

Installation der Eventing REST API

Voraussetzungen

Die API ist in Java entwickelt und muss in einem Application Server installiert werden. Folgende Voraussetzungen müssen erfüllt sein:

1. Java 8 JVM (Oracle oder OpenJDK)
2. Application Server (z.B. Tomcat ab Version 7)
3. Zugriff auf die PostgreSQL-Datenbank

Installation

Die API kann wie jede andere Webapp im Tomcat installiert werden (z.B. Kopie in den Order `<tomcat-dir>\webapps`, Hochladen via Tomcat Manager). Danach muss die Konfiguration angepasst werden.

Konfiguration

Anpassung der Datenbank-Verbindung

Die Verbindung zur Datenbank wird in der Datei `<tomcat-dir>\webapps\wv-eventing-webapp\WEB-INF\classes\wv\hibernate-eventing.cfg.xml` definiert.

Der Eintrag `<property name="connection.url">` muss entsprechend angepasst werden. Er folgt dabei diesem Schema:

`jdbc:postgresql://<host>:<port>/<db-name>?currentSchema=<db-schema>`

Also z.B.:

`jdbc:postgresql://localhost:5432/wv-eventing?
currentSchema=sensorweb2`

Ggf. muss auch der Nutzer (`connection.username`) und das Passwort (`connection.password`) für den Zugriff auf die Datenbank angepasst werden.

Konfiguration der Berechtigungen

Die Berechtigungen für die Nutzung der API werden per Gruppenzuweisungen vorgenommen. Um diese anzupassen, kann die Datei `<tomcat-dir>\webapps\wv-eventing-webapp\WEB-INF\classes\wv\group-policies.json` angepasst werden. Diese Datei ist beispielsweise wie folgt aufgebaut:

```
{
  "adminGroupNames": [
    "sensorweb-admins",
    "sensorweb-administrators"
  ],
  "groupPrefix": "sensorweb-",
  "adminSuffix": "_admin"
}
```

In dem JSON-Array `adminGroupNames` werden die Gruppennamen eingetragen, die administrativen Zugang zum System haben (Anlegen von Regeln und Abonnements).

Ist ein Nutzer nicht in einer der Administratorgruppen, kann er sich am System anmelden, hat jedoch keine Schreibrechte.

Die Eigenschaft `groupPrefix` definiert einen Präfix, der Gruppennamen voran gestellt sein muss. Ist dies für eine Gruppe nicht der Fall, wird sie in der API nicht berücksichtigt. Der Wert `adminSuffix` definiert die Administrationsgruppe zu einer anderen bestehenden Gruppe. Z.B. ist nach oben stehender Konfiguration die Gruppe `sensorweb-opladen_admin` die Administrationsgruppe von `sensorweb-opladen`:

Mitglieder der Administrationsgruppe können die Gruppen-Abonnements verwalten.

Weitere Konfiguration

In der generellen Konfigurationsdatei (`<tomcat-dir>\webapps\wv-eventing-webapp\WEB-INF\classes\config-default.json`) stehen weitere Konfigurationsmöglichkeiten zur Verfügung.

- **Verkündfte Series REST API:** Die Basis-URL einer verknüpften (= gleiche Datenbank) Series REST API lässt sich über den Parameter `timeSeriesApiBaseUrl` einstellen. Wichtig ist, dass am Ende `%s` steht, da dies als Platzhalter für eine Series-ID genutzt wird.

Neustart

Nach Anpassung der Konfiguration muss der Anwendungskontext (oder der gesamte Tomcat) neu gestartet werden. Dies kann beispielsweise über den Tomcat-Manager durchgeführt werden.

Reverse Proxy Umgebungen

Wird die API hinter einem Reverse Proxy zur Verfügung gestellt, sind spezielle Konfigurationen notwendig, um die internen Verlinkungen (`href`-Attribut einer Ressourcen-Darstellung) korrekt zu generieren.

Es wird hier auf die gängige Praxis bei Reverse Proxies zurückgegriffen, die spezielle Header (`X-Forwarded-For`, `X-Forwarded-Host`, etc.) definiert. Zusätzlich wird noch ein weiterer Header `X-Forwarded-ContextPath` eingeführt, der berücksichtigt, dass die Eventing API ggf. in dem Proxy auf einem anderen Context-Pfad hört als in dem Application Server (z.B. `/eventing-api` unter Apache HTTPD und `/eventing-webapp` unter Apache Tomcat).

Apache HTTP Beispielkonfiguration

Die folgende Definition kann z.B. in die `httpd.conf`-Datei integriert werden.

```
<Location "/apache-eventing/">
    ProxyPass "http://tomcat-server:8080/wv-eventing-webapp/"
    ProxyPassReverse "http://tomcat-server:8080/wv-eventing-
webapp/"
    Header add X-Forwarded-ContextPath "/apache-eventing"
    RequestHeader set X-Forwarded-ContextPath "/apache-
eventing"
    ProxyPreserveHost On
</Location>

# add ssl config
<IfModule ssl_module>
# Secure (SSL/TLS) connections
Include conf/httpd-ssl.conf
</IfModule>
```

Wichtig ist hier, dass die `Location` mit dem Wert des Headers (`Header add X-Forwarded-ContextPath ...` und `RequestHeader set X-Forwarded-ContextPath ...`) übereinstimmt.

Wird der Server auch via HTTPS angeboten muss in der entsprechenden Konfiguration (z.B. `httpd-ssl.conf`) ebenfalls eine `Location` definiert werden. Diese verfügt über einen zusätzlichen Header `X-Forwarded-Proto`:

```
# same location as for http, but with additional header
<Location "/apache-eventing/">
    ProxyPass "http://host.domain:port/webapp-name/"
    ProxyPassReverse "http://host.domain:port/webapp-name/"
    Header add X-Forwarded-ContextPath "/apache-eventing"
    RequestHeader set X-Forwarded-ContextPath "/apache-
eventing"
```

```
    RequestHeader set X-Forwarded-Proto "https"
    ProxyPreserveHost On
</Location>
```

Nginx Beispielkonfiguration

Die folgende Konfiguration kann z.B. in den **sites-enabled**-Ordner (z.B. Linux: **/etc/nginx/sites-enabled**) integriert werden.

```
upstream tomcat {
    server tomcat-server:8080 fail_timeout=0;
}

server {
    listen 80 default_server;

    listen [::]:80 default_server;
    client_max_body_size 10M;

    # uncomment for SSL support
    #listen 443 ssl;
    #ssl_certificate /etc/nginx/ssl/nginx.crt;
    #ssl_certificate_key /etc/nginx/ssl/nginx.key;

    root /var/www/html;

    index index.html index.htm index.nginx-debian.html;

    server_name nginx_server;

    location /nginx-eventing/ {
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-ContextPath "/nginx-
eventing";
        proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;
```

```
    proxy_set_header X-Forwarded-Host $host;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_redirect    off;
    proxy_connect_timeout 240;
    proxy_send_timeout  240;
    proxy_read_timeout  240;
    proxy_pass          http://tomcat/wv-eventing-webapp/;
}

}
```

Wichtig ist hier, dass die **location** mit dem Wert des Headers (**proxy_set_header X-Forwarded-ContextPath ...**) übereinstimmt.

API Kurzdokumentation

- **Basis-URL:** `http://host.domain:port/webapp-name/v1`

Der erste Teil (`http://host.domain:port/webapp-name`) ist abhängig von der Installation.

Für Collection-Ressourcen (z.B. `/v1/templates`) kann über den Query-Parameter `?expanded=true` die Ausgabe mit zusätzlichen Details versehen werden.

Lesender Zugriff

Die API kann über den Browser einfach genutzt werden, eine entsprechende Browser-Extension vorausgesetzt (z.B. JSONView für Chrome). Jede Ressource kommuniziert die Links zu Kind-Ressourcen oder anderen Ressourcen im bereitgestellten JSON (**href**-Attribut).

Paging

Dynamische Ressourcen, die eine Ergebnisliste zurückgeben, verfügen über Paging-Funktionalität. Dies kann per URL-Query-Parameter gesteuert werden. Beispiel:

```
http://host.domain:port/webapp-name/v1/events?  
limit=10&offset=20
```

Das resultierende JSON spiegelt diese Information wider und listet zusätzlich die Gesamtzahl der verfügbaren Einträge zurück. Beispiel:

```
{  
  "data": [  
    {  
      "eventType": {  
        "href": "http://host.domain:port/webapp-  
name/v1/eventTypes/4",  
        "id": "4"  
      },  
      "href": "http://host.domain:port/webapp-  
name/v1/events/2",  
      "id": "2",  
      "label": "Example Event",  
      "notificationLevel": {  
        "href": "http://host.domain:port/webapp-  
name/v1/notificationLevels/9",  
        "id": "9"  
      },  
      "publication": {  
        "href": "http://host.domain:port/webapp-  
name/v1/publications/1",  
        "id": "1"  
      },  
      "subscription": {  
        "href": "http://host.domain:port/webapp-  
name/v1/subscriptions/1",
```

```

        "id": "1"
      },
      "timestamp": 1509720300000,
      "timestampCreated": 1509721200000
    }
  ],
  "metadata": {
    "limit": 10,
    "offset": 1,
    "total": 2
  }
}

```

Abonnement Anlegen

HTTP POST (mit entsprechender Basic Authentication):

<http://host.domain:port/webapp-name/v1/subscriptions>

mit dem Payload:

```

{
  "publicationId": "1",
  "template": {
    "id": "2",
    "parameters": {
      "groupId": {
        "value": 1
      }
    }
  },
  "deliveryMethods": [ {
    "id": "wv-email",
    "parameters": {
    }
  } ],
  "enabled": true,
}

```



```
"endOfLife":"2018-06-19T13:22:08.248+02:00"
}
```

Die Werte für die verlinkten IDs können über die entsprechenden Ressourcen ermittelt werden:

- <http://host.domain:port/webapp-name/v1/publications>
- <http://host.domain:port/webapp-name/v1/templates>
- <http://host.domain:port/webapp-name/v1/groups>
- <http://host.domain:port/webapp-name/v1/users>
- <http://host.domain:port/webapp-name/v1/deliveryMethods>

Abonnement Entfernen

HTTP DELETE (mit entsprechender Basic Authentication):

<http://host.domain:port/webapp-name/v1/subscriptions/:id>

Notification (Regelgruppe) Anlegen

HTTP POST (mit entsprechender Basic Authentication):

<http://host.domain:port/webapp-name/v1/templates>

mit dem Payload:

```
{
  "definition": {
    "content": {
      "publication": 1,
      "rules": [
        {
          "threshold": 22.45,
          "trend": 11,
          "primaryRule": true,
          "level": 1
        }
      ]
    }
  }
}
```

```
    },  
    {  
      "threshold":22.45,  
      "trend":12,  
      "level": 9  
    }  
  ]  
}  
}
```

Die Werte für die verlinkten IDs können über die entsprechenden Ressourcen ermittelt werden:

- <http://host.domain:port/webapp-name/v1/publications>
- <http://host.domain:port/webapp-name/v1/trends>

Change Log

RC4 (2017-11-09)

- Anpassung an das aktualisierte Datenmodell (Subscription->Notification->Rule)
- Paging für alle Collection-Ressourcen
- Anpassung der verschiedenen menschenlesbaren Labels
- Integration der internen Verlinkungen (href-Attribut)
- Einführung des Query-Parameter `?expanded=true`
- Erstellung eines Test-Plans inkl. Postman-Beispiel-Requests
- Unterstützung für Reverse-Proxy-Umgebungen