

xmlxml

XMLFileManagerProvider

FineReport

```
com.fr.file.XMLFileManager
```

WEB-INF/resourcesxmlFineReportAPI

ExpirationConfigManagerExpirationConfigManagerProvider

```
public interface ExpirationConfigManagerProvider extends RemoteXMLFileManagerProvider, java.io.Serializable,
    FCloneable {

    int getValidMonthCount();

    void setValidMonthCount(int validMonthCount);

    int getWarningDayCount();

    void setWarningDayCount(int count);
}
```

ExpirationConfigManagerProvider

```
public class ExpirationConfigManager extends XMLFileManager implements ExpirationConfigManagerProvider {

    private static final String XML_TAG = "ExpirationConfigManager";

    private static ExpirationConfigManagerProvider manager = null;

    public synchronized static ExpirationConfigManager getInstance() {
        return (ExpirationConfigManager) getProviderInstance();
    }

    public synchronized static ExpirationConfigManagerProvider getProviderInstance() {
        if (manager == null) {
            if (isClusterMember()) {
                return manager;
            }
            manager.readXMLFile();
        }
        return manager;
    }
}
```

```

private synchronized static boolean isClusterMember() {
    switch (BaseClusterHelper.getClusterState()) {
        case LEADER:
            manager = new ExpirationConfigManager();
            RPC.registerSkeleton(manager);
            return false;
        case MEMBER:
            String ip = BaseClusterHelper.getMainServiceIP();
            manager = (ExpirationConfigManagerProvider) RPC.getProxy(ExpirationConfigManager.class, ip);
            return true;
        default:
            manager = new ExpirationConfigManager();
            break;
    }
}

return false;
}

static {
    GeneralContext.addEnvChangedListener(new EnvChangedListener() {
        public void envChanged() {
            ExpirationConfigManager.envChanged();
        }
    });
}

private synchronized static void envChanged() {
    manager = null;
}

private int validMonthCount = 1;
private int warningDayCount = 6;

public int getValidMonthCount() {
    return validMonthCount;
}

public void setValidMonthCount(int validMonthCount) {
    this.validMonthCount = validMonthCount;
}

@Override
public int getWarningDayCount() {
    return warningDayCount;
}

@Override
public void setWarningDayCount(int count) {
    warningDayCount = count;
}

@Override
public String fileName() {
    return "expiration.xml";
}

@Override
public void readXML(XMLableReader reader) {
    if (reader.isAttr()) {
        validMonthCount = reader.getAttrAsInt("validMonthCount", 1);
        warningDayCount = reader.getAttrAsInt("warningDayCount", 6);
    }
}

@Override
public void writeXML(XMLPrintWriter writer) {
    writer.startTAG(XML_TAG);
}

```

```

        writer.attr("validMonthCount", validMonthCount);
        writer.attr("warningDayCount", warningDayCount);
        writer.end();
    }

    @Override
    public boolean writeResource() throws Exception {
        return GeneralContext.getEnvProvider().writeResource(ExpirationConfigManager.getProviderInstance());
    }

    public Object clone() throws CloneNotSupportedException {
        return super.clone();
    }
}

```

```
int count = ExpirationConfigManager.getProviderInstance().getWarningDayCount();
```

xml

```

ExpirationConfigManagerProvider configManager = ExpirationConfigManager.getProviderInstance();
configManager.setWarningDayCount(warningCount);
configManager.setValidMonthCount(month);
FRContext.getCurrentEnv().writeResource(configManager);

```

readFromInputStreamExpirationConfigManager

```

//
ExpirationConfigManager manager = new ExpirationConfigManager();
// inputstream
XMLTools.readInputStreamXML(manager, input);
//
expirationConfigManager= manager;
// xml
GeneralContext.getEnvProvider().writeResource(datasourceManager);

```

```

<extra-core>
  <XMLFileManagerProvider class="com.fr.plugin.demo.ExpirationConfigManager">
</extra-core>

```

webplugin.xml

```
<extra-core>  
  <XMLFileManagerProvider class="com.fr.plugin.demo.ExpirationConfigManager">  
</extra-core>  
<move-after-install>  
  <File name="myconfig.xml" dir="/resources" operation="0"/>  
</move-after-install>
```

myconfig.xml|WEB-INF/resources|dir