# Graduation Project
# PRMA
# Weekly Report

26/10/2014 – 30/10/2014

Cai Bowen

# Progress

Scheduled:

1. Finish transaction management
2. Add caching in application layer (DAO layer).
3. Add filtering layer in storage, avoid irrelevant data.
4. Test store and transaction.
5. Benchmark storage

Progress:

1. Done.
2. Have created two kind of cache: full hash based cache and LIRS cache, but have not integrated into data layer.
3. Created filters, not yet integrated.
4. Done (without cache or filter).
5. pending

# 1. JDBC Operations

I wrote a mini yet robust JDBC framework, and integrated it into one of my previous project *Gplume*.

This framework borrowed ideas from *springframework-jdbc* and *springframework-transaction.*

*Gplume-jdbc* provide support for easy JDBC operation and declarative transaction management.

URL: Gplume-jdbc

https://github.com/xkommando/Gplume/tree/master/gplume/src/main/java/com/caibowen/gplume/jdbc

# 1. JDBC Operations

## Example :

Query :

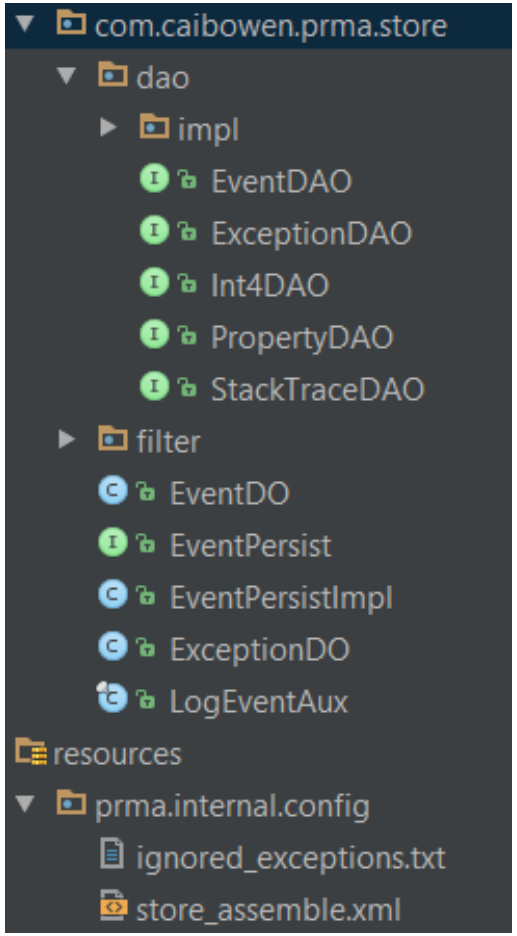
```java
@Nullable
@Override
public String get(final int key) {
    return queryForObject(new StatementCreator() {
        @Override
        public PreparedStatement createStatement(Connection con) throws SQLException {
            PreparedStatement ps = con.prepareStatement(
                    "SELECT `value` FROM " + tableName + " WHERE id=" + key + " LIMIT 1");
            return ps;
        }
    }, RowMapping.STR_ROW_MAPPING);
}
```

transaction:

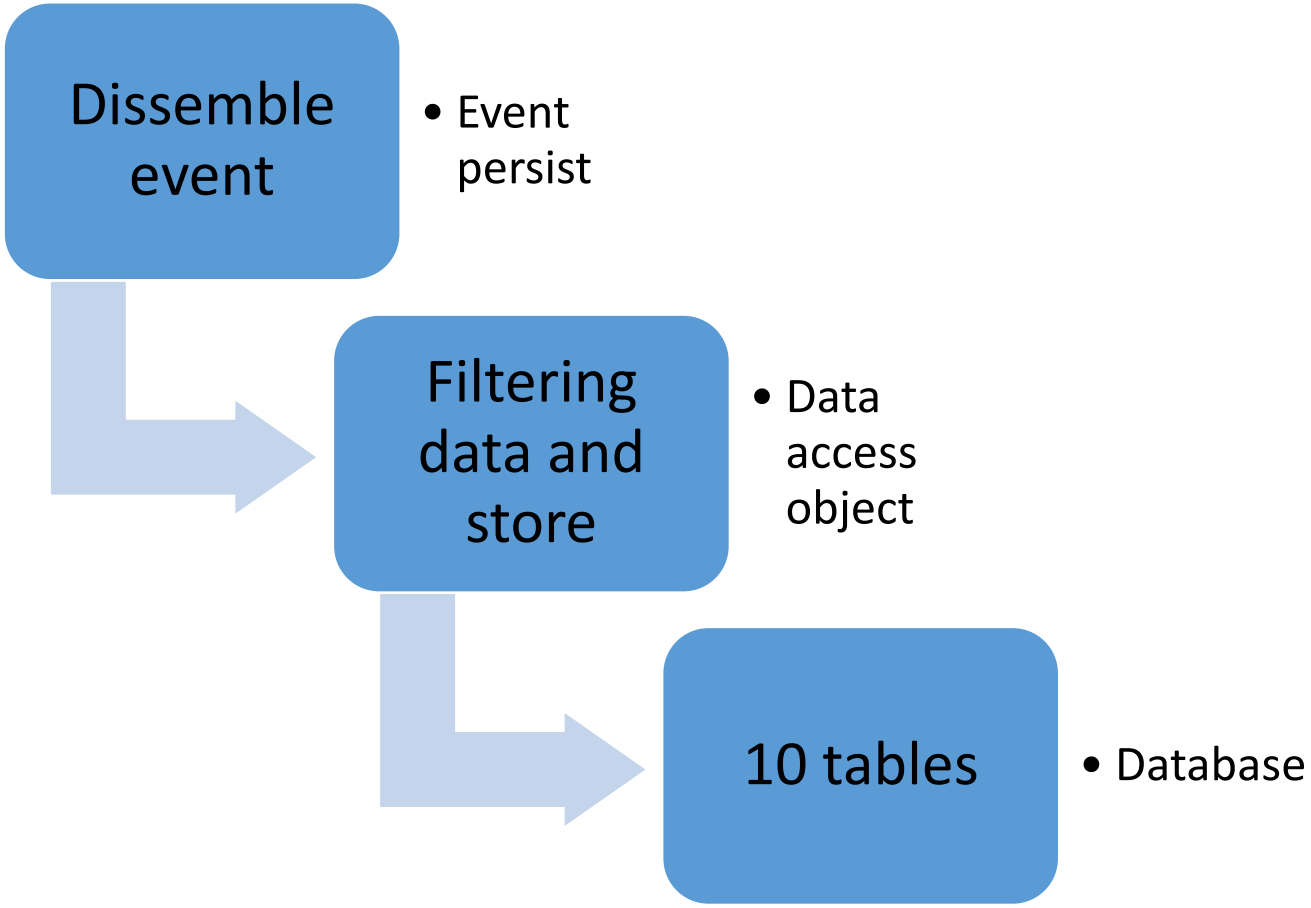
```java
eventDAO.execute(new TransactionCallback<Object>() {
    @Override
    public Object withTransaction(@Nonnull Transaction tnx) throws Exception {
        // operation 1
        // operation 2
        eventDAO.execute(new TransactionCallback<Object>() {
            @Override
            public Object withTransaction(@Nonnull Transaction tnx) throws Exception {
                // nested transaction is isolated.
                return null;
            }
        });
        // fail on any of the operations will trigger rollback automatically
        return null;
    }
});
```

# 2. Storage Layer design

## *Log event store in PRMA*




Jdbc operates are managed in one transaction



# 2. Storage Layer design

All components in PRMA are configurated by XML

and are assembled by Gplume XMLBeanAssembler.

```
<bean id="exceptNameDAO" class="com.caibowen.prma.store.dao.impl.StrDAOImple">
    <construct value="`exception_name`"/>
    <property name="dataSource" ref="dataSource"/>
</bean>
<bean id="exceptMsgDAO" class="com.caibowen.prma.store.dao.impl.cao.Int4FullCAO">
    <property name="db">
        <bean id="exceptMsgDAO" class="com.caibowen.prma.store.dao.impl.StrDAOImple">
            <construct value="`exception_msg`"/>
            <property name="dataSource" ref="dataSource"/>
        </bean>
    </property>
</bean>
```

```
<bean id="eventPersist" class="com.caibowen.prma.store.EventPersistImpl">
    <property name="threadDAO" ref="threadDAO"/>
    <property name="loggerDAO" ref="loggerDAO"/>
    <property name="stackTraceDAO" ref="stackTraceDAO"/>
    <property name="propertyDAO" ref="propertyDAO"/>
    <property name="eventDAO" ref="eventDAO"/>
    <property name="exceptionDAO" ref="exceptionDAO"/>
</bean>
```

# 3. Next

1. Integrate cache to data layer to reduce database operation, promoting insertion, as well as querying speed.
2. Add filters to data layer, filter out irrelevant information before stored in DB
3. Design monitor