链滴

spring boot 实现分库分表 (yml 方式)

作者：zousiliang
原文链接：https://ld246.com/article/1567322304545
来源网站：链滴

spring boot 实现分库分表 (yml方式)

数据源分片策略使用单独yml文件配置，通过Java config方式定义数据源，这种方式比较推荐

源码地址:https://github.com/zousiliang/fast/tree/tag5.0

添加数据：

http://localhost/curPrice/add?createdBy=1&luck=1

查看数据

http://localhost/curPrice/all

创建两个数据库 并同时创建两个表

```
DROP TABLE IF EXISTS `t_cut_price0`;
CREATE TABLE `t_cut_price0` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `luck` varchar(50) DEFAULT NULL COMMENT '幸运值',
  `created_by` varchar(255) DEFAULT NULL COMMENT '创建人',
  `created_date` datetime DEFAULT NULL  COMMENT '创建时间',
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;


DROP TABLE IF EXISTS `t_cut_price1`;
CREATE TABLE `t_cut_price1` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `luck` varchar(50) DEFAULT NULL COMMENT '幸运值',
  `created_by` varchar(255) DEFAULT NULL COMMENT '创建人',
  `created_date` datetime DEFAULT NULL  COMMENT '创建时间',
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

pom.xml:

```
<dependency>
    <groupId>io.shardingjdbc</groupId>
    <artifactId>sharding-jdbc-core</artifactId>
    <version>2.0.3</version>
</dependency>
```

DataSourceConfig文件:

```
@Configuration
public class DataSourceConfig {
```

```
    @Autowired
    private Filter statFilter;

    private static final String SHARDING_YML_PATH = "dataSource.yml";

    /**
     * 构建dataSource
     * 这里没有使用ShardingDataSourceFactory
     * 因为要为durid数据源配置监听Filter
     * @return
     * @throws SQLException
     * @throws IOException
     */
    @Bean
    public DataSource dataSource() throws SQLException, IOException {
        YamlShardingConfiguration config = parse();
        ShardingRule rule = config.getShardingRule(Collections.<String, DataSource>emptyMap
));
        rule.getDataSourceMap().forEach((k,v)->{
            DruidDataSource d = (DruidDataSource) v;
            d.setProxyFilters(Lists.newArrayList(statFilter));
        });
        return new ShardingDataSource(rule, config.getShardingRule().getConfigMap(), config.g
tShardingRule().getProps());
    }

    /**
     * 解析yml
     * @return
     * @throws IOException
     * @throws FileNotFoundException
     * @throws UnsupportedEncodingException
     */
    private YamlShardingConfiguration parse() throws IOException, FileNotFoundException, Un
upportedEncodingException {
        Resource certResource = new ClassPathResource(SHARDING_YML_PATH);
        try (
          InputStreamReader inputStreamReader = new InputStreamReader(certResource.getIn
utStream(), "UTF-8")
        ) {
          return new Yaml(new Constructor(YamlShardingConfiguration.class)).loadAs(inputStre
mReader, YamlShardingConfiguration.class);
        }
    }
}
```