#_heading=h.44sinio) Receive a single web hook 3

[2.4.](#_heading=h.2jxsxqh) Remove an existing web hook 3

[2.5.](#_heading=h.z337ya) Web hook object 4

[2.6.](#_heading=h.3j2qqm3) Object types 5

[2.7.](#_heading=h.1y810tw) Event types 5

[2.8.](#_heading=h.4i7ojhp) Example 5

[2.9.](#_heading=h.2xcytpi) Web hook payload 6

[3.](#_heading=h.1ci93xb) Event sourcing micro service 7

[3.1.](#_heading=h.3whwml4) Diagram 7

[3.2.](#_heading=h.2bn6wsx) Configuration 7

[4.](#_heading=h.qsh70q) Payload Delivery micro service 9

[4.1.](#_heading=h.3as4poj) Configuration 9

[4.2.](#_heading=h.1pxezwc) Alternate delivery with Microsoft Azure 9

[4.3.](#_heading=h.49x2ik5) Comparison of delivery method Azure with custom solution 9

[5.](#_heading=h.2p2csry) Activity diagram 10

[6.](#_heading=h.147n2zr) Database digram 11

# Introduction

Web hook platform is a service that sends notifications to handlers provided by partners. Notifications will be triggered at the moment of a new booking event created, modified or canceled.

The service allows to specify exactly what the partner wants to receive. Filters can be applied to the object type (accounting, reservation...), event type (creation, cancelation...) and additionally for maximum flexibility use the JSON filtering syntax.

The service APIs are organized by types and fields as well as by endpoints. Access to all features is provided under one or more endpoints for each partner's subscription.

Web hook platform proposal is a service in corporate cloud that consists of three components:

* API gateway for partners to manage the details of the subscription
* MicroService to source booking events based on corporate Queue
* Microservice to deliver events to partners

# Gateway API

The role of the API gateway is:

* Configure partner Callback Url to let be notified when certain events happen
* Set up an event map to collect only important data
* Enable Access Secret and other credentials

### Create web hook

Configure a web hook object to represent your web hook endpoint, payload URL & secrect, events and objects.

|  |  |
| --- | --- |
| **POST** | **/api/v1/web hooks** |

### Retrieves a list of web hooks

Retrieves a list of web hooks.

|  |  |  |
| --- | --- | --- |
| **GET** | **/api/v1/web hooks/list** | /all |

### Receive a single web hook

Retrieves a single web hook subscription. The properties desired in the result can be specified.

|  |  |
| --- | --- |
| **GET** | **/api/v1/web hooks/{Id}** |

### Remove an existing web hook

Delete a web hook subscription

|  |  |
| --- | --- |
| **DELETE** | **/api/v1/web hooks/{Id}** |

### Web hook object

|  |  |  |  |
| --- | --- | --- | --- |
| 1. | **Id** | string | required |
|  | Unique identifier | | |
| 2. | **name** | string | required |
|  | Name for web hook | | |
| 3. | **payload\_url** | string | required |
|  | URL of the target to which to POST event batches. Only ports 80 for http and 443 for https can be set. | | |
| 4. | **secret** | string | required |
|  | API Key Secret  obtained within registration | | |
| 5. | **events** | string array | required |
|  | Events you'd like to listen | | |
| 6. | **objects** | string array | required |
|  | Objects you're interested | | |
| 7. | **filters** | Array of objects |  |
|  | Custom JSON filters for objects | | |
| 7. | **isActive** | bool |  |
|  | The status of the web hook. When false, the target will no longer receive batches. | | |

### Object types

* Accounting
* Reservation
* Loyalty

### Event types

* Creation
* Modification

### Example

For example, a partner wants to subscribe for events only about accounts from Germany, reservations with children or VIP guests

|  |
| --- |
|  |

### Web hook payload

Each web hook event payload also contains properties unique to the event. You can find the unique properties in the individual event type sections.

But common properties are:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Parameter | Type | Location |
| 1. | **signature** | string | header |
|  | This header is sent if the web hook is configured with a secret. This is the HMAC hex digest of the request body, and is generated using the SHA-256 hash function and the **secret** as the HMAC key | | |
| 2. | **created** | The timestamp when the web hook was created | body |
| 3. | **object** | string | body |
|  | The object is involved | | |
| 5. | **event** | string | body |
|  | The event is fired | | |
|  | **data** |  | body |
| 6. | The details of object | | |

# Event sourcing micro service

The task of this service is to listen to the topic of the corporate message queue, pick events in which partners are interested, multiply each message for all active subscriptions matching the selection conditions and put them in the delivery queue.

### Diagram

|  |
| --- |
| * **S** |

### Configuration

At the global configuration level, the following is defined:

* Source Stream Credentials
  + Cluster Name
  + Broker Id, Host & Port
  + Consumer Group
  + List of involved Topics
* Delivery Query Credentials
  + Cluster Name
  + Broker Id, Host & Port
  + Topic

# Payload Delivery micro service

The task of this service is to continuously and instantly deliver events to partners through assigned channels, repeat delivery attempts in the event of an external system failure, and notice failed points.

### Configuration

At the global configuration level, the following is defined:

* Delivery Query Credentials
  + Cluster Name
  + Broker Id, Host & Port
  + Topic
* Retry Policy Settings
  + Time Limit

Any web hook batch that does not receive an HTTP 200 response will be retried for a total of this period in minutes before the data is discarded.

* + Frag Limit

Any web hook batch that does not receive an HTTP 200 response will be retried for a total of this time before the data is discarded.

### Alternate delivery with Microsoft Azure

The task of this service is to continuously and instantly deliver events to partners through assigned channels, repeat delivery attempts in the event of an external system failure, and notice failed points.

### Comparison of delivery method Azure with custom solution

# Activity diagram

# Database digram

