

公共模块开发指南

迅雷前端 孙彬彬

导航

- What - 什么是公共模块
- Why - 为什么要开发公共模块
- Who - 公共模块的用户
- How - 如何开发公共模块

What - 什么是公共模块

- 公共模块，是能够让编程者快速开发出应用程序的工具集合
- 通常包括一个或多个 API 和说明文档

What - 公共模块的特点



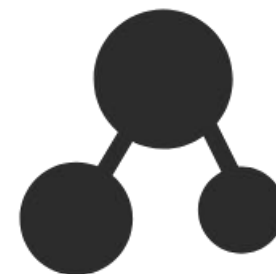
原生



友好



可测试



可复用

What - 我开发的公共模块

- XL9 API
- Xunlei Analysis

模块列表_XNPM - Xunlei x


xnpm.repo.xunlei.cn/#/packages?author=binbin

XNPM

模块 开发者 文章

Search packages...

登录



binbin


sunbinbin@xunlei.com

9 个模块

@xunlei/xl91.5.1


xl9 api for js in xl9 desktop app

xl9, api

binbin


@xunlei/xlm1.0.5

调用 Mac 迅雷本地接口

binbin

@xunlei/sget1.0.3

安全地读对象的属性

binbin

@xunlei/gallery0.1.0

@xunlei/dorayaki0.4.0

@xunlei/impressbook1.0.0

Why - 造轮子



Why - 庞大的项目

庞大的业务需求



Why - 重复建设



Why - 繁琐的细节

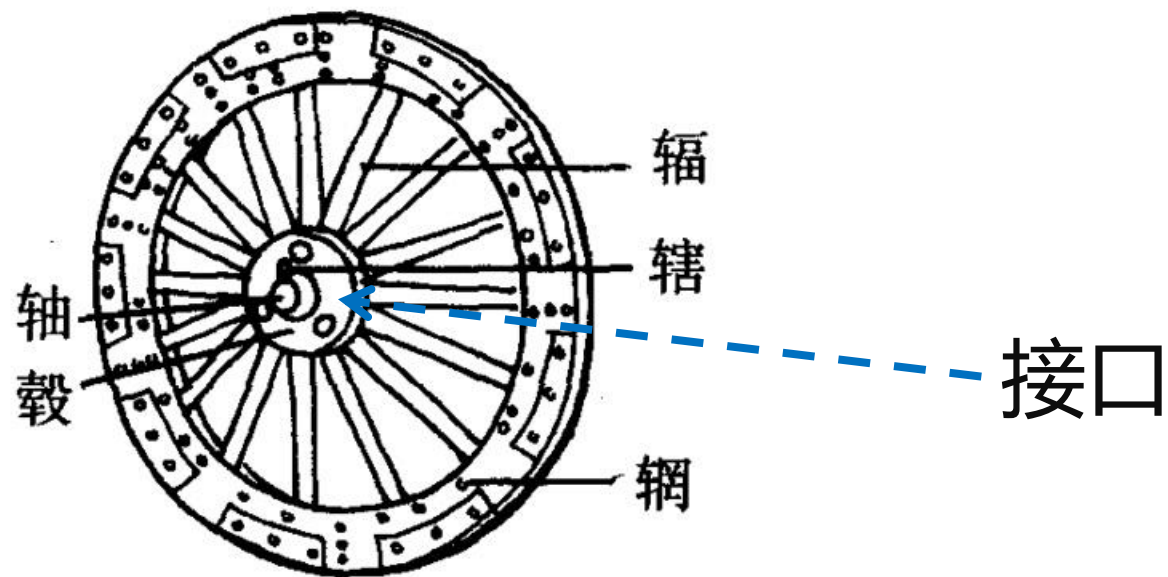
轴 zhóu

辖 xiá

辐 fú

辋 wǎng

毂 gǔ



Why - 为什么要开发公共模块

- 解决模块复用问题，避免重复建设
- 提高开发效率
- 代码封装，隐藏技术细节

Who - 面向用户

开发者

1. 原生
2. 封装友好
3. 可测试
4. 可复用



How - 怎么做

安装

接口设计

单元测试

版本管理

文档

1TH

2TH

3TH

4TH

5TH

安装

CDN NPM 模块化

安装

CDN

```
<script src="//path/to/my-module.js"></script>
```

NPM

```
npm install my-module
```

CDN VS NPM

CDN

NPM

安装

提供资源的在线地址

使用 Node.js 模块管理工具安装

作用域

注册全局变量

模块内作用域

更新

更新后自动全网覆盖

需要开发者手动更新

浏览器使用

直接使用

需要打包后使用

CDN - 同步

```
<script src="//path/to/my-module.js"></script>
```

```
<script src="//path/to/index.js"></script>
```

阻塞、白屏时间

CDN

- 强依赖 - 同步
- 弱依赖 - 异步

CDN - 异步

```
<script src="//path/to/my-module.js" async></script>
```

```
<script src="//path/to/my-module.js" defer></script>
```

异步图解



异步引用反例

```
<script async src="//path/to/my-module.js"></script>  
<script>  
  myModule.foo();  
</script>
```

正确姿势1

```
<script async src="//path/to/my-module.js"></script>
<script>
  document.addEventListener('DOMContentLoaded', function () {
    myModule.foo()
  })
</script>
```

正确姿势2

```
<script>
```

```
  window.myModule = window.myModule || []
```

```
</script>
```

```
<script async src="//path/to/my-module.js"></script>
```

```
<script>
```

```
  myModule.push('foo')
```

```
</script>
```

操作队列

myModule



覆盖对象

```
myModule = {  
  push: function () {  
    _foo()  
  }  
}
```

模块规范

- ES Module - Node.js 支持
- CommonJS - Node.js 支持
- AMD - RequireJS
- CMD - SeaJS
- UMD - 兼容多规范

CommonJS & ES Module

```
var $ = require('jquery')
```

```
module.exports = function (a, b) {  
  return $.extend(a, b)  
}
```

```
import $ from 'jquery'
```

```
export default function (a, b) {  
  return $.extend(a, b)  
}
```

AMD - 依赖前置

```
define(['jquery'], function ($) {  
    return function (a, b) {  
        $.extend(a, b)  
    }  
})
```

CMD - 依赖就近

```
define(function(require, exports, module) {  
    var $ = require('jquery')  
  
    module.exports = function (a, b) {  
        return $.extend(a, b)  
    }  
})
```

CDN 导出

```
window.myModule = function (a, b) {  
    $.extend(a, b)  
}
```

安装 - 打包工具



Webpack 打包库文件

```
module.exports = {  
  entry: './app.js',  
  output: {  
    filename: 'bundle.js',  
    library: 'myModule',  
    libraryTarget: 'umd'  
  }  
}
```


UMD

```
(function (root, factory) {  
  if (typeof define === 'function' && define.amd) {  
    define(['jquery'], factory); // AMD  
  } else if (typeof exports === 'object') {  
    module.exports = factory(require('jquery')); // CommonJS CMD  
  } else {  
    root.myModule = factory(root.jQuery); // 浏览器全局变量  
  }  
})(this, function ($) {  
  return function (a, b) {  
    return $.extend(a, b)  
  }  
})
```

接口设计

命名 同步 异步 参数 跨平台

接口命名 - 正确拼写

不可更改

- creat -> create
- referer -> referrer
- taost -> toast
- pannel -> panel

接口命名 - 语义

- isAuth 是否已认证
- addItem 增
- removeItem 删
- getItem 查
- updateItem 改

幂等性

一个函数无论在什么上下文中运行、运行多少次，最后的结果总是相同的。

- 严格限定模块对外部作用域的修改
- 对关键成员作访问控制，冻结写权限

接口命名 - 时态

- beforeCreate
- created
- beforeMount
- mounted
- beforeUpdate
- updated
- beforeDestroy
- destroyed

同步接口

```
function foo () {  
    return JSON.parse(json)  
}
```

同步接口

```
try {  
    var val = foo ()  
} catch (e) {  
    // handle error  
}
```


异步接口

```
function foo () {  
  setTimeout(function () {  
    return JSON.parse(json)  
  }, 1000)  
}
```

异步接口 - 回调

```
function foo (cb) {  
  setTimeout(function () {  
    try {  
      cb(null, JSON.parse(json))  
    } catch (e) {  
      cb(e)  
    }  
  }, 1000)  
}
```

异步接口 - 回调

```
foo(function (err, val) {  
    if (!err) {  
        // use val  
    } else {  
        // handle error  
    }  
})
```

异步接口 - Promise

```
function foo (cb) {  
  return new Promise(function (resolve, reject) {  
    setTimeout(function () {  
      try {  
        resolve(JSON.parse(json))  
      } catch (e) {  
        reject(e)  
      }  
    }, 1000)  
  })  
}
```

异步接口 - Promise

```
foo()  
  .then(function (val) {  
    // use val  
  })  
  .catch(function (err) {  
    // handle error  
  })
```

参数设计

- 相关性越高的参数越要前置
- 可省略的参数后置

`$.get(url, data, success, dataType)`

参数设计

```
// traditional  
$.get(url, data, success, dataType)
```

```
// or  
$.ajax({  
  url,  
  data,  
  success,  
  dataType  
})
```

参数设计

优点

- 不用关心参数顺序
- 不必担心参数列表过长

缺点

- 无法突出参数的优先级
- 设置参数默认值麻烦

跨语言

- 长整数转字符串

Number.MAX_SAFE_INTEGER: 9007199254740991
9184180752792413183 -> '9184180752792413183'

- 对象转JSON

{ isVIP: true } -> '{"isVIP":true}'

单元测试

测试框架 断言 覆盖率

单元测试

对软件中的最小可测试单元进行检查和验证。

单元测试框架



Jasmine



Mocha



Qunit

断言

断言表示为一些布尔表达式。

```
assert Expression
```

```
assert 1 + 1 === 2
```

断言库

- Node.js' built-in assert module
- should.js
- expect.js
- chai

同步

```
var chai = require('chai');
var expect = chai.expect;

describe('Array', function () {
  describe('#indexOf()', function () {
    it('should return -1 when the value is not present', function () {
      expect([1, 2, 3].indexOf(4)).to.equal(-1);
    });
  });
});
```

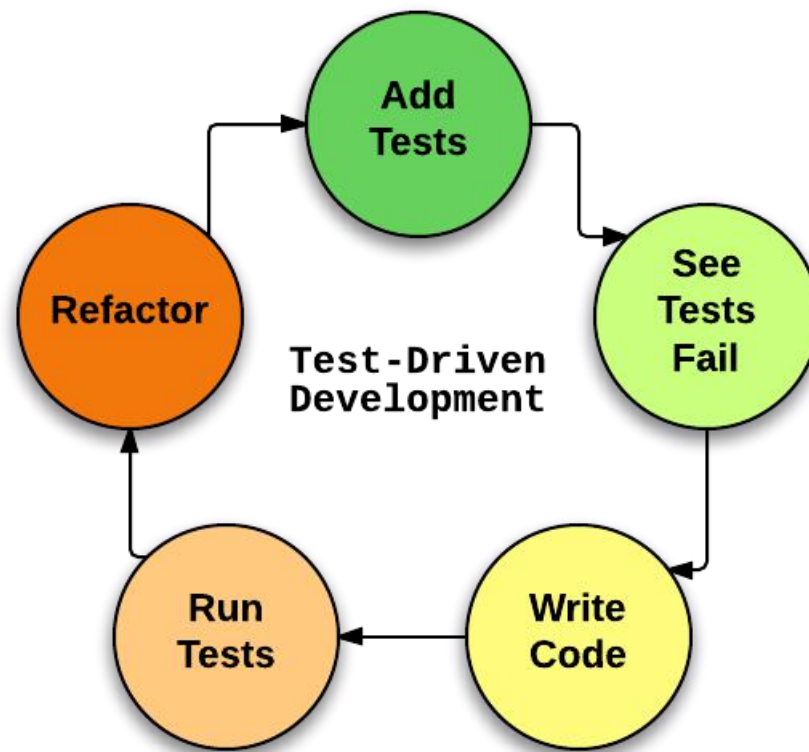
异步

```
describe('setTimeout()', function () {  
  it('should start a timer and excute function', function (done) {  
    setTimeout(function () {  
      done();  
    }, 1000);  
  });  
});
```


异步

```
describe('Promise.resolve()', function () {  
  it('should return a resolved promise', function () {  
    return Promise.resolve();  
  });  
});
```

TDD 测试驱动开发



代码覆盖率

- 行覆盖率 (line coverage) : 是否每一行都执行了 ?
- 函数覆盖率 (function coverage) : 是否每个函数都调用了 ?
- 分支覆盖率 (branch coverage) : 是否每个if代码块都执行了 ?
- 语句覆盖率 (statement coverage) : 是否每个语句都执行了 ?

Istanbul

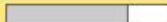


Istanbul

Code coverage report for ./

Statements: **75%** (3 / 4) Branches: **50%** (1 / 2) Functions: **100%** (1 / 1) Lines: **100%** (3 / 3)

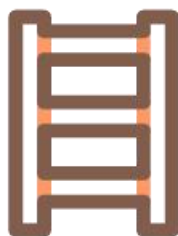
[All files](#) » ./

File ▲	Statements	Branches	Functions	Lines
sqrt.js 	75.00% (3 / 4)	50.00% (1 / 2)	100.00% (1 / 1)	100.00% (3 / 3)

版本管理

版本号 提交 更新日志

版本号管理



3.1.0



3.1.1



3.2.0



4.0.0

语义化版本 semver

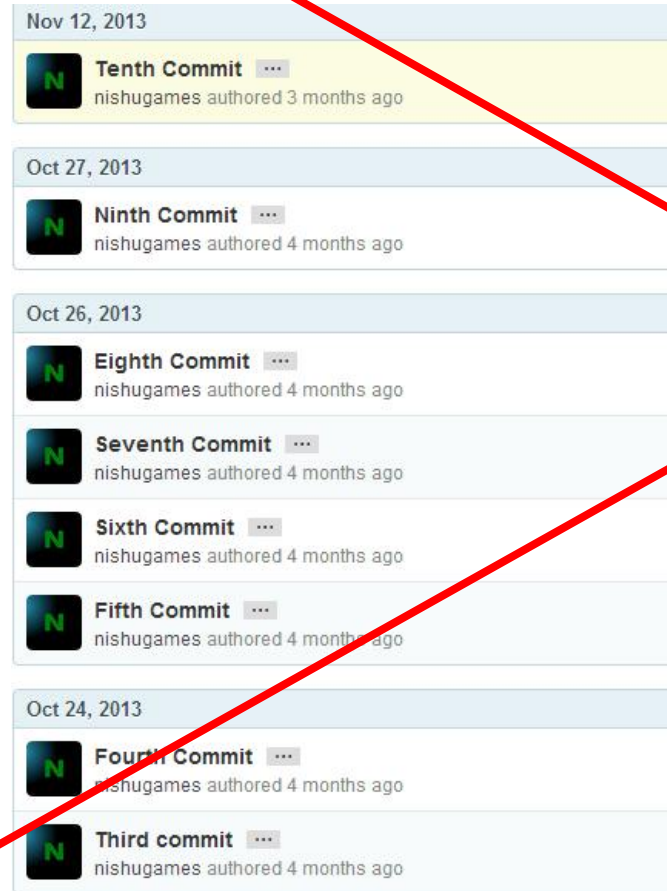
主版本号.次版本号.修订号

- 主版本号：当你做了**不兼容**的 API 修改，
- 次版本号：当你做了**向下兼容**的功能性新增，
- 修订号：当你做了**向下兼容**的问题修正。

Commit Message

```
git commit -m "This is my first commit"
```

Commit Message



<http://whatthecommit.com/>

- It worked for me...
- We Had To Use Dark Magic To Make This Work
- omgsosorry

Commits on Jan 5, 2016



docs(error/\$rootScope/inprog): add missing "\$timeout" ...

wytesk133 committed with petebacondarwin 10 days ago



docs(tutorial/2): add e2e test missing filename ...

monkpit committed with petebacondarwin 3 hours ago



revert: feat(\$compile): Allow ES6 classes as controllers with `bindTo... ...

petebacondarwin committed 3 hours ago



fix(\$animateCss): only (de)register listeners when events have been a... ...

Narretz committed 25 days ago



fix(loader): use `false` as default value for `transclude` in compone... ...

sarod committed with petebacondarwin 18 days ago

Commits on Jan 3, 2016



feat(\$compile): Allow ES6 classes as controllers with `bindToControll... ...

Igalfaso committed 21 days ago



chore(*): Updated year in licence ...

leuanG committed with Igalfaso 3 days ago

Commits on Jan 1, 2016



docs: reorganize information about interpolation ...

Narretz committed on Dec 6, 2015



docs: add docs for ngPattern, ngMinlength, ngMaxlength ...

Narretz committed on Sep 24, 2015



docs(\$interpolateProvider): remove superfluous ng-app attribute ...

Narretz committed 4 days ago

Angular 规范

<type>(<scope>): <subject>

// 空一行

<body>

// 空一行

<footer>

Commitizen

```
$ commitizen init cz-conventional-changelog --save -  
-save-exact
```

```
$ git cz
```

Commitizen

```
→ ng-poopy master ✗ git add .  
→ ng-poopy master ✗ git cz
```

All commit message lines will be cropped at 100 characters.

```
? Select the type of change that you're committing: (Use arrow keys)  
> feat:    A new feature  
fix:      A bug fix  
docs:     Documentation only changes  
style:    Changes that do not affect the meaning of the code  
          (white-space, formatting, missing semi-colons, etc)  
refactor: A code change that neither fixes a bug or adds a feature  
perf:     A code change that improves performance  
test:     Adding missing tests  
chore:    Changes to the build process or auxiliary tools  
          and libraries such as documentation generation
```

Change Log

0.0.4 (2015-07-21)

Bug Fixes

- **glossary:** fix bad things in the glossary ([9e572e9](#))
- **intro:** fix issue with the intro not working ([6de0d8a](#))

Features

- **part2:** add better veggie lorem ipsum text ([06912f4](#))
- **part2:** add much better lorem ipsum text ([7aaf238](#))

BREAKING CHANGES

- Adding better veggie lorem ipsum text is great and all but it breaks all the other meaty components. Be mindful of this when you test.

conventional-changelog

```
$ conventional-changelog -p angular -i CHANGELOG.md  
-w
```


文档

README 在线文档

文档

- 内在：源码
- 外在：文档
- 文档是用户的**第一印象**



README

项目门面

- 简要介绍
- 安装
- 使用示例

vue-lazy-component/R x

安全 | <https://github.com/xunleif2e/vue-lazy-component/blob/master/README.md>

>

This repository

Search

Pull requests

Issues

Marketplace

Explore

+

xunleif2e / vue-lazy-component

Unwatch

12

Unstar

200

Fork

30

<> Code

Issues 0

Pull requests 0

Projects 0

Wiki

Insights

Settings

Branch: master

vue-lazy-component / README.md

Find file

Copy path

binggg docs(i18n): write readme in english

e7313d7 on 29 Sep

1 contributor

194 lines (136 sloc) | 7.16 KB

Raw

Blame

History

Vue Lazy Component

npm

v1.0.10

commitizen

friendly

git-flow

workflow

issues

0 open

forks

30

license

MIT

Vue.js 2.x 组件级懒加载方案 Vue.js 2.x component level lazy loading component

特性

- 支持 组件可见或即将可见时懒加载
- 支持 组件延时加载
- 支持 加载真实组件前展示骨架组件，提高用户体验
- 支持 真实组件代码分包异步加载

README - 其他

- 徽章 Badge - 构建结果、版本信息等
- 接口 API
- 更新日志 Change Log
- 开源协议 License

在线文档

- <http://speed.xunlei.com/speed.js>
- <http://speed.xunlei.com/doc/index.html>

JSDoc

- 根据 javascript 文件中**注释信息**，生成 JavaScript 应用程序或库、模块的 API 文档的工具
- 记录命名空间，类，方法，方法参数等

JSDoc

162

163

164

165

166

167

export function removeCookie (key) {
 document.cookie = `\${key}=; expires=\${new Date(-1).toUTCString()}; domain=xunlei.com; path=/`;
}

setC

问题

输出

sunbinbin@D
\$

setCookie

SVGTextContentElement

SVGFETurbulenceElement

MSInputMethodContext

SVGPathSegCurvetoCubicAbs

SVGPathSegCurvetoCubicRel

SVGPathSegCurvetoCubicSmoothAbs

SVGPathSegCurvetoCubicSmoothRel

SVGElementInstance

SVGElementInstanceList

MediaStreamTrack

MediaStreamTrackEvent

function setCookie(key: string, value: string, {path, domain, maxAge, expires, secure}?: {
 path?: string;
 domain?: string;
 maxAge?: number;
 expires: any;
 secure: any;
}): void

设置cookie

JSDoc

```
/**  
 * Book类， 代表一个书本.  
 * @constructor  
 * @param {string} title - 书本的标题.  
 * @param {string} author - 书本的作者.  
 */  
function Book(title, author) {  
    this.title=title;  
    this.author=author;  
}
```

Home - Documentatio x

misc-xl9-ssl.xunlei.com/res/api/xl9/v1/doc/index.html

Home

TUTORIALS

在首页中使用迅雷9 API
登录与登录态
使用插件接口

GLOBAL

attach
autoJump
call
createTask
detach
getConfigValue
getJumpKey
getPeerId
getSessionId
getSkinId
getSkinInfo
getTabIdByContext
getThunderVersion
getUserId
isMainPageFirstShow
isTabExist
isThunder
login
logout
onChangeSkin
onUserLogin
onUserLogout
openNewTab
selectTab
setConfigValue
showLoginDlg
trackEvent

迅雷9 API 用户接口文档

迅雷9客户端启动chrome内核，所有接口采取异步调用方式，逐步取代IE内核接口。本项目的目的是为了兼容新旧内核的同步和异步版本接口，同时处理一些常见的业务逻辑。

安装

在迅雷9首页

```
definePlugin(['xl9_v1'], function (xl9) {});
```

在迅雷9非首页

```
<script src="http://misc.xl9.xunlei.com/res/api/xl9/v1/xl9.js"></script>
```

升级

如果API的调用接口不变，意味着你的代码不需要做出改变，API的版本号也不会变化。别担心，API可能只是在修复bug或者增加新特性，不会影响你的代码。

当版本号改变时（v1到v2）：

```
http://misc.xl9.xunlei.com/res/api/xl9/v2/xl9.js
```

你应该修改代码来使用新的API。

misc-xl9-ssl.xunlei.com/res/api/xl9/v1/doc/global.html#call

总结 & QA

Summary

