

# ExportHandleProvider

## ExportHandleProvider

```
package com.finebi.dashboard.impl.service.export;

import com.finebi.dashboard.api.service.export.ExportContext;
import com.fr.stable.fun.mark.Mutable;

import java.io.OutputStream;

/**
 *
 * Created by Hiram on 2018/11/14.
 */
public interface ExportHandleProvider extends Mutable {
    String XML_TAG = "ExportHandleProvider";

    int CURRENT_LEVEL = 1;

    /**
     *
     * @param originalOutputStream
     * @param type
     * @return
     */
    OutputStream handleStream(OutputStream originalOutputStream, ExportType type);

    /**
     *
     * @param originalOutputStream
     * @param exportContext
     * @return
     */
    OutputStream handleStream(OutputStream originalOutputStream, ExportContext exportContext);
}
```

```
<extra-core>
  <ExportHandleProvider class="" />
</extra-core>
```

## AbstractExportHandleProvider

```
package com.finebi.provider.api.export;

import com.finebi.dashboard.impl.service.export.ExportHandleProvider;
import com.finebi.dashboard.impl.service.export.ExportType;
import com.fr.stable.fun.mark.API;

import java.io.OutputStream;

/**
 * excel
 * Created by Hiram on 2018/11/15.
 */
@API(level = ExportHandleProvider.CURRENT_LEVEL)
public abstract class AbstractExportHandleProvider implements ExportHandleProvider {

    @Override
    public int currentAPILevel() {
        return CURRENT_LEVEL;
    }

    @Override
    public String mark4Provider() {
        return getClass().getName();
    }

    @Override
    @Deprecated
    public OutputStream handleStream(OutputStream originalOutputStream, ExportType type) {
        return originalOutputStream;
    }
}
```

## ExportHandleProvider

```
package com.finebi.stable.fun;

import com.fr.stable.fun.mark.Mutable;

import java.io.OutputStream;

/**
 * Created by Hiram on 2018/11/14.
 */
public interface ExportHandleProvider extends Mutable {
    String XML_TAG = "ExportHandleProvider";

    int CURRENT_LEVEL = 1;

    /**
     *
     * @param originalOutputStream
     * @param type
     * @return
     */
    OutputStream handleStream(OutputStream originalOutputStream, ExportType type);
}
```

```
<extra-core>
  <ExportHandleProvider class="com.fr.plugin.demo.export.CountExportHandle" />
</extra-core>
```

CountExportHandleCountOutputStreamwriteclose

## blocked URL

### CountExportHandle

```
import com.finebi.stable.fun.ExportType;
import com.finebi.stable.fun.impl.AbstractExportHandleProvider;
import java.io.OutputStream;
public class CountExportHandle extends AbstractExportHandleProvider {
    @Override
    public OutputStream handleStream(OutputStream originalOutputStream, ExportType type) {
        return new CountOutputStream(originalOutputStream);
    }
}
```

## CountOutputStream

```
import com.fr.log.FineLoggerFactory;
import java.io.IOException;
import java.io.OutputStream;

public class CountOutputStream extends OutputStream {

    private OutputStream out;

    private int count;

    public CountOutputStream(OutputStream out) {
        this.out = out;
    }

    @Override
    public void write(int b) throws IOException {
        count++;
        out.write(b);
    }

    @Override
    public void write(byte[] b) throws IOException {
        count += b.length;
        out.write(b);
    }

    @Override
    public void write(byte[] b, int off, int len) throws IOException {
        count += len;
        out.write(b, off, len);
    }

    @Override
    public void flush() throws IOException {
        out.flush();
    }

    @Override
    public void close() throws IOException {
        FineLoggerFactory.getLogger().info("==== export length: {} =====", count);
        out.close();
    }
}
```