



< Developer  
Days2023 >

# Event Triggers to Systems Behind the Firewall



Speaker: **Menno Pieters**;  
title: Principal Solution Architect;  
company: **SailPoint**;



# Agenda

**Introduction**

**Usage**

**API**

**Demo**

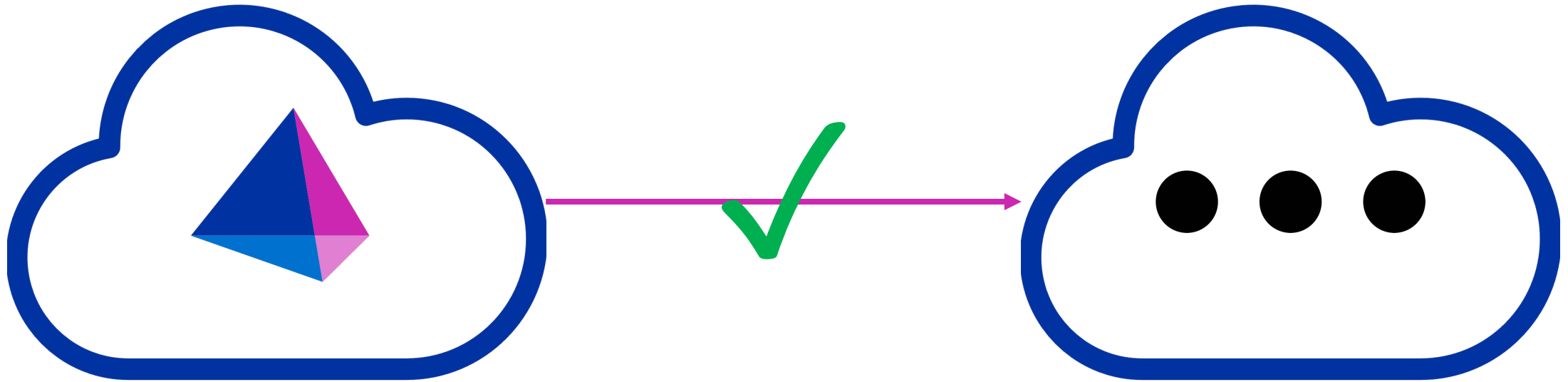


```
if localVarHTTPResponse.StatusCode >= 300 {  
    newErr := &GenericOpenAPIError{  
        body: localVarBody,  
        error: localVarHTTPResponse.Status,  
    }  
}
```

# Introduction

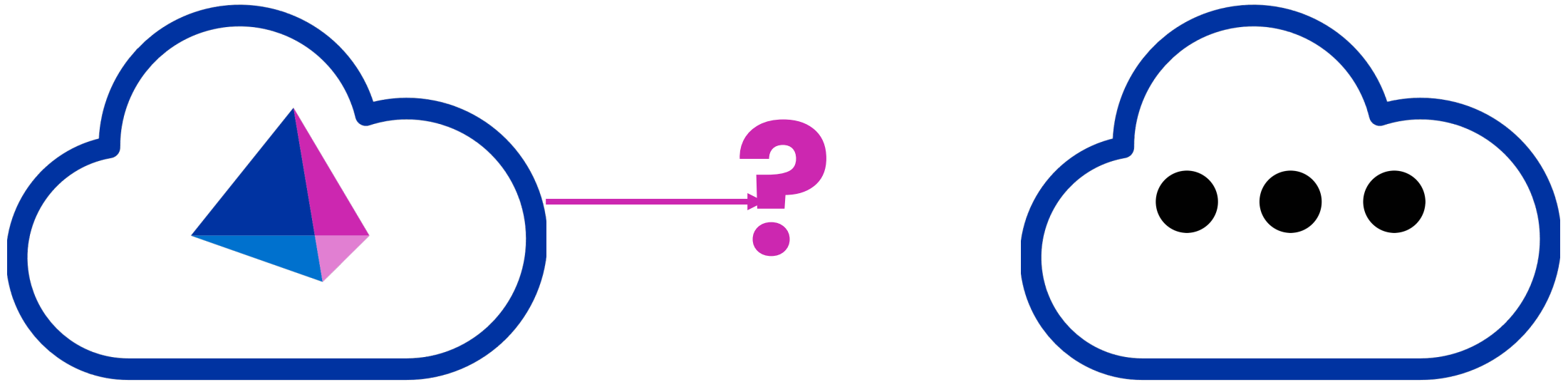
```
if localVarHTTPResponse.StatusCode == 400 {  
    var v ErrorResponseDto  
    err = a.client.decode(&v, localVarBody, localVarHTT  
    if err != nil {  
        newErr.error = err.Error()  
        return localVarReturnValue, localVarHTTPResp  
    }  
}
```

# Business Problem



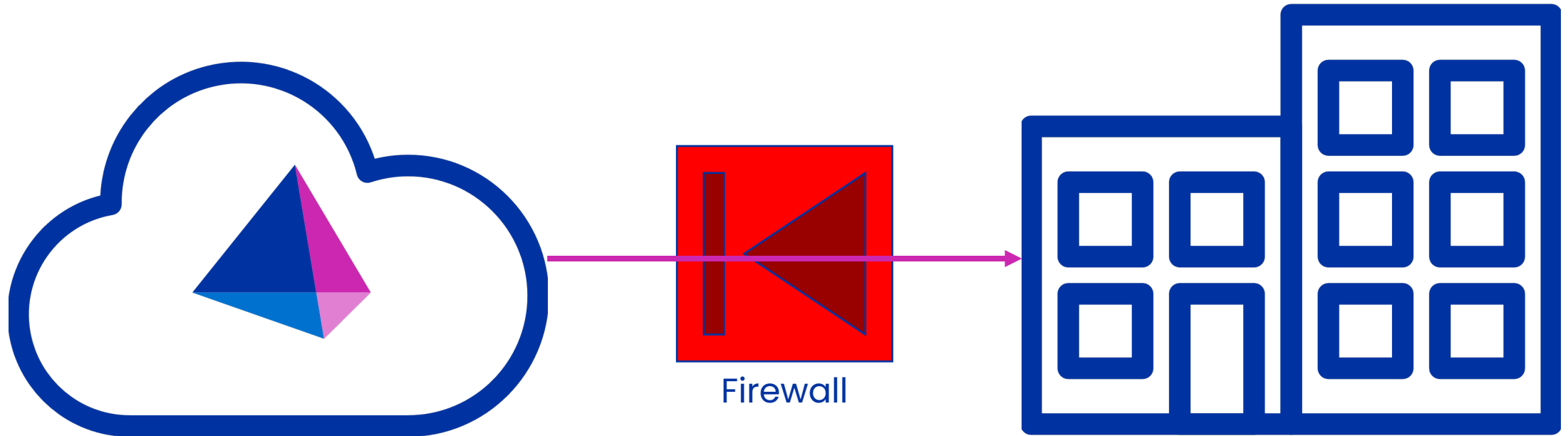
IdentityNow Triggers are cloud to cloud

# Business Problem



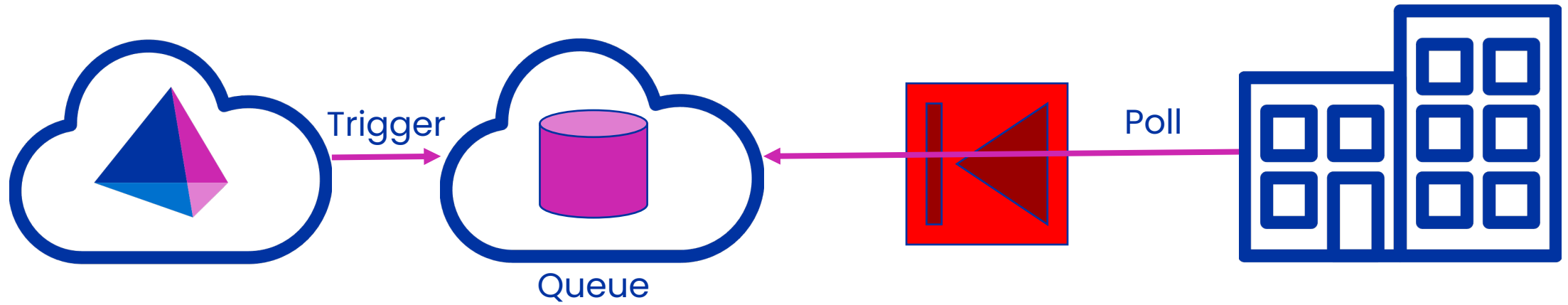
IdentityNow Triggers are mostly “fire and forget”, no retry.

# Business Problem



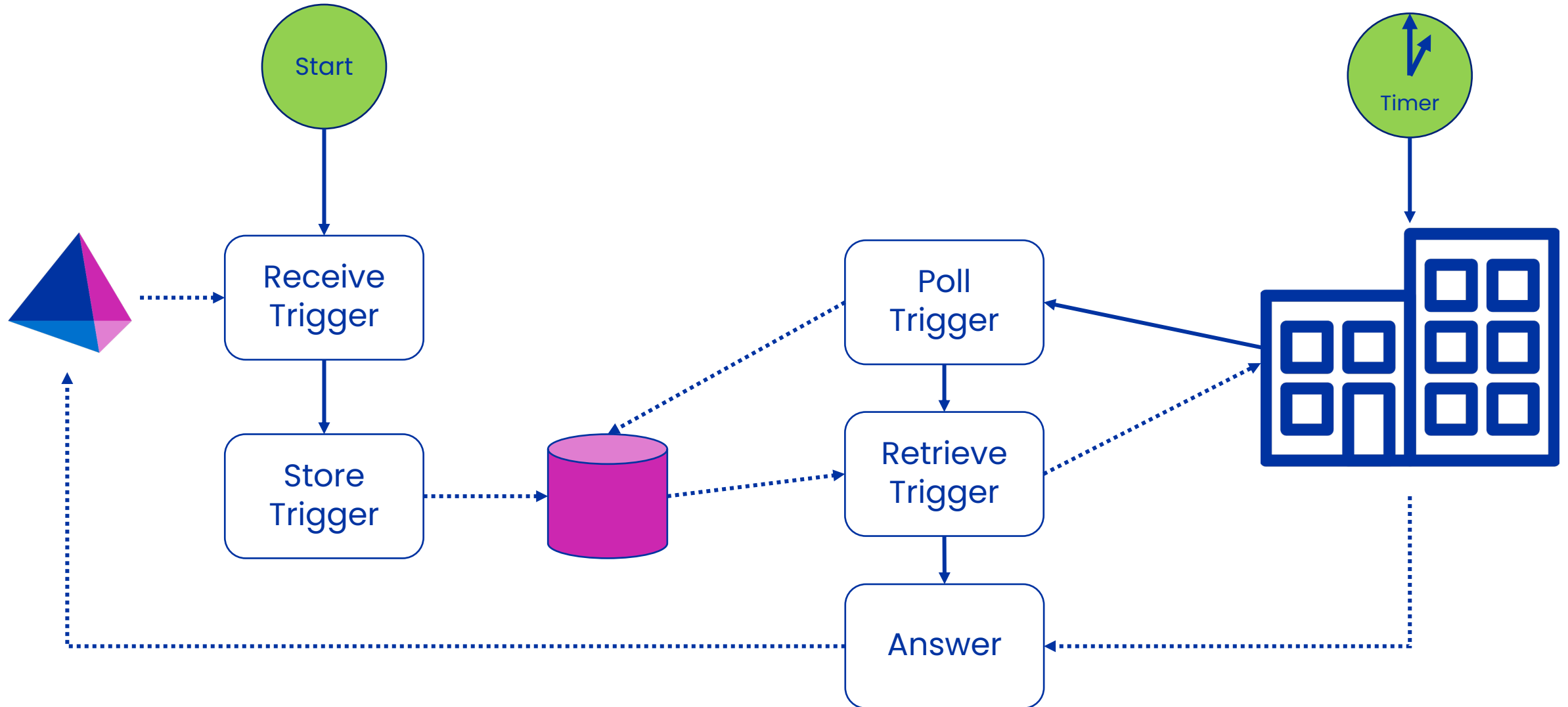
Notifying on-prem systems requires firewalls to be opened

# Solution



Trigger Cloud Queue

# Solution





```
if localVarHTTPResponse.StatusCode >= 300 {  
    newErr := &GenericOpenAPIError{  
        body: localVarBody,  
        error: localVarHTTPResponse.Status,  
    }  
}
```



# Usage

```
if localVarHTTPResponse.StatusCode == 400 {  
    var v ErrorResponseDto  
    err = a.client.decode(&v, localVarBody, localVarHTT  
    if err != nil {  
        newErr.error = err.Error()  
        return localVarReturnValue, localVarHTTPResp  
    }  
}
```



# Deployment

- Create database (MySQL)
- Extract war file
- Edit `WEB-INF/classes/queue.properties`

```
db.pool.url=jdbc:mysql://localhost/etscloudqueue?useSSL=false
```

```
db.pool.user=etscloudqueue
```

```
db.pool.password=*****
```

```
db.pool.driver=com.mysql.jdbc.Driver
```

```
db.queue.maxage=3600
```

```
admin.user=spadmin
```

```
admin.pass={SSHA256}*****
```

```
tokenSalt=*****
```

```
if localVarHTTPResponse.StatusCode >= 300 {  
    newErr := &GenericOpenAPIError{  
        body: localVarBody,  
        error: localVarHTTPResponse.Status,  
    }  
}
```



# API

```
if localVarHTTPResponse.StatusCode == 400 {  
    var v ErrorResponseDto  
    err = a.client.decode(&v, localVarBody, localVarHTT  
    if err != nil {  
        newErr.error = err.Error()  
        return localVarReturnValue, localVarHTTPResp  
    }  
}
```

# Terminology



- `{{url}}`: base url for the service, e.g. <https://ets.example.com/queue>
- `{{user_id}}`: user identifier (GUID)
- `{{queue}}`: queue identifier (GUID)

# Ping



```
GET {{url}}/rest/ping
```

Returns:

```
Pong
```



# Admin Authentication

- User / Password defined in `queue.properties`

```
admin.user=spadmin
```

```
admin.pass={SSHA256}ZE...Ho=
```

- BASIC authentication
  - Use a long password!
- Generate hash:
  - Command line (see README)
  - API: `GET {{url}}/rest/admin/hash/{{password}}`

```
GET {{url}}/rest/admin/hash/V3ryS3cr3tP@ssw0rd
```

```
{SSHA256}bVNpM0tGcWU=$ovUcqCqc8+Y+EFBAm/Pb5ZALaU97KlgBa4Km+H0dRis=
```



# Admin Features

- List/create/delete queues
- List/create/delete users
- List/create/delete user tokens
- Authorize/de-authorize users to queues (read and/or write)



# IdentityNow Client Features

- Post trigger to queue

```
POST {{url}}/rest/event/trigger/{{queue}}
{
  "fileName": "Search Export.2020-07-22 12'34 GMT",
  "ownerName": "John Doe",
  "query": "attributes.cloudLifecycleState:inactive AND
@accounts(disabled:false AND source.name:LDAP)",
  "searchName": "Inactive Identities with Active Accounts",
  ...
}
```





# Receiving Client Features

- Peek: check for new message without removing it from the queue

```
GET {{url}}/rest/event/peek/{{queue}}
```

- Poll: get one message and remove it from the queue

```
GET {{url}}/rest/event/poll/{{queue}}
```

Both will return one message, exactly as it was queued.

First in, first out.

```
if localVarHTTPResponse.StatusCode >= 300 {  
    newErr := &GenericOpenAPIError{  
        body: localVarBody,  
        error: localVarHTTPResponse.Status,  
    }  
}
```



# Demo

## Postman API

```
if localVarHTTPResponse.StatusCode == 400 {  
    var v ErrorResponseDto  
    err = a.client.decode(&v, localVarBody, localVarHTT  
    if err != nil {  
        newErr.error = err.Error()  
        return localVarReturnValue, localVarHTTPResp  
    }  
}
```

# More Information

Git Repo:

<https://github.com/menno-pieters-sp/ets-cloud-queue>





**[Thank you!]**