```
package com.fr.stable.fun;

import com.fr.stable.fun.mark.Immutable;
import com.fr.stable.pool.DataSourceAdapter;

/**
 * Created by daniel on 2017/5/4.
 *
 */
public interface DataSourceProcessor extends Immutable {


    String XML_TAG = "DataSourceProcessor";

    int CURRENT_LEVEL = 1;

    DataSourceAdapter createDataSource();

}
```

:

```
<extra-core>
    <DataSourceProcessor class="com.fr.plugin.****.DataSourceProcessor"/>
</extra-core>
```

DataSourceAdapter

`DataSourceAdapter`

```
package com.fr.stable.pool;

import javax.sql.DataSource;
import java.sql.SQLException;

/**
 * Created by daniel on 2017/5/4.
 *
 */
public interface DataSourceAdapter {

    void setDriverClassName(String driverClassName);

    void setUrl(String url);

    void setUsername(String username);

    void setPassword(String password);

    void setInitialSize(int initialSize);

    void setMaxActive(int maxActive);

    void setMaxIdle(int maxIdle);

    void setMinIdle(int minIdle);

    void setMaxWait(long maxWait);

    void setTestOnBorrow(boolean testOnBorrow);

    void setTestOnReturn(boolean testOnReturn);

    void setTestWhileIdle(boolean testWhileIdle);

    void setMinEvictableIdleTimeMillis(long minEvictableIdleTimeMillis);

    void setTimeBetweenEvictionRunsMillis(long timeBetweenEvictionRunsMillis);

    void setNumTestsPerEvictionRun(int numTestsPerEvictionRun);

    void setValidationQuery(String validationQuery);

    /*
    *
     */
    void close() throws SQLException;

    /**
     * datasource
     * @return
     */
    DataSource get();

    String getDriverClassName();

    int getMaxActive();

    int getMaxIdle();

    int getNumActive();

    int getNumIdle();
}
```

DBCPAdapter

```
package com.fr.data.pool;

import com.fr.stable.pool.AbstractDataSourceAdapter;
import com.fr.third.org.apache.commons.dbcp.BasicDataSource;

import java.sql.SQLException;

/**
 * Created by daniel on 2017/5/4.
 */
public class DBCPDataSource extends AbstractDataSourceAdapter {

    private BasicDataSource dataSource;

    public DBCPDataSource(){
        this.dataSource = new BasicDataSource();
    }

    @Override
    public void setDriverClassName(String driverClassName) {
        get().setDriverClassName(driverClassName);
    }

    @Override
    public void setUrl(String url) {
        get().setUrl(url);
    }

    @Override
    public void setUsername(String username) {
        get().setUsername(username);
    }

    @Override
    public void setPassword(String password) {
        get().setPassword(password);
    }

    @Override
    public void setInitialSize(int initialSize) {
        get().setInitialSize(initialSize);
    }

    @Override
    public void setMaxActive(int maxActive) {
        get().setMaxActive(maxActive);
    }

    @Override
    public void setMaxIdle(int maxIdle) {
        get().setMaxIdle(maxIdle);
    }

    @Override
    public void setMinIdle(int minIdle) {
        get().setMinIdle(minIdle);
    }

    @Override
    public void setMaxWait(long maxWait) {
        get().setMaxWait(maxWait);
    }

    @Override
    public void setTestOnBorrow(boolean testOnBorrow) {
        get().setTestOnBorrow(testOnBorrow);
    }

    @Override
    public void setTestOnReturn(boolean testOnReturn) {
        get().setTestOnReturn(testOnReturn);
```

```
    }

    @Override
    public void setTestWhileIdle(boolean testWhileIdle) {
        get().setTestWhileIdle(testWhileIdle);
    }

    @Override
    public void setMinEvictableIdleTimeMillis(long minEvictableIdleTimeMillis) {
        get().setMinEvictableIdleTimeMillis(minEvictableIdleTimeMillis);
    }

    @Override
    public void setTimeBetweenEvictionRunsMillis(long timeBetweenEvictionRunsMillis) {
        get().setTimeBetweenEvictionRunsMillis(timeBetweenEvictionRunsMillis);
    }

    @Override
    public void setNumTestsPerEvictionRun(int numTestsPerEvictionRun) {
        get().setNumTestsPerEvictionRun(numTestsPerEvictionRun);
    }

    @Override
    public void setValidationQuery(String validationQuery) {
        get().setValidationQuery(validationQuery);
    }

    @Override
    public void close() throws SQLException {
        get().close();
    }

    @Override
    public BasicDataSource get() {
        return dataSource;
    }

    @Override
    public String getDriverClassName() {
        return get().getDriverClassName();
    }

    @Override
    public int getMaxActive() {
        return get().getMaxActive();
    }

    @Override
    public int getMaxIdle() {
        return get().getMaxIdle();
    }

    @Override
    public int getNumActive() {
        return get().getNumActive();
    }

    @Override
    public int getNumIdle() {
        return get().getNumIdle();
    }
}
```

DataSource

AbstractDataSourceAdapter

```
@Override
public void setMaxIdle(int maxIdle) {
    //doNothing druid
   // get().setMaxIdle(maxIdle);
}
```