

Java CDR Logger

About

CDRLogger is a Java servlet application that works together with [mod_xml_cdr](#) to store CDR records into a database table. The application parses the CDR XML data and stores it into pre-configured fields. Current application 'war' packaging only includes and tests with PostgreSQL database driver. Other database driver support could be easily included (MySQL, SQL Server, Oracle, etc).

Requirement

- Java Runtime Environment 1.5
- Java Servlet Container (Tomcat, JBoss, Weblogic, etc)
- Database Server: PostgreSQL (MySQL, Oracle, SQL Server, etc)

Installation w/ Tomcat

- Make sure JRE and Tomcat is successfully installed.
- Download cdr-logger war package from [here](#)
- Copy cdr-logger-x.x.war to tomcat web apps directory (replace x.x with the right version, ie. cdr-logger-1.0.war)

```
cp <your-download-path>/cdr-logger-x.x.war <tomcat-webapps-dir>
```

- If tomcat is running, the war package should automatically expanded into its own directory

Database Connection Setup

Edit cdr-logger-x.x/WEB-INF/classes/hibernate.conf.xml under tomcat web apps. Change the database connection setup according to your database settings. For example for postgresql:

```
<property name="connection.url">jdbc:postgresql://localhost/mydb</property>
<property name="connection.username">postgres</property>
<property name="connection.driver_class">org.postgresql.Driver</property>
<property name="dialect">org.hibernate.dialect.PostgreSQLDialect</property>
<property name="connection.password">yourpassword</property>
<property name="transaction.factory_class">org.hibernate.transaction.JDBCTransactionFactory</property>
```

Testing The Installation

Restart tomcat, under Ubuntu should be like

```
/etc/init.d/tomcat5.5 restart
```

point your browser to

```
http://yourip:port/cdr-logger-x.x/schema
```

It should return some error message similar to below

```
Unrecognized 'null' operation type. Type should be 'insert' or 'create'.
```

This indicate, the CDRLogger application properly installed.

Automatic Schema Generation

Now, to automatically generate database schema, point your browser to :

```
http://yourip:port/cdr-logger-x.x/schema?type=create
```

This will start automatic schema generation. Check your database, it should generate following database table :

```

CREATE TABLE freeswitch cdrs
(
  id bigint NOT NULL,
  start_stamp character varying(32),
  answer_stamp character varying(32),
  end_stamp character varying(32),
  profile_start_stamp character varying(32),
  answered_time character varying(32),
  hangup_time character varying(32),
  created_time character varying(32),
  billsec integer,
  billmsec integer,
  billusec integer,
  total_duration_sec integer,
  total_duration_msec integer,
  sip_from_uri character varying(64),
  sip_to_uri character varying(64),
  sip_call_id character varying(64),
  custom_tag_1 text,
  custom_tag_2 text,
  custom_tag_3 text,
  custom_tag_4 text,
  custom_tag_5 text,
  custom_tag_6 text,
  custom_tag_7 text,
  custom_tag_8 text,
  custom_tag_9 text,
  xml_data text,
  CONSTRAINT freeswitch_cdrs_pkey PRIMARY KEY (id)
)
WITH (OIDS=FALSE);
ALTER TABLE freeswitch_cdrs OWNER TO postgres;

CREATE TABLE freeswitch_cdr_data
(
  id bigint NOT NULL,
  category character varying(64),
  "name" character varying(64),
  "value" text,
  cdr_id bigint NOT NULL,
  CONSTRAINT freeswitch_cdr_data_pkey PRIMARY KEY (id),
  CONSTRAINT fkabdb2bd73aa16d74 FOREIGN KEY (cdr_id)
    REFERENCES freeswitch_cdrs (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (OIDS=FALSE);
ALTER TABLE freeswitch_cdr_data OWNER TO postgres;

```

You can test the application to insert test record, point your browser to :

<http://yourip:port/cdr-logger-x.x/schema?type=insert>

Then check freeswitch_cdrs table, it should create a sample record.

Configure mod_xml_cdr

Setup mod_xml_cdr to point to CDRLogger application.

```

<configuration name="xml_cdr.conf" description="XML CDR CURL logger">
<settings>
  <param name="url" value="http://yourip:port/cdr-logger-x.x/cdr"/>
  <param name="retries" value="2"/>
  <param name="delay" value="120"/>
  <param name="log-dir" value="/var/log/cdr"/>
  <param name="err-log-dir" value="/var/log/cdr/errors"/>
  <param name="encode" value="true"/>
</settings>

```

Make a test call and check the database table. if there is a new CDR record populated. Sit down and enjoy your coffee ;)

Additional Configuration

vi cdr-logger-x.x/WEB-INF/web.xml, there is a couple settings that you can tune. Don't forget to restart tomcat to make sure it pickup the latest changes.

- Turn on/off raw xml storage (storeRawXML)
- Customize xml tag storage (custom_tag_n)
- Turn on/off automatic schema generation url (allowChemaGenerator)

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://java.sun.com/xml/ns/j2ee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  version="2.4"
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
  <description>Freeswitch CDR Logger</description>
  <display-name>com.jkr.freeswitch.cdr.logger
  </display-name>
  <distributable />
  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
    <welcome-file>index.html</welcome-file>
  </welcome-file-list>
  <servlet>
    <servlet-name>CDRLogger</servlet-name>
    <servlet-class>com.jkr.freeswitch.cdr.logger.FreeswitchCDRLogger</servlet-class>
    <!-- Log raw cdr XML -->
    <init-param>
      <param-name>StoreRawXML</param-name>
      <param-value>>false</param-value>
    </init-param>
    <init-param>
      <param-name>custom_tag_1</param-name>
      <param-value>caller_id_number</param-value>
    </init-param>
    <init-param>
      <param-name>custom_tag_2</param-name>
      <param-value>caller_id_name</param-value>
    </init-param>
    <init-param>
      <param-name>custom_tag_3</param-name>
      <param-value>destination_number</param-value>
    </init-param>
    <init-param>
      <param-name>custom_tag_4</param-name>
      <param-value>uuid</param-value>
    </init-param>
    <init-param>
      <param-name>custom_tag_5</param-name>
      <param-value>chan_name</param-value>
    </init-param>
    <init-param>
      <param-name>custom_tag_6</param-name>
      <param-value></param-value>
    </init-param>
    <init-param>
      <param-name>custom_tag_7</param-name>
      <param-value></param-value>
    </init-param>
    <init-param>
      <param-name>custom_tag_8</param-name>
      <param-value></param-value>
    </init-param>
    <init-param>
      <param-name>custom_tag_9</param-name>
      <param-value></param-value>
    </init-param>
  </servlet>
  <servlet>
    <servlet-name>CDRSchema</servlet-name>
    <servlet-class>com.jkr.freeswitch.cdr.logger.FreeswitchCDRSchema</servlet-class>
    <!-- Log raw cdr XML -->
    <init-param>
      <param-name>allowShemaGenerator</param-name>
      <param-value>>true</param-value>
    </init-param>
  </servlet>
  <servlet-mapping>
    <servlet-name>CDRLogger</servlet-name>
    <url-pattern>/cdr</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
    <servlet-name>CDRSchema</servlet-name>
    <url-pattern>/schema</url-pattern>
  </servlet-mapping>
</web-app>

```