

配置_StatFilter

irfen edited this page on 8 Jul · 26 revisions

Druid内置提供一个StatFilter，用于统计监控信息。

1. 别名配置

StatFilter的别名是stat，这个别名映射配置信息保存在druid-xxx.jar!/META-INF/druid-filter.properties。

在spring中使用别名配置方式如下：

```
<bean id="dataSource" class="com.alibaba.druid.pool.DruidDataSource" init-method="init" destroy-method="close">
    ...
    <property name="filters" value="stat" />
</bean>
```

2. 组合配置

StatFilter可以和其他的Filter配置使用，比如：

```
<bean id="dataSource" class="com.alibaba.druid.pool.DruidDataSource" init-method="init" destroy-method="close">
    ...
    <property name="filters" value="stat,log4j" />
</bean>
```

在上面的配置中，StatFilter和Log4jFilter组合使用。

3. 通过proxyFilters属性配置

别名配置是通过filters属性配置的，filters属性的类型是String。如果需要通过bean的方式配置，使用proxyFilters属性。

```
<bean id="stat-filter" class="com.alibaba.druid.filter.stat.StatFilter">
    <property name="slowSqlMillis" value="10000" />
    <property name="logSlowSql" value="true" />
</bean>
```

```
</bean>
<bean id="dataSource" class="com.alibaba.druid.pool.DruidDataSource"
    init-method="init" destroy-method="close">
    ... ..
    <property name="filters" value="log4j" />
    <property name="proxyFilters">
        <list>
            <ref bean="stat-filter" />
        </list>
    </property>
</bean>
```

其中filters和proxyFilters属性是组合关系的，不是替换的，在上面的配置中，dataSource有了两个Filter，StatFilter和Log4jFilter。

4. SQL合并配置

当你程序中存在没有参数化的sql执行时，sql统计的效果会不好。比如：

```
select * from t where id = 1
select * from t where id = 2
select * from t where id = 3
```

在统计中，显示为3条sql，这不是我们希望要的效果。StatFilter提供合并的功能，能够将这3个SQL合并为如下的SQL

```
select * from t where id = ?
```

配置StatFilter的mergeSql属性

```
<bean id="stat-filter" class="com.alibaba.druid.filter.stat.StatFilter">
    <property name="mergeSql" value="true" />
</bean>
<bean id="dataSource" class="com.alibaba.druid.pool.DruidDataSource"
    init-method="init" destroy-method="close">
    ... ..
    <property name="proxyFilters">
        <list>
            <ref bean="stat-filter" />
        </list>
    </property>
```

```
</bean>
```

StatFilter支持一种简化配置方式，和上面的配置等同的。如下：

```
<bean id="dataSource" class="com.alibaba.druid.pool.DruidDataSource" init-method="init" destroy-method="close">
    ...
    <property name="filters" value="mergeStat" />
</bean>
```

mergeStat是MergeStatFilter缩写，我们看MergeStatFilter的实现：

```
public class MergeStatFilter extends StatFilter {
    public MergeStatFilter() {
        super.setMergeSql(true);
    }
}
```

从实现代码来看，仅仅是一个mergeSql的缺省值。

也可以通过connectProperties属性来打开mergeSql功能，例如：

```
<bean id="dataSource" class="com.alibaba.druid.pool.DruidDataSource" init-method="init" destroy-method="close">
    ...
    <property name="filters" value="stat" />
    <property name="connectionProperties" value="druid.stat.mergeSql=true" />
</bean>
```

或者通过增加JVM的参数配置：

```
-Ddruid.stat.mergeSql=true
```

4.1 合并SQL对tddl的支持

在druid-0.2.17版本之后，sql合并支持tddl，能够对分表进行合并。

5. 慢SQL记录

StatFilter属性slowSqlMillis用来配置SQL慢的标准，执行时间超过slowSqlMillis的就是慢。slowSqlMillis的缺省值为3000，也就是3秒。

```
<bean id="stat-filter" class="com.alibaba.druid.filter.stat.StatFilter">
  <property name="slowSqlMillis" value="10000" />
  <property name="logSlowSql" value="true" />
</bean>
```

在上面的配置中，slowSqlMillis被修改为10秒，并且通过日志输出执行慢的SQL。

slowSqlMillis属性也可以通过connectProperties来配置，例如：

```
<bean id="dataSource" class="com.alibaba.druid.pool.DruidDataSource" init-method="init"
...
  <property name="filters" value="stat" />
  <property name="connectionProperties" value="druid.stat.slowSqlMillis=5000" />
</bean>
```

6. 合并多个DruidDataSource的监控数据

缺省多个DruidDataSource的监控数据是各自独立的，在Druid-0.2.17版本之后，支持配置公用监控数据，配置参数为useGlobalDataSourceStat。例如：

```
<bean id="dataSource" class="com.alibaba.druid.pool.DruidDataSource" init-method="init"
...
  <property name="useGlobalDataSourceStat" value="true" />
</bean>
```

或者通过jvm启动参数来指定，例如：

```
-Ddruid.useGlobalDataSourceStat=true
```

全部使用jvm启动参数来配置，可以这样：

```
-Ddruid.filters=mergeStat -Ddruid.useGlobalDataSourceStat=true
```