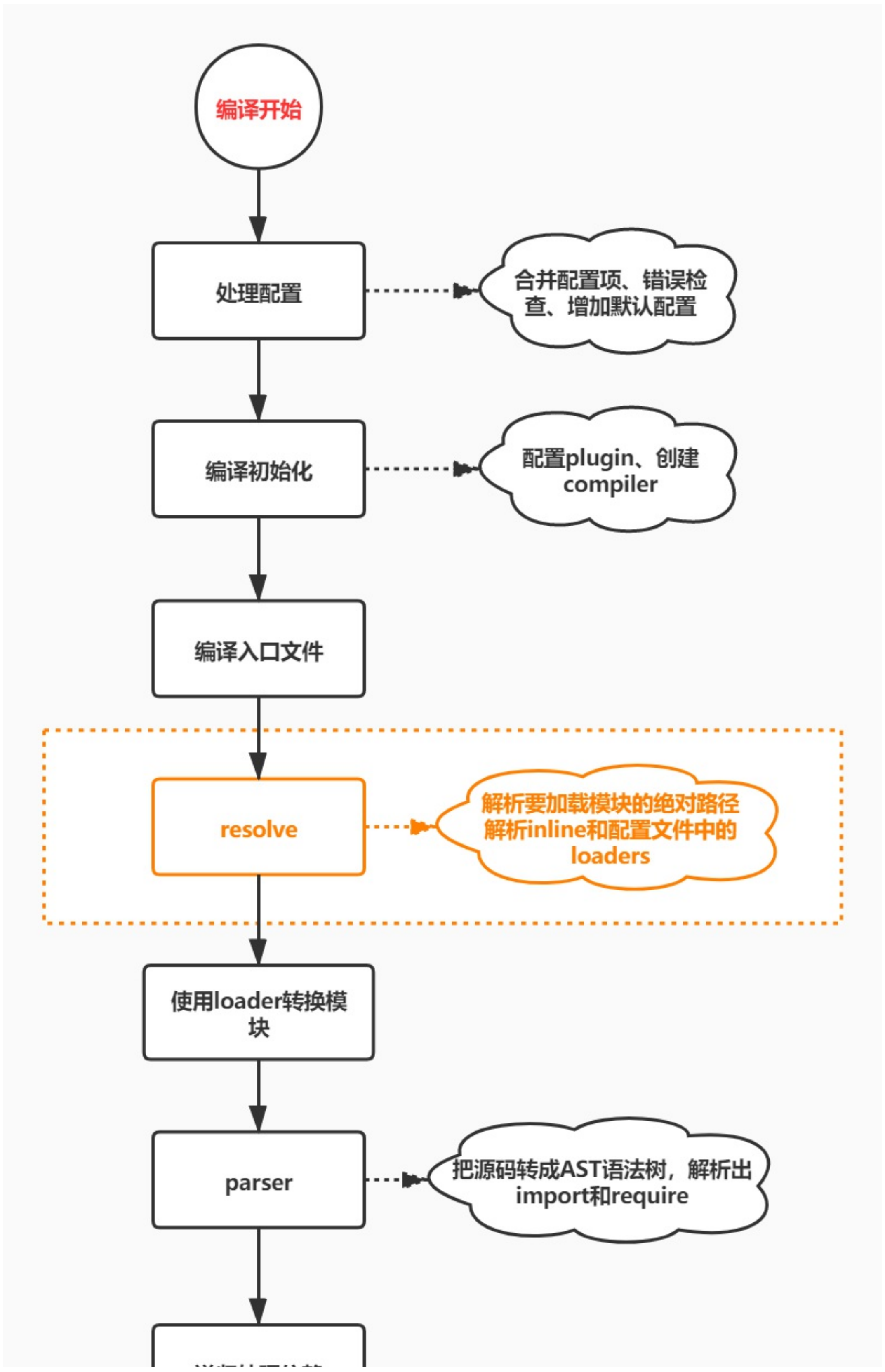
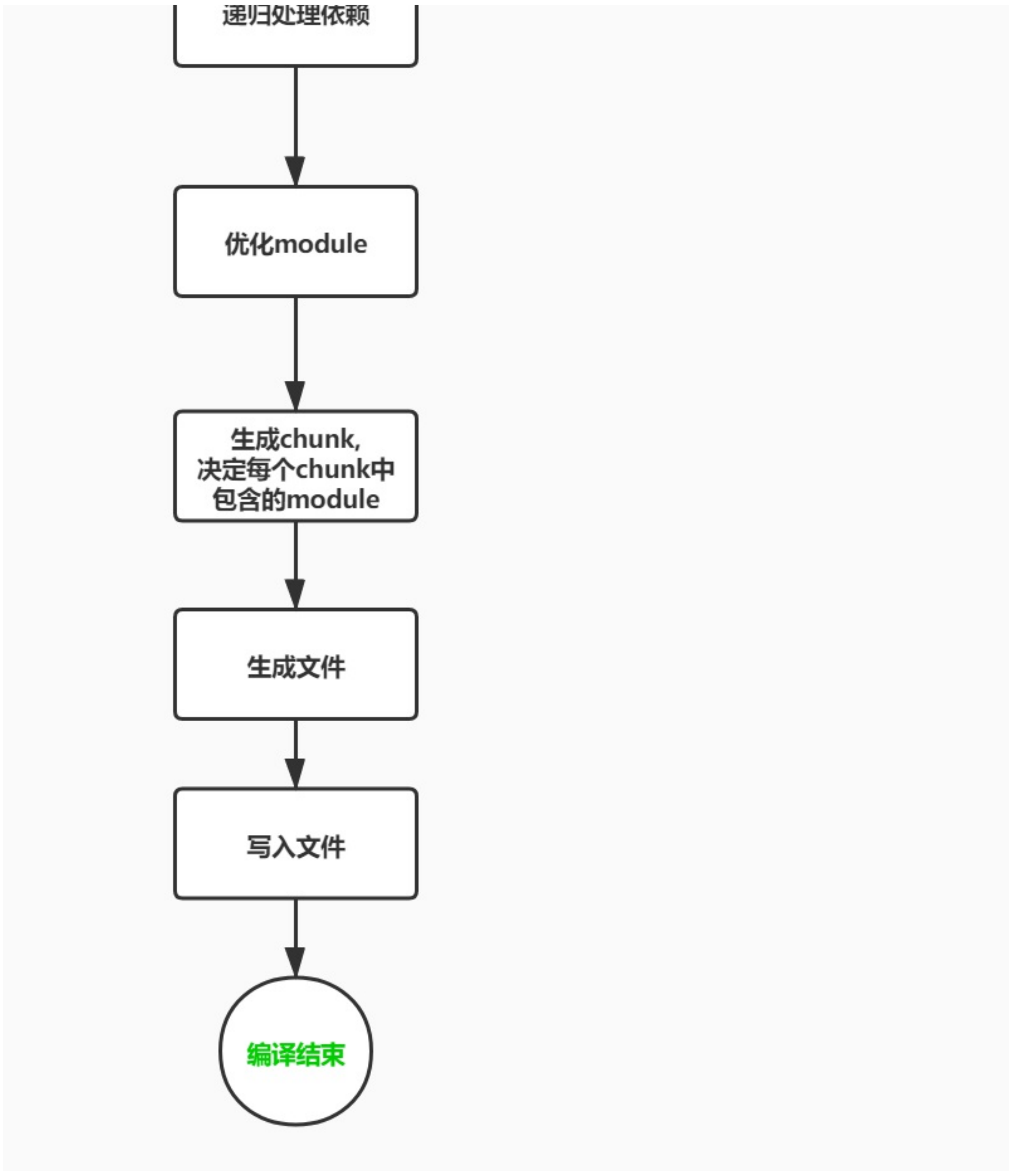
link: null
title: 珠峰架构师成长计划
description: null
keywords: null
author: null
date: null
publisher: 珠峰架构师成长计划
stats: paragraph=43 sentences=66, words=371

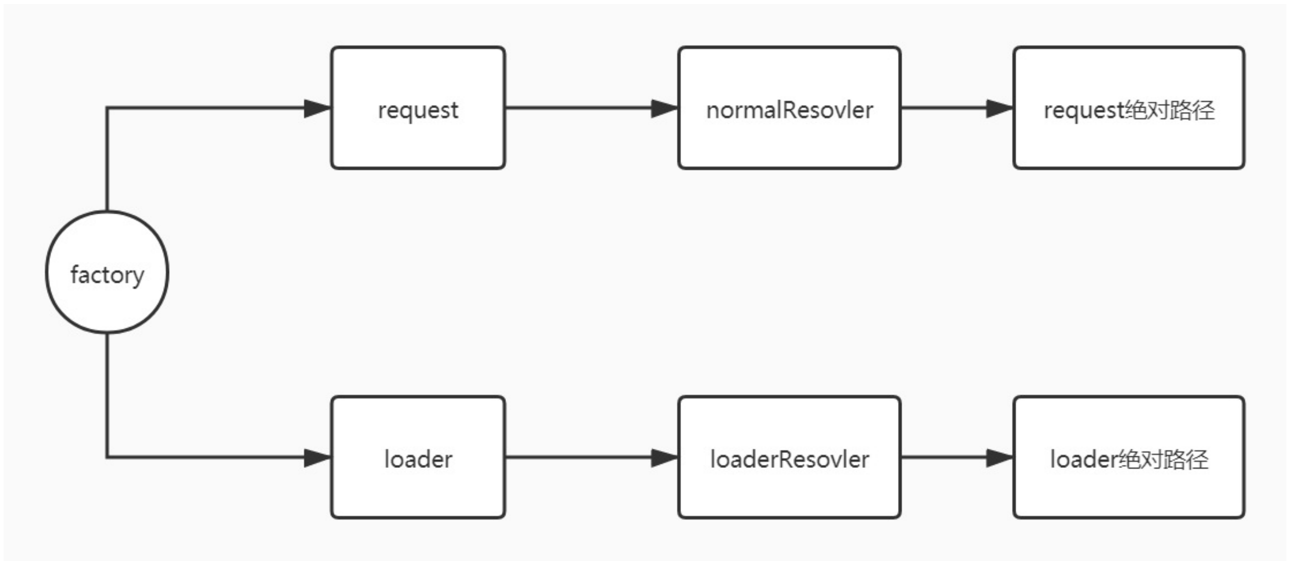
1. webpack整体工作流程

- resolve (https://webpack.docschina.org/configuration/resolve/#resolve)


编译开始
处理配置
合并配置项、错误检查、增加默认配置
编译初始化
配置plugin、创建compiler
编译入口文件
resolve
解析要加载模块的绝对路径 解析inline和配置文件中的loaders
使用loader转换模块
parser
把源码转成AST语法树，解析出import和require



2. resolve入口 [#]



- resolve 流程开始的入口在 `factory` 阶段
- [NormalModuleFactory (https://github.com/webpack/webpack/blob/v4.43.0/lib/NormalModuleFactory.js#L123-L158)]

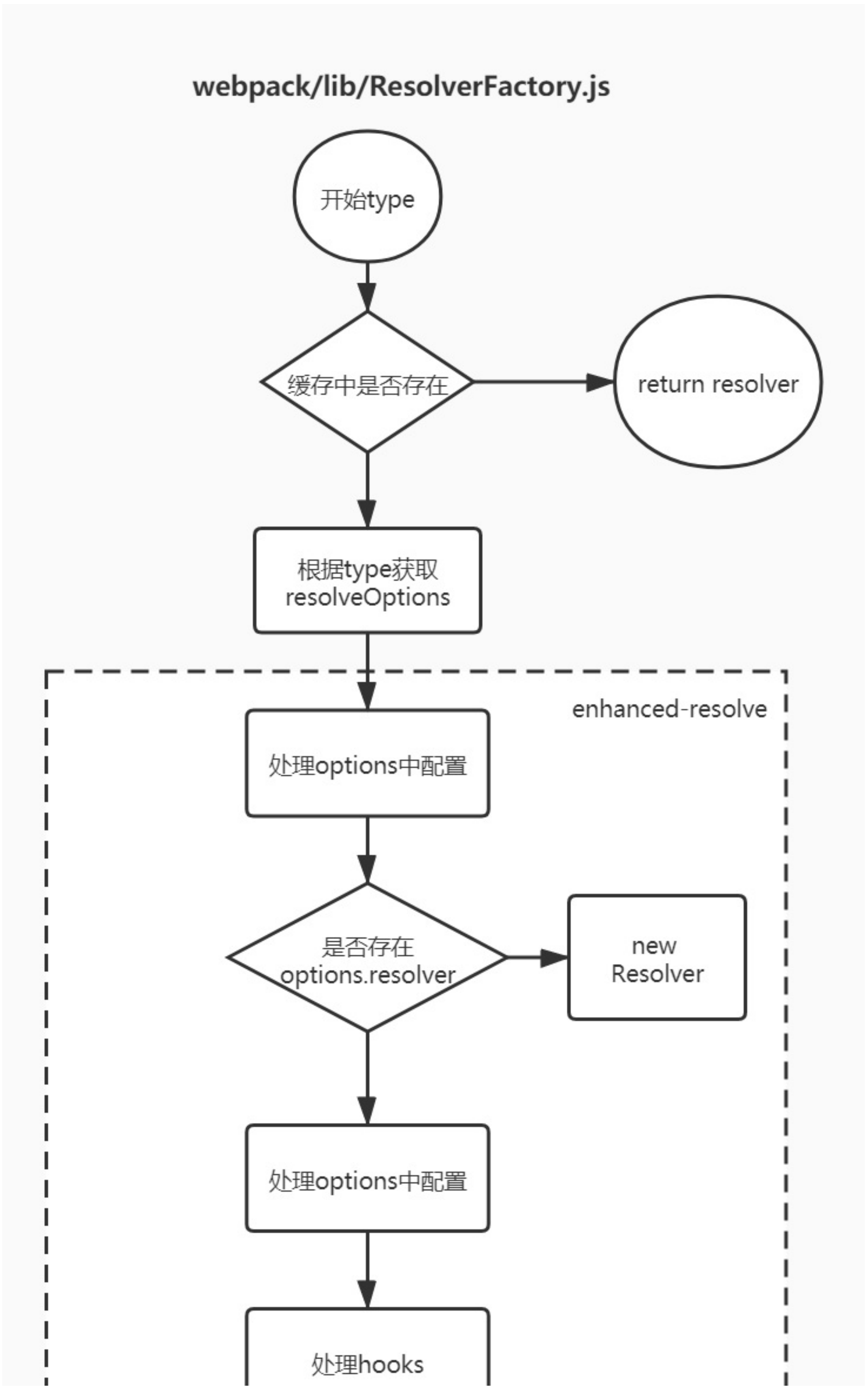
```
this.hooks.factory.tap("NormalModuleFactory", () => (result, callback) => {
    let resolver = this.hooks.resolver.call(null);
    resolver(result, (err, data) => {
        this.hooks.afterResolve.callAsync(data, (err, result) => {    });
    });
});
```

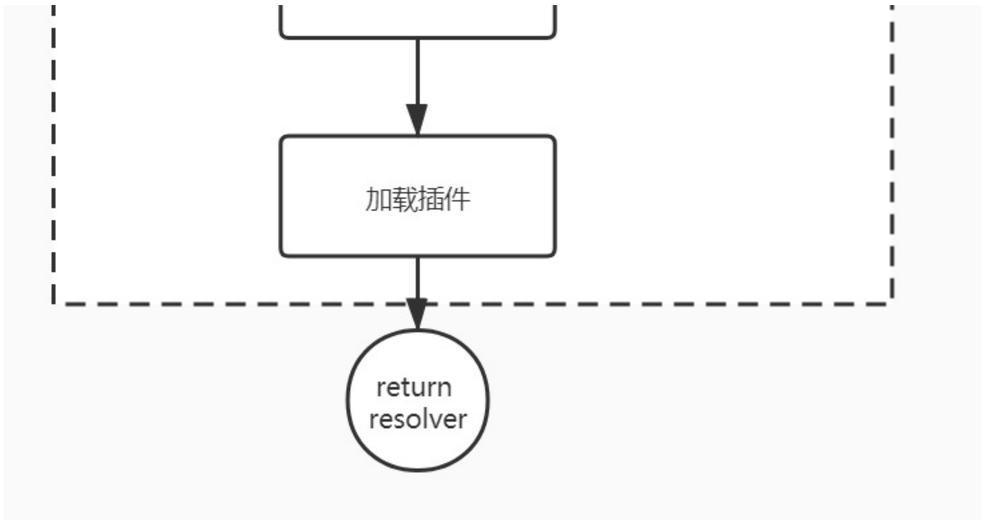
- NormalModuleFactory (https://github.com/webpack/webpack/blob/v4.43.0/lib/NormalModuleFactory.js#L159-L371)

```
this.hooks.resolver.tap("NormalModuleFactory", () => (data, callback) => {

    let elements = request.split("!");
    let resource = elements.pop();
    const loaderResolver = this.getResolver("loader");
    const normalResolver = this.getResolver("normal");
    let loaders = loaderResolver.resolve(elements);
    let request = normalResolver.resolve(resource);
})
```

ResolverFactory (https://github.com/webpack/webpack/blob/v4.43.0/lib/ResolverFactory.js#L44-L52)





```
    get(type, resolveOptions) {
        const newResolver = this._create(type, resolveOptions);
        return newResolver;
    }
```

```
resolve(context, path, request, resolveContext, callback) {
    return this.doResolve(this.hooks.resolve,obj);
}
```

3. resolve流程

- UnsafeCachePlugin (https://github.com/webpack/enhanced-resolve/blob/v4.1.1/lib/UnsafeCachePlugin.js) 增加缓存
- ParsePlugin (https://github.com/webpack/enhanced-resolve/blob/v4.1.1/lib/ParsePlugin.js) 初步解析路径
- DescriptionFilePlugin (https://github.com/webpack/enhanced-resolve/blob/v4.1.1/lib/DescriptionFilePlugin.js) 查看package.json文件
- AliasFieldPlugin (https://github.com/webpack/enhanced-resolve/blob/v4.1.1/lib/AliasFieldPlugin.js) 读取package.json中的别名
- AliasPlugin (https://github.com/webpack/enhanced-resolve/blob/v4.1.1/lib/AliasPlugin.js)取配置项中的别名
- ModuleKindPlugin (https://github.com/webpack/enhanced-resolve/blob/v4.1.1/lib/ModuleKindPlugin.js) 判断是模块
- ModulesInHierachicDirectoriesPlugin (https://github.com/webpack/enhanced-resolve/blob/v4.1.1/lib/ModulesInHierachicDirectoriesPlugin.js)
- JoinRequestPlugin (https://github.com/webpack/enhanced-resolve/blob/v4.1.1/lib/JoinRequestPlugin.js) 连接得到两个完整的路径
- FileKindPlugin (https://github.com/webpack/enhanced-resolve/blob/v4.1.1/lib/FileKindPlugin.js) 判断是否为一个 directory
- DirectoryExistsPlugin (https://github.com/webpack/enhanced-resolve/blob/v4.1.1/lib/DirectoryExistsPlugin.js) 判断directory是否存在
- MainFieldPlugin (https://github.com/webpack/enhanced-resolve/blob/v4.1.1/lib/MainFieldPlugin.js) 读取package.json中的main字段
- UseFilePlugin (https://github.com/webpack/enhanced-resolve/blob/v4.1.1/lib/UseFilePlugin.js) 试图读取目录下的index文件
- FileExistsPlugin (https://github.com/webpack/enhanced-resolve/blob/v4.1.1/lib/FileExistsPlugin.js) 读取 request.path 所在的文件，看文件是否存在
- ResultPlugin (https://github.com/webpack/enhanced-resolve/blob/v4.1.1/lib/ResultPlugin.js) 生成结果

3.1 doResolve

- Resolver.js (https://github.com/webpack/enhanced-resolve/blob/v4.1.1/lib/Resolver.js)

```
let { AsyncSeriesBailHook } = require('tapable');
let resolveHook = new AsyncSeriesBailHook(["request", "innerContext"]);
let resultHook = new AsyncSeriesBailHook(["content", "innerContext"]);
let resolver = {
    doResolve(resolveHook, request, callback) {
        return resolveHook.callAsync(request, (err, result) => {
            if (err) return callback(err);
            if (result) return callback(null, result);
        });
    }
}
resultHook.tapAsync('resultHook', (content, callback) => {
    console.log('resultHook');
    callback(null, content);
});
resolveHook.tapAsync('resolveHook', (request, callback) => {
    console.log('resolveHook');
    let content = request + '的内容';
    resolver.doResolve(resultHook, content, (err, result) => {
        callback(err, result);
    });
});
resolver.doResolve(resolveHook, './src/index.js', (err, result) => {
    console.log(result);
    console.log('完成');
})
```

3.1 resolve流程

- node_modules_ enhanced-resolve@4.1.1 (mailto:enhanced-resolve@4.1.1)@enhanced-resolve

3.1.1 index.js

index.js

```
let a = 10;
```

3.1.2 cli.js

```
const webpack = require("webpack");
const webpackOptions = require("./webpack.config");
const compiler = webpack(webpackOptions);
compiler.run((err, stats) => {
    console.log(err);
    console.log(stats);
});
```

3.1.3 流程

```

const obj = {path: path,request: request};
```

UnsafeCachePlugin

```
resolver.doResolve('newResolve');
```

ParsePlugin

```

resolver.doResolve('parsedResolve', obj);
```

DescriptionFilePlugin

```

resolver.doResolve('describedResolve',obj);
```

AliasFieldPlugin

```
browser
```

AliasPlugin

```
alias
```