

com.fr.io.collection.ExportCollection

-
- [FineReport V10 2018-07-30](#)
-
-
-

//——FineReport V10 2018-07-30

ExportCollection.java

```
package com.fr.io.collection;

import com.fr.common.annotations.Open;
import com.fr.general.DeclareRecordType;
import com.fr.general.RecordType;
import com.fr.io.exporter.AppExporter;
import com.fr.log.FineLoggerFactory;
import com.fr.log.LogUtils;
import com.fr.web.core.ErrorHandlerHelper;
import com.fr.web.core.ReportRepositoryDeal;
import com.fr.web.core.ReportSessionIDInfor;
import com.fr.web.core.utils.ExportUtils;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.io.OutputStream;

/**
 * Created by richie on 15/11/30.
 *
 */
@Open
public class ExportCollection {
    private AppExporter exporter;
    private RecordType recordType;

    public static ExportCollection create() {
        return new ExportCollection();
    }

    private ExportCollection() {
    }

    public AppExporter getExporter() {
        return exporter;
    }

    public void setExporter(AppExporter exporter) {
        this.exporter = exporter;
    }

    public RecordType getRecordType() {
        return recordType;
    }

    public void setRecordType(RecordType recordType) {
        this.recordType = recordType;
    }
}
```

```

    *
    *
    */
    @Deprecated
    public void setRecordType(DeclareRecordType recordType) {
        this.recordType = recordType;
    }

    public void doExport(HttpServletRequest req, HttpServletResponse res, ReportSessionIDInfor sessionIDInfor,
String format)
        throws Exception {

        if (getRecordType() == null) {
            ErrorHandlerHelper.getErrorHandler().error(req, res,
                "Cannot recognize the specified export format:" + format +
                ",\n\nThe correct format can be PDF,Excel,Word,SVG,CSV,Text or Image.");
            return;
        }

        OutputStream outputStream = res.getOutputStream();
        try {
            //
            LogUtils.exportAndLogRecordType(exporter, outputStream, new ReportRepositoryDeal(req,
sessionIDInfor), recordType);
        } catch (Exception e) {
            ExportUtils.resetContent(res);
            ErrorHandlerHelper.getErrorHandler().error(req, res, e);
            FineLoggerFactory.getLogger().error(e.getMessage(), e);
        }
        try {
            outputStream.flush();
            outputStream.close();
        } catch (IOException ignored) {
        }
    }
}

```

//

ExportCollection

AppExporterAppExporter

RecordTypeRecordTypeDeclareRecordType

RecordTypegetTypeMark20000getTypeString plugin_ getProductType

[com.fr.report.fun.ExportOperateProvider](#)

[com.fr.report.fun.ExportExtensionProcessor](#)