VICARD: Care Provider System Integration with VVK-Online proxy API

Release Information	1.00
Target audience	This documentation is intended for developers and system integrators which enable customer systems for interaction with the VICARD Check-in proxy API.
Summary	The Check-in API is used to access check-in information e.g. the latest check-in and to manage check-in locations. The reader of this documentation will find the API description in detail as well as further information about the access and authorization mechanisms.

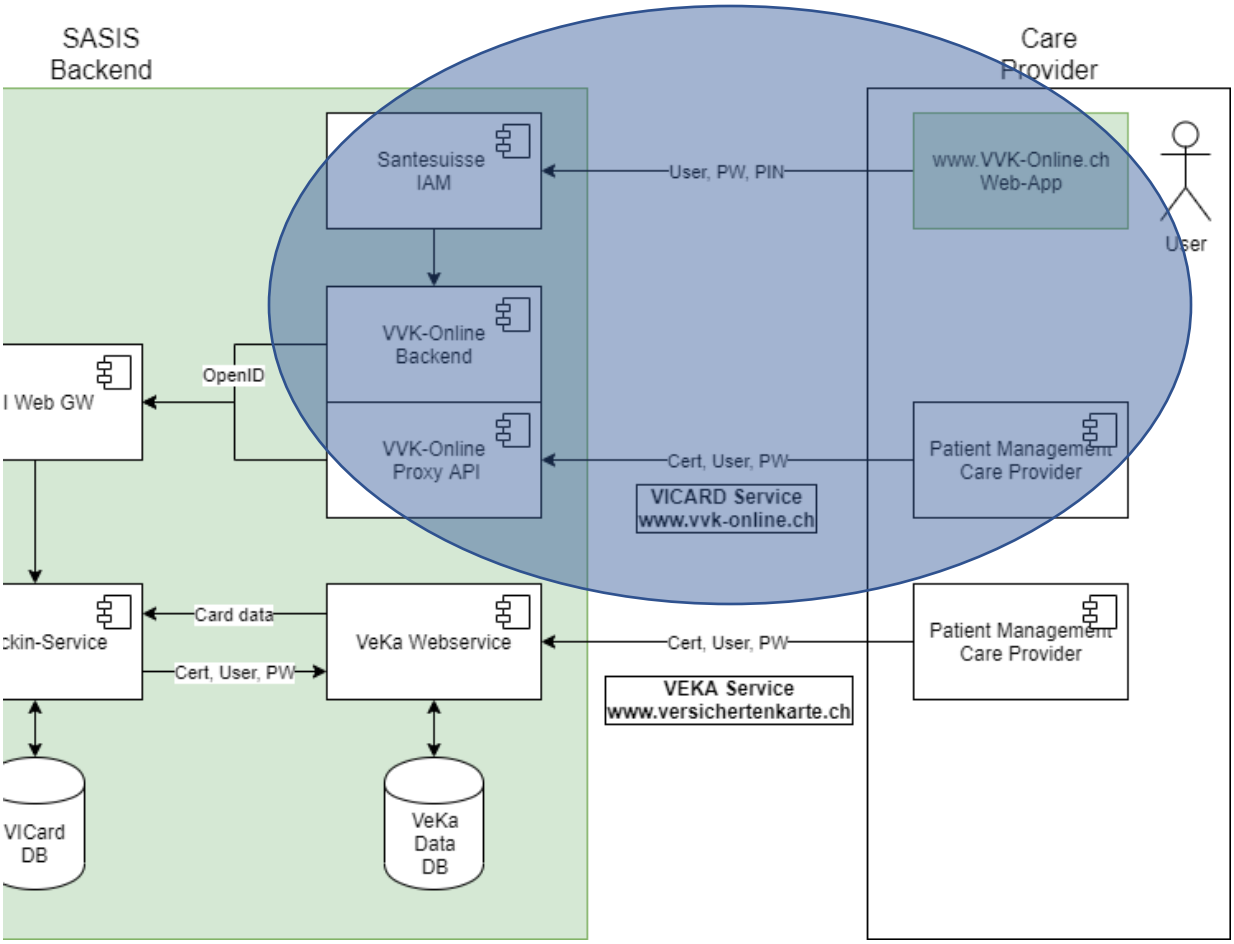
Content

The System Integration is done on the VICARD Check-in Live environment.

As an Integrator, you should request the Testuser VVK-Online Credentials from SASIS, Veka-Center: `support@sasis.ch`.

Introduction

When a VICARD user/insured person performs a check-in with his mobile, the check-in information is stored using the Checkin-Service called by the mobile.



This Checkin-Service is an extension to the existing SASIS Webservice (Veka-Service) used by Care Providers. It provides endpoints for two different Use Cases:

1. Get the latest Checkin for a check-in location
2. Manage the Checkin locations

A major requirement was to maintain the current authentication processes.

That's why VVK Online exposes a proxy API that handles all requests from external clients that require certificate authentication.

Authentication

The currently used VVK-Service authentication credentials will be valid and can be reused in order to access the VVK Online proxy API.

For successful authentication, each request to the VVK Online proxy API must provide the following credentials:

1. Client certificate, to be passed with each request.
2. A JSON-formatted credentials object (username, password, zsr), to be passed with each request body that is structured as shown below:

```
{
"userName": "testUser",
"password": "testPassword",
"zsr": "testZsr",
 }
```

API Functions

Check-Ins

The main use of the API is the retrieval of *check-ins of insured persons* by using

`/api/checkin/checkins/location/{checkinLocationGuid}/latest/{lookbackMinutes}`

Such a check-in contains the following information:

{

"checkinId": 208,

"checkinLocationGuid": "123e4567-e89b-12d3-a456-426655440000",

"checkInDateTime": "2020-12-07T08:20:15.0957081",

"cardIdentificationNumber": "80756003760019034683",

"givenName": "Elizabeth",

"familyName": "Lincoln",

"birthDate": "1941-09-26",

"gender": "female",

"personalIdentificationNumber": "7569999999892",

"institutionIdentificationNumber": "123",

"institutionIdentificationName": "Sanacur 2 (Testversicherer)",

"cardValidFrom": "2018-03-01",

"cardValidUntil": "2024-03-31" }

}

The resulting information can be used in most existing care-provider systems to identify the insured person that checked in. (which is today typically done by checking/scanning the physical VEKA card or entering the data manually).

Time and Location of check-ins

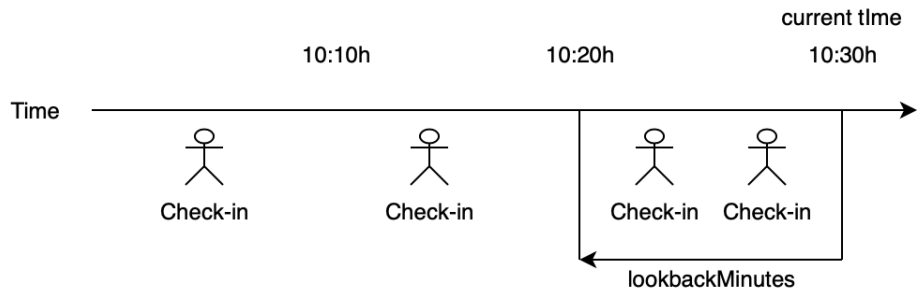
This endpoint above returns a list of check-ins that can be filtered using ***time*** and ***location***.

Note the fields

- `checkInDateTime (from check-in process) and`
- `checkinLocationGuid (from QR/NFC)`

from the check-in data above.

Time filtering is based on `checkInDateTime`. It is done by using the parameter "lookbackMinutes". This number defines the time interval "current time to current time-lookbackMinutes" (see figure below). A lookbackMinutes value of 10 returns the check-ins in the last 10 minutes. A value of 60 returns the check-ins of the last hour.



The **location** filtering is done by using the "checkinLocationGuid". The locations are manageable by the API.

Use

```
/api/checkin/checkinlocations
```

to create a new check-in location. When a `checkinlocation is created it is associated with the ZSR (corresponding to "institutionIdentificationNumber") of the logged in end user.`

Use

```
/api/checkin/checkinlocations/{guid}
```

to get details of a check-in location

Use

```
/api/checkin/checkinlocations/zsr/{zsr}
```

to get check-in locations by associated ZSR numbers.

Use

```
/api/checkin/checkinlocations/update/{guid}
```

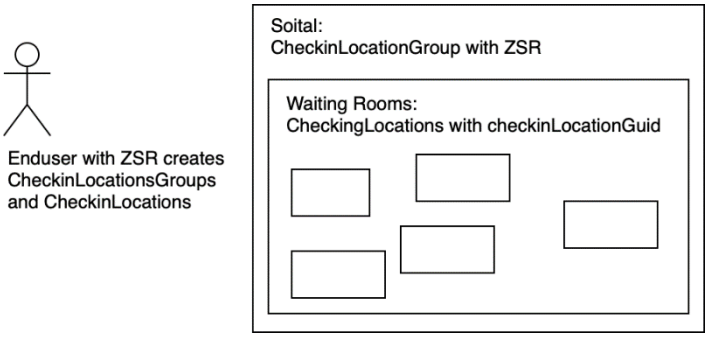
to update an existing check-in location

Use

```
/api/checkin/checkinlocations/delete/{guid}
```

to delete an existing check-in location.

CheckinLocationGroups



CheckinLocations are associated with CheckinLocationGroups. Every CheckinLocationGroup has a ZSR.

E.g. Unispital Zürich (represented as CheckinLocationGroup) can have several waiting-rooms (represented as CheckinLocations).

Use

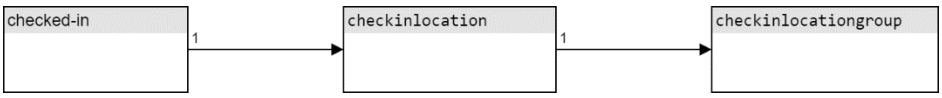
`/api/checkin/checkinlocationgroups/zsr/{zsr}`

to retrieve the check-in location group that is associated with the ZSR.

If no group exists for the ZSR, the endpoint will create a group and return it. An already existing group will be returned directly.

Every logged-in end user is associated with a ZSR (via the login) that used to create the check-in locations.

Swagger documentation



The swagger documentation includes the definition of:

- the different endpoint routes
- HTTP methods
- route parameters
- request bodies
- response status codes
- response bodies

It also provides the ability to test the endpoints directly in the documentation page.

Link: https://www.vvk-online.ch/Web/swagger/ui/index#/