

com.fr.web.struct.Atom

-
- //—FineReport V10 2018-11-08
- //
-
-

//—FineReport V10 2018-11-08

Atom.java

```
package com.fr.web.struct;

import com.fr.common.annotations.Open;
import com.fr.web.struct.browser.RequestClient;
import com.fr.web.struct.category.ScriptPath;
import com.fr.web.struct.category.StylePath;

/**
 * web
 */
@Open
public interface Atom {

    /**
     * javascript
     *
     * @return
     */
    @Deprecated
    ScriptPath script();

    /**
     * css
     *
     * @return css
     */
    @Deprecated
    StylePath style();

    /**
     * javascript
     *
     * @param req
     * @return
     */
    ScriptPath script(RequestClient req);

    /**
     * css
     *
     * @param req
     * @return css
     */
    StylePath style(RequestClient req);

    /**
     * web
     * @return
     */
    Filter filter();
}
```

Component.java

```
package com.fr.web.struct;

import com.fr.common.annotations.Open;
import com.fr.web.struct.browser.RequestClient;
import com.fr.web.struct.category.ScriptPath;
import com.fr.web.struct.category.StylePath;

/**
 *
 */
@Open
public abstract class Component implements Atom {

    @Override
    @Deprecated
    public ScriptPath script() {
        return ScriptPath.EMPTY;
    }

    @Override
    @Deprecated
    public StylePath style() {
        return StylePath.EMPTY;
    }

    @Override
    public ScriptPath script(RequestClient req) {
        return script();
    }

    @Override
    public StylePath style(RequestClient req) {
        return style();
    }

    @Override
    public Filter filter() {
        return null;
    }
}
```

AssembleComponent.java

```
package com.fr.web.struct;

import com.fr.common.annotations.Open;

/**
 *
 */
@Open
public abstract class AssembleComponent extends Component {

    /**
     *
     * @return
     */
    public abstract Atom[] refer();

    /**
     *
     * @return
     */
    public Atom[] children() {
        return new Atom[0];
    }
}
```

Filter.java

```
package com.fr.web.struct;

import com.fr.common.annotations.Open;

@Open
public interface Filter {

    boolean accept();

}
```

//

@Open3

ScriptPath script(); StylePath style();

ScriptPath script(RequestClient req) [RequestClient](#) ScriptPath JS

ScriptPath5

ScriptPath.build(String path, FileType fileType, ParserType parserType);

ScriptPath.build(String path, FileType fileType); ScriptPath.build(String path, FileType fileType, ParserType.PLAINE);

ScriptPath.build(String path, ParserType parserType); ScriptPath.build(String path, FileType.PLAINE, ParserType parserType);

ScriptPath.build(String path); ScriptPath.build(String path, FileType.PLAINE, ParserType.PLAINE);

FileTypeParserType

FileType.PLAIN	PLAINpathcom/fr/plugin/js/main.js
FileType.CLASS	calssclassclasscom.fr.gen.TextGeneratorjsCLASSpathclasscom.fr.plugin.js.JsMaker

FileTypeJS \${FineReport} JS

ParserType.PLAIN	JS
ParserType.DYNAMIC	\${}JS

StylePath style(RequestClient req) ScriptPath script(RequestClient req) ScriptPathScriptPathJSStylePath

Filter filter()webJS/

AtomComponentAssembleComponent

webAssembleComponent