

翻译师 => **What?**

## what > dubbo2.js 使用示例

```
import {Dubbo, java, TDubboCallResult} from 'dubbo2.js';
```

```
//定义dubbo方法类型接口
```

```
//方便代码自动提示
```

```
interface IDemoService {  
  sayHello(name: string): TDubboCallResult<string>;  
  
  echo(): TDubboCallResult<string>;  
  
  test(): TDubboCallResult<void>;  
  
  getUserInfo(): TDubboCallResult<{  
    status: string;  
    info: {id: number; name: string};  
  }>;  
}
```

```
//创建dubbo对象
```

```
const dubbo = new Dubbo({  
  application: {name: 'node-dubbo'},  
  //zookeeper address  
  register: 'localhost:2181',  
  dubboVersion: '2.0.0',  
  interfaces: ['com.alibaba.dubbo.demo.DemoService']  
});
```

```
//代理本地对象->dubbo对象
```

```
const demoService = dubbo.proxyService<IDemoService>({  
  dubboInterface: 'com.alibaba.dubbo.demo.DemoService',  
  version: '1.0.0',  
  methods: {  
    sayHello(name) {  
      //仅仅做参数hessian化转换  
      return [java.String(name)];  
    },  
  
    echo() {},  
  
    test() {},  
  
    getUserInfo() {  
      //仅仅做参数hessian化转换  
      return [  
        java.combine('com.alibaba.dubbo.demo.UserRequest', {  
          id: 1,  
          name: 'nodejs',  
          email: 'node@qianmi.com',  
        }  
      )],  
    ];  
  },  
});
```

how-to-usage



what > dubbo2.js 使用示例

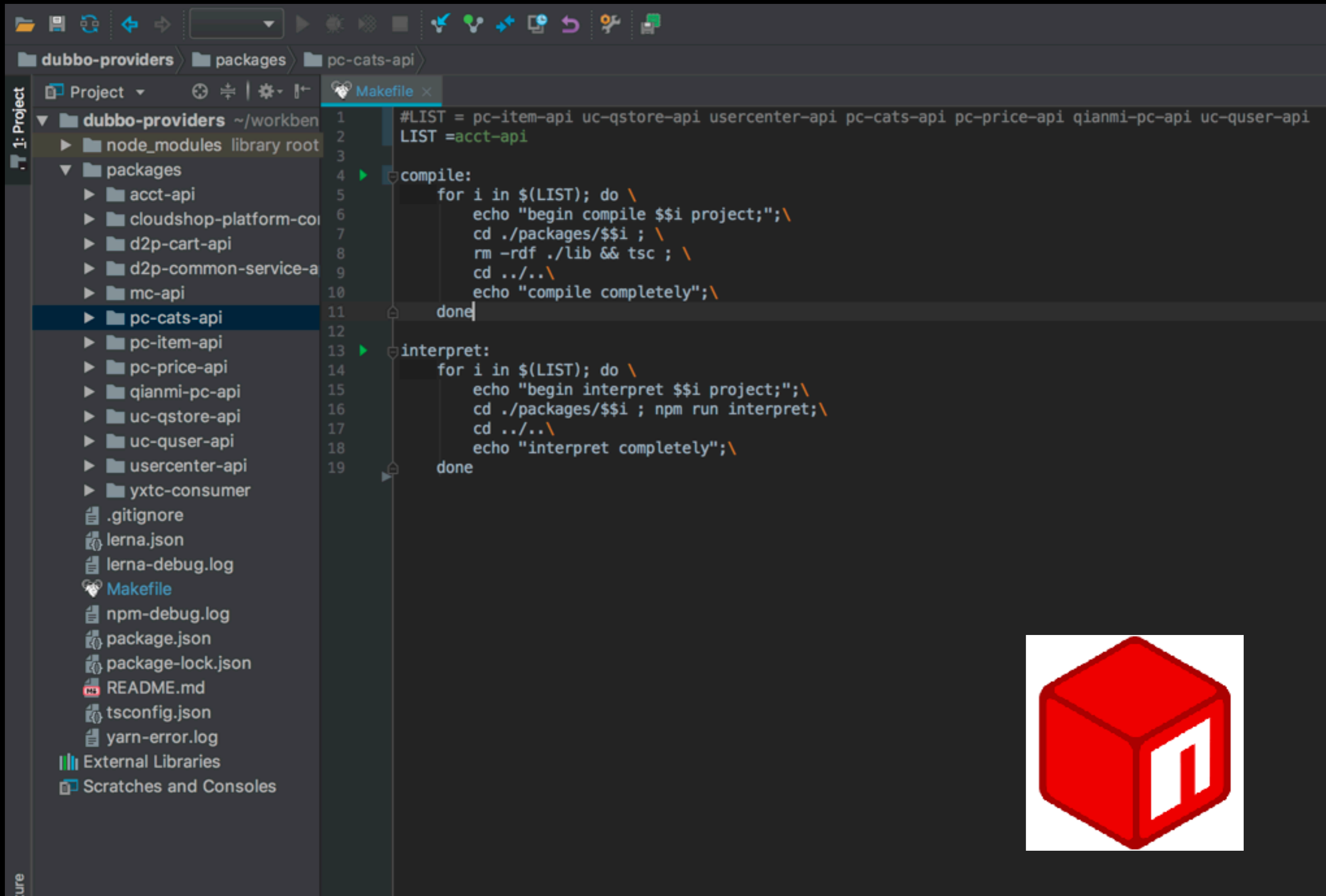
**Target:** **simplify** the process, **improve** develop Experience!

```
import {D2pMarketingQueryProvider} from '@qianmi/d2p-cart-api/lib/com/qianmi/cloudshop/api/marketing/d2p/D2pMarketingQueryProvider';
const dubbo = new Dubbo({
  application: {name: 'd2p-visitor-bff'},
  dubboInvokeTimeout: 10,
  //zookeeper address
  register: app.config.zookeeper,
  dubboVersion: '2.4.13',
  logger: app.logger as ILogger,
  interfaces: [
    'com.qianmi.cloudshop.api.marketing.d2p.D2pMarketingQueryProvider'
  ],
});
let D2pMarketingQuery = D2pMarketingQueryProvider(dubbo);
```

*Tip* npm install interpret-util dubbo2.js;

**how-to-usage**

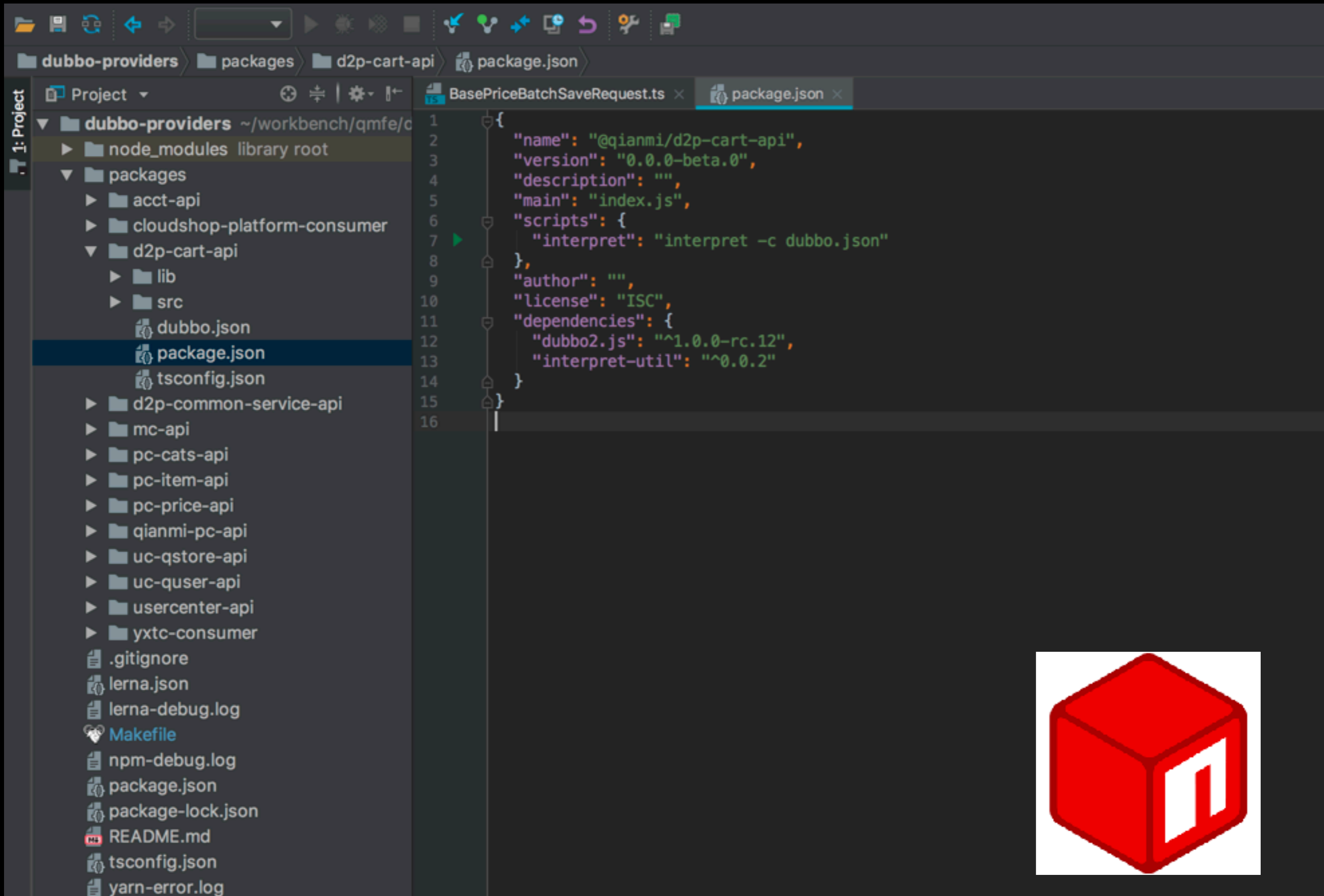
# what > dubbo2.js 使用示例



```
1 #LIST = pc-item-api uc-qstore-api usercenter-api pc-cats-api pc-price-api qianmi-pc-api uc-quser-api
2 LIST =acct-api
3
4 compile:
5     for i in $(LIST); do \
6         echo "begin compile $$i project";\
7         cd ./packages/$$i ; \
8         rm -rdf ./lib && tsc ; \
9         cd ../../\
10        echo "compile completely";\
11    done
12
13 interpret:
14     for i in $(LIST); do \
15         echo "begin interpret $$i project";\
16         cd ./packages/$$i ; npm run interpret;\
17         cd ../../\
18        echo "interpret completely";\
19    done
```



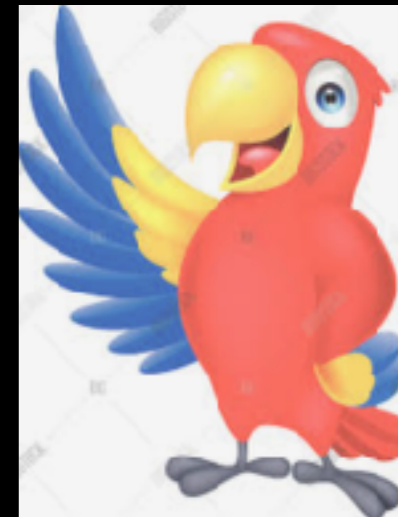
# what > dubbo2.js 使用示例



The screenshot shows an IDE interface with a project explorer on the left and a code editor on the right. The project explorer shows a directory structure for 'dubbo-providers' with a sub-directory 'd2p-cart-api' containing 'package.json'. The code editor displays the content of 'package.json'.

```
1 {
2   "name": "@qianmi/d2p-cart-api",
3   "version": "0.0.0-beta.0",
4   "description": "",
5   "main": "index.js",
6   "scripts": {
7     "interpret": "interpret -c dubbo.json"
8   },
9   "author": "",
10  "license": "ISC",
11  "dependencies": {
12    "dubbo2.js": "^1.0.0-rc.12",
13    "interpret-util": "^0.0.2"
14  }
15 }
16
```





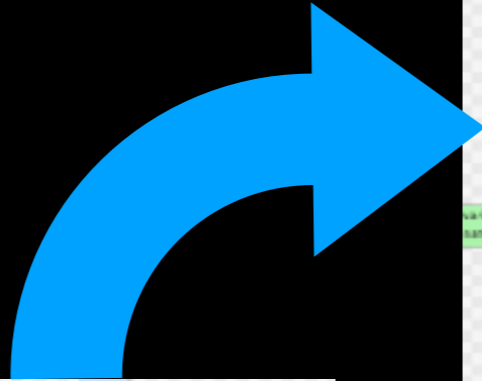
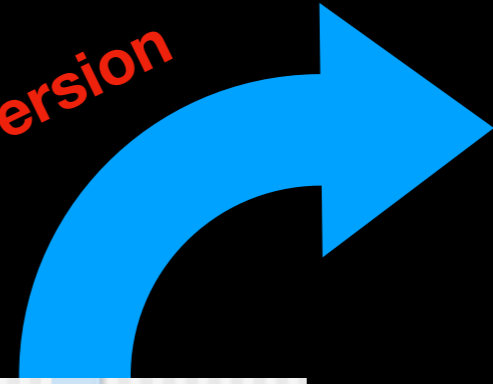
翻译师 => How ?

# how > 翻译流程

- operate Ast
  - > Seamless
  - > Code hints
  - > Parameter conversion



- read Ast
  - > 继承折叠
  - > 依赖分析



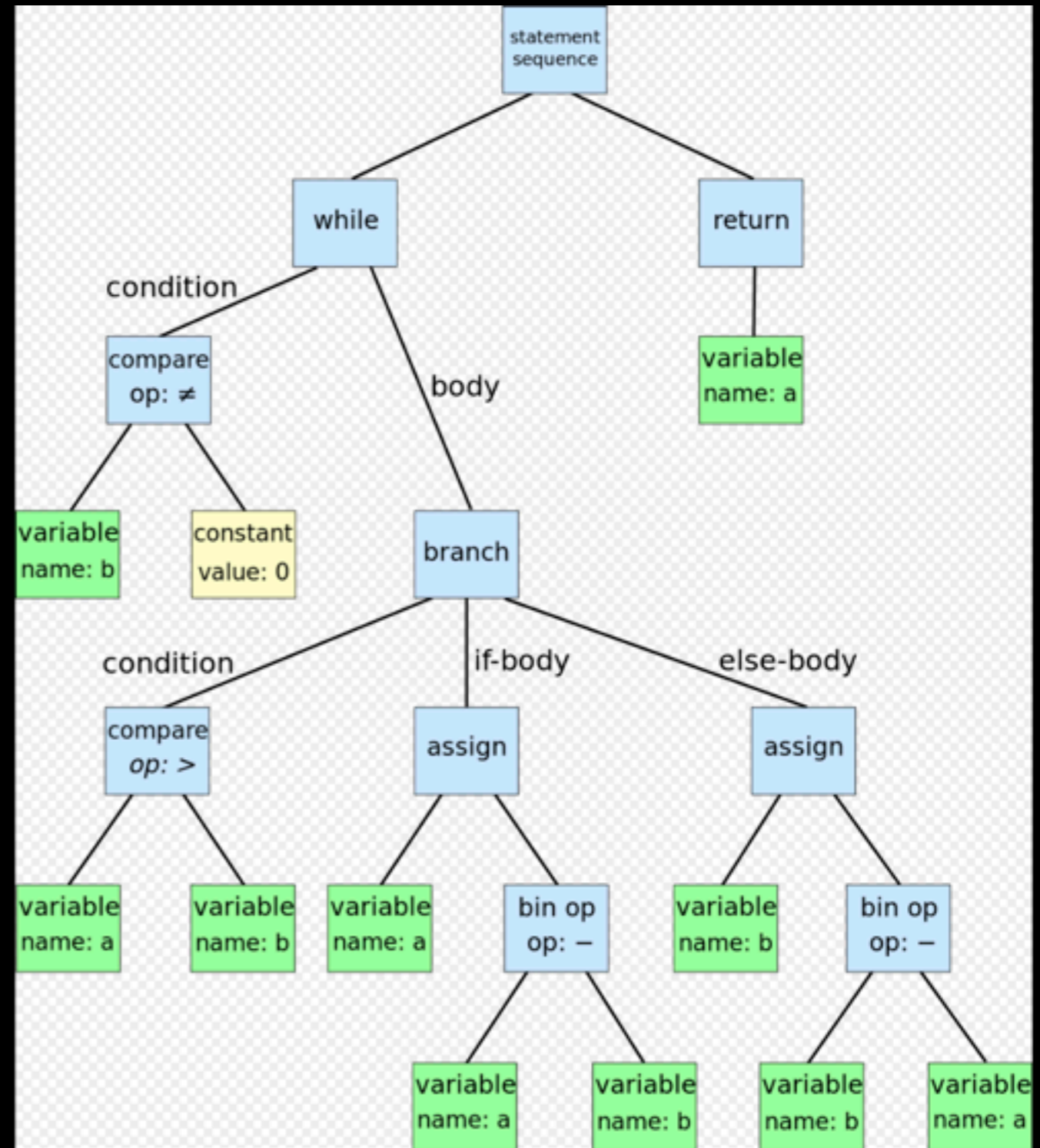
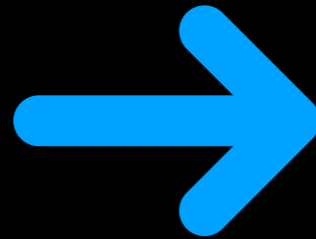


how > 翻译流程 > extra Ast

**Extract Ast**

# how > ast抽象语法树

```
while b ≠ 0
  if a > b
    a := a - b
  else
    b := b - a
return a
```



```
package com.qianmi;

public class HelloDubbo {

    public String dubbo2js;

    public void hello(String name) {
        System.out.println("hello, "+name);
    }
}
```

```
javac HelloDubbo
javac -parameters HelloDubbo
javap -verbose HelloDubbo
```

HelloDubbo.java ×

HelloDubbo.class ×

Decompiled .class file, bytecode version: 53.0 (Java 9)

```
1 //
2 // Source code recreated from a .class file by IntelliJ IDEA
3 // (powered by Fernflower decompiler)
4 //
5
6 package com.qianmi;
7
8 public class HelloDubbo {
9     public String dubbo2js;
10
11     public HelloDubbo() {
12     }
13
14     public void hello(String var1) {
15         System.out.println("hello, " + var1);
16     }
17 }
18
```

```
→ qianmi git:(master) ✗ javap -verbose HelloDubbo
```

```
警告: 文件 ./HelloDubbo.class 不包含类 HelloDubbo
```

```
Classfile /Users/dong/workbench/qmfe/java-providers/pc-micro-price/pc-price-api/src/main/java/com/qianmi/HelloDubbo.class
```

```
  Last modified 2018年4月28日; size 930 bytes
```

```
  MD5 checksum f52807f7d3e8718d2a4ecd1f59c330c6
```

```
  Compiled from "HelloDubbo.java"
```

```
public class com.qianmi>HelloDubbo
```

```
  minor version: 0  
  major version: 53  
  flags: (0x0021) ACC_PUBLIC, ACC_SUPER  
  this_class: #5           // com/qianmi/HelloDubbo  
  super_class: #6         // java/lang/Object  
  interfaces: 0, fields: 1, methods: 2, attributes: 3
```

```
Constant pool:
```

```
  #1 = Methodref          #6.#19           // java/lang/Object."<init>":()V  
  #2 = Fieldref           #20.#21           // java/lang/System.out:Ljava/io/PrintStream;  
  #3 = InvokeDynamic       #0:#25           // #0:makeConcatWithConstants:(Ljava/lang/String;)Ljava/lang/String;  
  #4 = Methodref           #26.#27           // java/io/PrintStream.println:(Ljava/lang/String;)V  
  #5 = Class                #28             // com/qianmi/HelloDubbo  
  #6 = Class                #29             // java/lang/Object  
  #7 = Utf8                 dubbo2js  
  #8 = Utf8                 Ljava/lang/String;  
  #9 = Utf8                 <init>  
 #10 = Utf8                 ()V  
 #11 = Utf8                 Code  
 #12 = Utf8                 LineNumberTable  
 #13 = Utf8                 hello  
 #14 = Utf8                 (Ljava/lang/String;)V  
 #15 = Utf8                 MethodParameters  
 #16 = Utf8                 name  
 #17 = Utf8                 SourceFile  
 #18 = Utf8                 HelloDubbo.java  
 #19 = NameAndType         #9:#10           // "<init>":()V  
 #20 = Class                #30             // java/lang/System  
 #21 = NameAndType         #31:#32           // out:Ljava/io/PrintStream;  
 #22 = Utf8                 BootstrapMethods  
 #23 = MethodHandle        6:#33           // REF_invokeStatic java/lang/invoke/StringConcatFactory.makeConcatWithConstants:(Ljava/  
g/invoke/MethodType;Ljava/lang/String;[Ljava/lang/Object;)Ljava/lang/invoke/CallSite;  
 #24 = String              #34             // hello, \u0001  
 #25 = NameAndType         #35:#36           // makeConcatWithConstants:(Ljava/lang/String;)Ljava/lang/String;  
 #26 = Class                #37             // java/io/PrintStream
```

```

#48 = Utf8                java/lang/invoke/MethodHandles
{
  public java.lang.String dubbo2js;
    descriptor: Ljava/lang/String;
    flags: (0x0001) ACC_PUBLIC

  public com.qianmi.HelloDubbo();
    descriptor: ()V
    flags: (0x0001) ACC_PUBLIC
    Code:
      stack=1, locals=1, args_size=1
        0: aload_0
        1: invokespecial #1          // Method java/lang/Object."<init>":()V
        4: return
   LineNumberTable:
      line 3: 0

  public void hello(java.lang.String);
    descriptor: (Ljava/lang/String;)V
    flags: (0x0001) ACC_PUBLIC
    Code:
      stack=2, locals=2, args_size=2
        0: getstatic      #2          // Field java/lang/System.out:Ljava/io/PrintStream;
        3: aload_1
        4: invokedynamic #3, 0       // InvokeDynamic #0:makeConcatWithConstants:(Ljava/lang/String;)Ljava/lang/String;
        9: invokevirtual #4          // Method java/io/PrintStream.println:(Ljava/lang/String;)V
       12: return
   LineNumberTable:
      line 8: 0
      line 9: 12
    MethodParameters:
      Name          Flags
      name
}
SourceFile: "HelloDubbo.java"
InnerClasses:
  public static final #43= #42 of #46;    // Lookup=class java/lang/invoke/MethodHandles$Lookup of class java/lang/invoke/MethodHandl
BootstrapMethods:
  0: #23 REF_invokeStatic java/lang/invoke/StringConcatFactory.makeConcatWithConstants:(Ljava/lang/invoke/MethodHandles$Lookup;Ljava/
g;[Ljava/lang/Object;)Ljava/lang/invoke/CallSite;
    Method arguments:
      #24 hello, \u0001

```

- javaparser :  
Java 9 Parser and Abstract Syntax Tree for Java
- asm:  
a very small and fast Java bytecode manipulation framework
- jexpose:  
Exposing class graph inside of jar
- ts-simple-ast:  
Wraps the TypeScript compiler API so it's simple.
- pegjs:  
Parser generator for JavaScript
- jsjavaparser:  
~~Java 1.7 Parser and Abstract Syntax Tree~~

# java to Ts

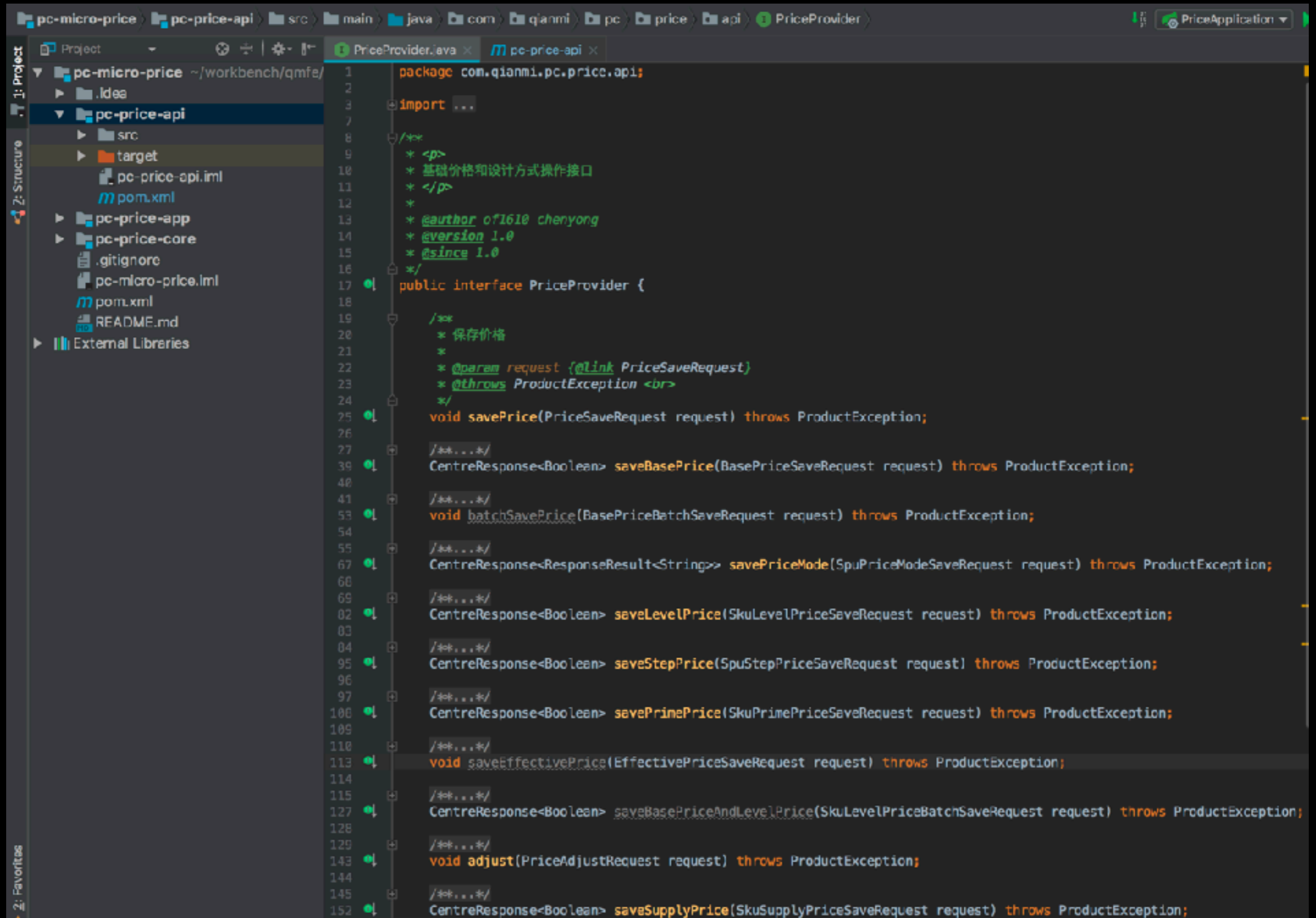
- 基本类型映射
- Interface转换
- Enum转换
- Bean转换



# how > 翻译流程 > 转换 > Java to Ts > 基本类型

Java类型	Typescript类型
java.lang.String	string
java.lang.Object	Object
java.lang.Integer/java.lang.int	number
java.lang.short /java.lang.Short	number
java.lang.long/java.lang.Long	number
java.lang.double/java.lang.Double	number
java.lang.float/java.lang.Float	number
java.lang.Void	void
java.lang.boolean/java.lang.Boolean	boolean
java.util.Set/java.util.LinkedHashSet java.util.List/java.util.Collection	Array
java.util.Date	Date
java.util.map/java.util.HashMap	{[name: \${nameType}]: \${value}}
java.math.BigDecimal	{value:string}
....	...

# how > 翻译流程 > 转换 > Java to Ts > Interface转换示例



The screenshot shows an IDE window with the following structure:

- Project: pc-micro-price
- pc-price-api
- src
- target
- pc-price-api.iml
- pom.xml
- pc-price-app
- pc-price-core
- .gitignore
- pc-micro-price.iml
- pom.xml
- README.md
- External Libraries

```
1 package com.qianmi.pc.price.api;
2
3 import ...
4
5 /**
6  * <p>
7  * 基础价格和设计方式操作接口
8  * </p>
9  *
10 * @author of1610 chenyong
11 * @version 1.0
12 * @since 1.0
13 */
14 public interface PriceProvider {
15
16     /**
17     * 保存价格
18     *
19     * @param request {@link PriceSaveRequest}
20     * @throws ProductException <br>
21     */
22     void savePrice(PriceSaveRequest request) throws ProductException;
23
24     /**
25     * ...
26     */
27     CentreResponse<Boolean> saveBasePrice(BasePriceSaveRequest request) throws ProductException;
28
29     /**
30     * ...
31     */
32     void batchSavePrice(BasePriceBatchSaveRequest request) throws ProductException;
33
34     /**
35     * ...
36     */
37     CentreResponse<ResponseResult<String>> savePriceMode(SpuPriceModeSaveRequest request) throws ProductException;
38
39     /**
40     * ...
41     */
42     CentreResponse<Boolean> saveLevelPrice(SkuLevelPriceSaveRequest request) throws ProductException;
43
44     /**
45     * ...
46     */
47     CentreResponse<Boolean> saveStepPrice(SpuStepPriceSaveRequest request) throws ProductException;
48
49     /**
50     * ...
51     */
52     CentreResponse<Boolean> savePrimePrice(SkuPrimePriceSaveRequest request) throws ProductException;
53
54     /**
55     * ...
56     */
57     void saveEffectivePrice(EffectivePriceSaveRequest request) throws ProductException;
58
59     /**
60     * ...
61     */
62     CentreResponse<Boolean> saveBasePriceAndLevelPrice(SkuLevelPriceBatchSaveRequest request) throws ProductException;
63
64     /**
65     * ...
66     */
67     void adjust(PriceAdjustRequest request) throws ProductException;
68
69     /**
70     * ...
71     */
72     CentreResponse<Boolean> saveSupplyPrice(SkuSupplyPriceSaveRequest request) throws ProductException;
73 }
```

## how > 翻译流程 > 转换 > Java to Ts > Interface转换示例

```
PriceProvider.ts x
11 import {SkuSupplyPriceBatchSaveRequest} from './request/SkuSupplyPriceBatchSaveRequest';
12 import {PriceSaveRequest} from './request/PriceSaveRequest';
13 import {SkuLevelPriceSaveRequest} from './request/SkuLevelPriceSaveRequest';
14 import {SkuPrimePriceSaveRequest} from './request/SkuPrimePriceSaveRequest';
15 import {argumentMap} from 'interpret-util';
16 import {TDubboCallResult, Dubbo} from 'dubbo2.js';
17
18 export interface IPriceProvider {
19
20     savePrice(request: PriceSaveRequest): TDubboCallResult<void>;
21
22     saveBasePrice( request: BasePriceSaveRequest ): TDubboCallResult<CentreResponse<boolean>>;
23
24     batchSavePrice(request: BasePriceBatchSaveRequest): TDubboCallResult<void>;
25
26     savePriceMode( request: SpuPriceModeSaveRequest ): TDubboCallResult<CentreResponse<ResponseResult<string>>>;
27
28     saveLevelPrice(request: SkuLevelPriceSaveRequest): TDubboCallResult<CentreResponse<boolean>>;
29
30     saveStepPrice( request: SpuStepPriceSaveRequest ): TDubboCallResult<CentreResponse<boolean>>;
31
32     savePrimePrice( request: SkuPrimePriceSaveRequest): TDubboCallResult<CentreResponse<boolean>>;
33
34     saveEffectivePrice(request: EffectivePriceSaveRequest): TDubboCallResult<void>;
35
36     saveBasePriceAndLevelPrice( request: SkuLevelPriceBatchSaveRequest): TDubboCallResult<CentreResponse<boolean>>;
37
38     adjust(request: PriceAdjustRequest): TDubboCallResult<void>;
39
40     saveSupplyPrice(request: SkuSupplyPriceSaveRequest ): TDubboCallResult<CentreResponse<boolean>>;
41
42     batchSaveSupplyPrice(request: SkuSupplyPriceBatchSaveRequest): TDubboCallResult<CentreResponse<boolean>>;
43
44 }
45
46 export const PriceProviderWrapper = {saveBasePriceAndLevelPrice: argumentMap...};
60
61 export function PriceProvider(dubbo: Dubbo): IPriceProvider {
62     return dubbo.proxyService<IPriceProvider>({
63         dubboInterface: 'com.gianmi.pc.price.api.PriceProvider',
64         methods: PriceProviderWrapper,
65     });
66 }
```

## how > 翻译流程 > 转换 > Java to Ts > Enum转换示例

```
TagEnum.java x
1 package com.qianmi.pc.item.constant;
2
3 /**
4  * 商品标签枚举
5  * <p>Filename: com.qianmi.pc.item.constant.TagEnum.java</p>
6  * <p>Date: 2017-04-14 13:46.</p>
7  *
8  * @author <a href="mailto:haojingguo@qianmi.com">0F2602-郝敬国</a>
9  * @version v1.0.0
10  *
11  */
12 public enum TagEnum{
13     COMMEND(1),
14     NEW(2),
15     PROMOTION(3),
16     HOT(4);
17
18     private int value;
19
20     private TagEnum(int value) { this.value = value; }
21
22
23     public int getValue() { return this.value; }
24
25
26
27 }
28
```

```
TagEnum.ts x
1 export enum TagEnum {
2     NEW = 1,
3     PROMOTION = 2,
4     HOT = 3,
5     COMMEND = 4,
6 }
7 //generate by interpret-cli dubbo2.js
8
```

## how > 翻译流程 > 转换 > Java to Ts > Bean转换示例

```
BasePriceBatchSaveRequest.java x CMIDBaseRequest.class x PCBaseRequest.class x
1 package com.qianmi.pc.price.api.request;
2
3 import ...
12
13 /**
14  * <p>com.qianmi.pc.price.api.request.BasePriceBatchSaveRequest.java</p>
15  * <p>Date: 2017/12/6</p>
16  *
17  * @author of2182-zdy
18  * @version V0.0.1
19  */
20 @Setter
21 @Getter
22 @NoArgsConstructor
23 public class BasePriceBatchSaveRequest extends CMIDBaseRequest {
24     private static final long serialVersionUID = 2280148562111337431L;
25
26     private Set<BasePriceB0> saveBasePrices;
27
28     @Override
29     public boolean checkParam() {
30         super.checkParam();
31         Validate.isTrue(CollectionUtils.isNotEmpty(saveBasePrices), message: "基本价格不能为空");
32         for (BasePriceB0 bo : saveBasePrices) {
33             Validate.isTrue(expression: bo.getPrice() != null || bo.getCost() != null || bo.getMarketPrice() != null ||
34         }
35
36         return true;
37     }
38 }
39
```

## how > 翻译流程 > 转换 > Java to Ts > Bean转换示例

```
BasePriceBatchSaveRaquest.java x CMIDBaseRequest.class x PCBaseRequest.class x
Decompiled .class file, bytecode version: 51.0 (Java 7) Download Sources Choos
1  + /.../
5
6  package com.qianmi.pc.base.api.request;
7
8  + import ...
10
11  public class CMIDBaseRequest extends PCBaseRequest {
12      private static final long serialVersionUID = 1632802856139921741L;
13      @NonNull
14      private String chainMasterId;
15
16      @ + public CMIDBaseRequest() {
17          }
18
19      + public boolean checkParam() {
20          if (this.chainMasterId != null && this.chainMasterId.length() != 0) {
21              return super.checkParam();
22          } else {
23              throw new IllegalArgumentException("站长编号为空");
24          }
25      }
26
27      + public Operator getOperator() { return super.getOperator().setChainMasterId(this.chainMasterId); }
30
31      @ + public void setChainMasterId(@NonNull String chainMasterId) {
32          if (chainMasterId == null) {
33              throw new NullPointerException("chainMasterId");
34          } else {
35              this.chainMasterId = chainMasterId;
36          }
37      }
38
39      @NonNull
40      + public String getChainMasterId() { return this.chainMasterId; }
43  }
44
```

# how > 翻译流程 > 转换 > Java to Ts > Bean转换示例

```
BasePriceBatchSaveRequest.java x CMIDBaseRequest.class x PCBaseRequest.class x
Decompiled .class file, bytecode version: 51.0 (Java 7) Download Sources Choos
1  + /.../
5
6  package com.qianmi.pc.base.api.request;
7
8  + import ...
11
12  public class PCBaseRequest extends CentreRequest {
13      private static final long serialVersionUID = -5729255926899808174L;
14      private String optUserCode;
15      private String optUserName;
16
17      @ public PCBaseRequest() {
18      }
19
20      public int getMaxLimitSize() { return 100; }
23
24      public <T> void limitMaxSize(Collection<T> ds) { this.limitMaxSize(ds, this.getMaxLimitSize()); }
27
28      public <T> void limitMaxSize(Collection<T> ds, int maxSize) {
29          this.limitMaxSize((Collection)ds, maxSize, (String)null);
30      }
31
32      public <T> void limitMaxSize(Collection<T> ds, String errorMessage) {
33          this.limitMaxSize(ds, this.getMaxLimitSize(), errorMessage);
34      }
35
36      @ public <T> void limitMaxSize(Collection<T> ds, int maxSize, String errorMessage) {
37          BatchAsserts.limitMaxSize(ds, maxSize, errorMessage);
38      }
39
40      public <T> void limitMaxSize(T[] ds) { this.limitMaxSize(ds, this.getMaxLimitSize()); }
43
44      public <T> void limitMaxSize(T[] ds, int maxSize) { this.limitMaxSize((Object[])ds, maxSize, (String)null); }
47
48      public <T> void limitMaxSize(T[] ds, String errorMessage) {
49          this.limitMaxSize(ds, this.getMaxLimitSize(), errorMessage);
50      }
51
52      public <T> void limitMaxSize(T[] ds, int maxSize, String errorMessage) {
53          BatchAsserts.limitMaxSize(ds, maxSize, errorMessage);
54      }
55
56      public boolean checkParam() { return true; }
```

## how > 翻译流程 > 转换 > Java to Ts > Bean转换示例

```
BasePriceBatchSaveRequest.ts x
1  import {BasePriceB0} from './bean/BasePriceB0';
2  import java from 'js-to-java';
3
4  export interface IBasePriceBatchSaveRequest {
5    saveBasePrices?: Array<BasePriceB0>;
6    chainMasterId?: string;
7    optUserCode?: string;
8    optUserName?: string;
9  }
10
11 export class BasePriceBatchSaveRequest {}
12 constructor(params: IBasePriceBatchSaveRequest) {
13   this.saveBasePrices = params.saveBasePrices;
14   this.chainMasterId = params.chainMasterId;
15   this.optUserCode = params.optUserCode;
16   this.optUserName = params.optUserName;
17 }
18
19 saveBasePrices?: Array<BasePriceB0>;
20 chainMasterId?: string;
21 optUserCode?: string;
22 optUserName?: string;
23
24 __fields2java() {
25   return {
26     $class: 'com.gianmi.pc.price.api.request.BasePriceBatchSaveRequest',
27     $: {
28       saveBasePrices: java.Set(
29         (this.saveBasePrices || []).map(paramItem => {
30           return paramItem.__fields2java();
31         }),
32       ),
33       chainMasterId: java.String(this.chainMasterId),
34       optUserCode: java.String(this.optUserCode),
35       optUserName: java.String(this.optUserName),
36     },
37   };
38 }
39 }
40
41 //generate by interpret-cli dubbo2.js
42
```



how > 翻译流程 > 转换 > js to hessian

# 参数-转换

- bean对象转换为hessian Param

## how > 翻译流程 > 转换 > js to hessian > hessian标准参数

```
var encoder = new Encoder();
var long = {
  $class: 'java.lang.Long',
  $: 1
}
encoder.write(long); // long type

var testObject = {
  $class: 'com.hessian.TestObject',
  $: {
    a: 1,
    b: 'test',
    c: {$class: 'java.lang.Long', $: 123}
  }
};
encoder.write(testObject);
```

## how > 翻译流程 > 转换 > js to hessian

```
Makefile x BasePriceBatchSaveRequest.ts x
1 import {BasePriceB0} from './bean/BasePriceB0';
2 import java from 'js-to-java';
3
4 export interface IBasePriceBatchSaveRequest {
5   saveBasePrices?: Array<BasePriceB0>;
6   chainMasterId?: string;
7   optUserCode?: string;
8   optUserName?: string;
9 }
10
11 export class BasePriceBatchSaveRequest {
12   constructor(params: IBasePriceBatchSaveRequest) {
13     this.saveBasePrices = params.saveBasePrices;
14     this.chainMasterId = params.chainMasterId;
15     this.optUserCode = params.optUserCode;
16     this.optUserName = params.optUserName;
17   }
18
19   saveBasePrices?: Array<BasePriceB0>;
20   chainMasterId?: string;
21   optUserCode?: string;
22   optUserName?: string;
23
24   __fields2java() {
25     return {
26       $class: 'com.gianmi.pc.price.api.request.BasePriceBatchSaveRequest',
27       $: {
28         saveBasePrices: java.Set(
29           (this.saveBasePrices || []).map(paramItem => {
30             return paramItem.__fields2java();
31           }),
32         ),
33         chainMasterId: java.String(this.chainMasterId),
34         optUserCode: java.String(this.optUserCode),
35         optUserName: java.String(this.optUserName),
36       },
37     };
38   }
39 }
40
41 //generate by interpret-cli dubbo2.js
```

## how > 翻译流程 > 转换 > js to hessian

```
BasePriceBatchSaveRequest.ts x PriceProvider.ts x package.json x lerna.json x
BasePriceBatchSaveRequest
Match Case

1 import {SkuLevelPriceBatchSaveRequest} from './request/SkuLevelPriceBatchSaveRequest';
2 import {CentreResponse} from '../../base/response/CentreResponse';
3 import {BasePriceSaveRequest} from './request/BasePriceSaveRequest';
4 import {EffectivePriceSaveRequest} from './request/EffectivePriceSaveRequest';
5 import {SpuPriceModeSaveRequest} from './request/SpuPriceModeSaveRequest';
6 import {ResponseResult} from '../../base/api/response/ResponseResult';
7 import {SkuSupplyPriceSaveRequest} from './request/SkuSupplyPriceSaveRequest';
8 import {PriceAdjustRequest} from './request/PriceAdjustRequest';
9 import {BasePriceBatchSaveRequest} from './request/BasePriceBatchSaveRequest';
10 import {SpuStepPriceSaveRequest} from './request/SpuStepPriceSaveRequest';
11 import {SkuSupplyPriceBatchSaveRequest} from './request/SkuSupplyPriceBatchSaveRequest';
12 import {PriceSaveRequest} from './request/PriceSaveRequest';
13 import {SkuLevelPriceSaveRequest} from './request/SkuLevelPriceSaveRequest';
14 import {SkuPrimePriceSaveRequest} from './request/SkuPrimePriceSaveRequest';
15 import {argumentMap} from 'interpret-util';
16 import {TDubboCallResult, Dubbo} from 'dubbo2.js';
17
18 export interface IPriceProvider {
45
46 export const PriceProviderWrapper = {
47   saveBasePriceAndLevelPrice: argumentMap,
48   saveBasePrice: argumentMap,
49   saveEffectivePrice: argumentMap,
50   savePriceMode: argumentMap,
51   saveSupplyPrice: argumentMap,
52   adjust: argumentMap,
53   batchSavePrice: argumentMap,
54   saveStepPrice: argumentMap,
55   batchSaveSupplyPrice: argumentMap,
56   savePrice: argumentMap,
57   saveLevelPrice: argumentMap,
58   savePrimePrice: argumentMap,
59 };
60
61 export function PriceProvider(dubbo: Dubbo): IPriceProvider {
62   return dubbo.proxyService<IPriceProvider>({
63     dubboInterface: 'com.gianmi.pc.price.api.PriceProvider',
64     methods: PriceProviderWrapper,
65   });
66 }
67
68 //generate by interpret-cli dubbo2.js
69
```

# how > 翻译流程 > 转换 > js to hessian

```
16  */
17  import debug from "debug";
18
19  const log = debug({ namespace: "j2t:core:paramEnhance" });
20
21  export function argumentMap() {
22    let _arguments = Array.from(arguments);
23
24    return _arguments.map(
25      callbackfn: argumentItem =>
26        argumentItem.__fields2java
27          ? paramEnhance(argumentItem.__fields2java())
28          : argumentItem
29    );
30  }
31
32
33  //删除对象中包含undefined 与null的值情况
34  function paramEnhance(javaParams: Array<object> | object) {
35    if (javaParams instanceof Array) {
36      for (let i = 0, ilen = javaParams.length; i < ilen; i++) {
37        let itemParam = javaParams[i];
38        minusRedundancy(itemParam);
39      }
40    } else {
41      minusRedundancy(javaParams);
42    }
43    return javaParams;
44  }
45
46  function minusRedundancy(itemParam: any) {
47    if (!itemParam) {
48      return;
49    }
50    for (var _key in itemParam.$) {
51      if (itemParam.$[_key] === null || itemParam.$[_key] === undefined) {
52        delete itemParam.$[_key];
53        log("删除 key %s from %j ", itemParam, _key);
54      }
55    }
56  }
57
58  export type JavaString=Object;
59  export type JavaBoolean=Object;
```

how >

# I come , i conquer !

0.抽取ast

from sourceCode

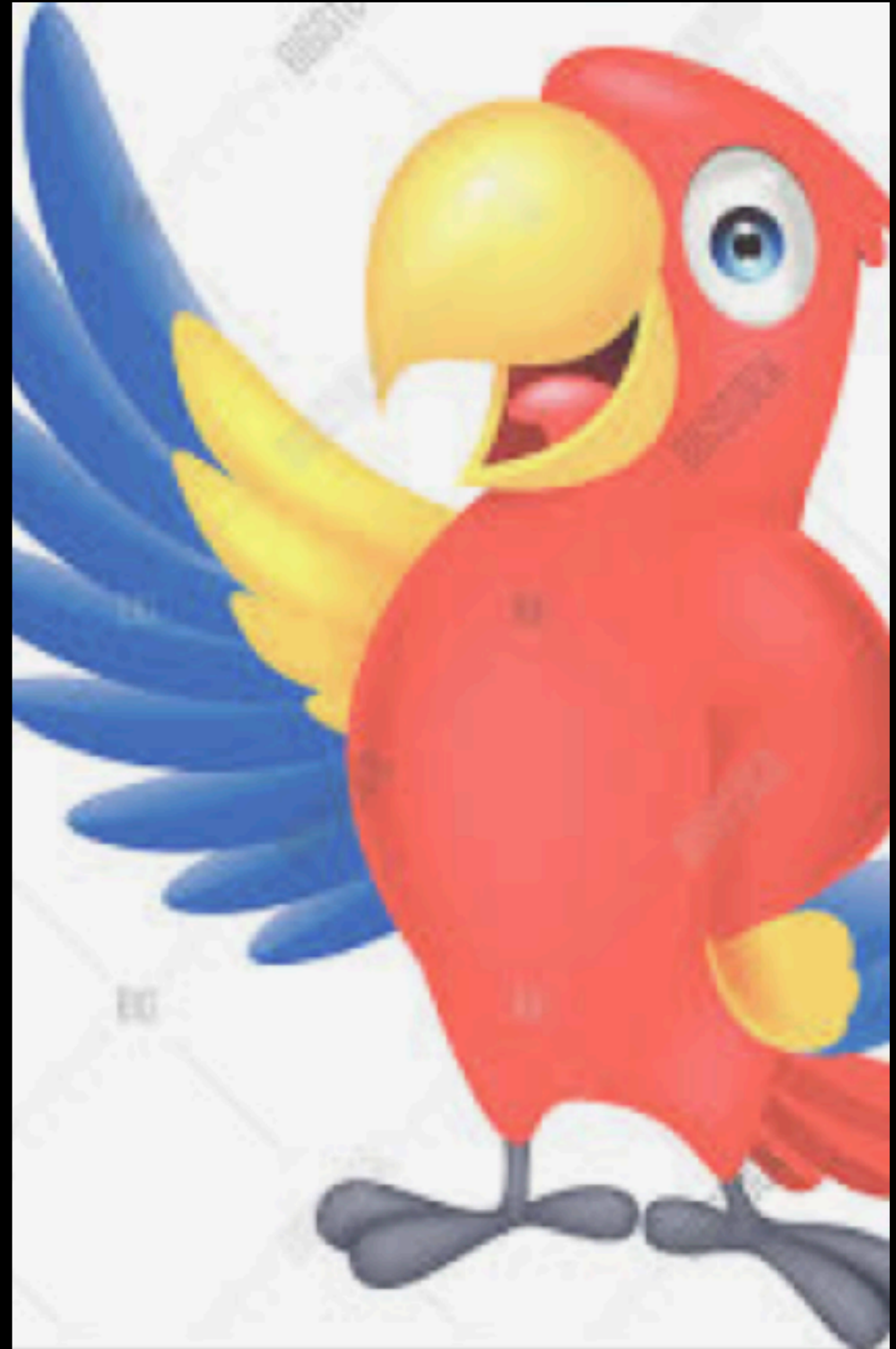
from byteCode

1.java接口转换

java => typescript

2.参数转换

javascript => hession





翻译师  $\Rightarrow$  **Error**

Error > 命名冲突

命名冲突



# Error > 命名冲突

```
import {CentreResponse} from './../../../../base/response/CentreResponse';
import {EmployeeExperience} from './../../../../bo/user/employee/EmployeeExperience';
import {EmployeeQueryRequest} from './../../../../bo/user/employee/EmployeeQueryRequest';
import {CentreCutPageResponse} from './../../../../base/response/CentreCutPageResponse';
import {EmployeeInfo} from './../../../../bo/user/employee/EmployeeInfo';
import {EmployeeEvaluateQueryRequest} from './../../../../bo/user/employee/EmployeeEvaluateQueryRequest';
import {WarehouseEmployeeQueryRequest} from './../../../../bo/user/employee/WarehouseEmployeeQueryRequest';
import {EmployeeInviteRequest} from './../../../../bo/user/employee/EmployeeInviteRequest';
import {EnumRole} from './../../../../bo/user/util/EnumRole';
import {EmployeeInfo} from './../../../../bo/user/message/EmployeeInfo';
import {EmployeeRoleCountInfo} from './../../../../bo/user/employee/EmployeeRoleCountInfo';
import {EmployeeBaseInfo} from './../../../../bo/user/employee/EmployeeBaseInfo';
import {CentreListResponse} from './../../../../base/response/CentreListResponse';
import {EmployeeDetailInfo} from './../../../../bo/user/employee/EmployeeDetailInfo';
import {EmployeeRegRequest} from './../../../../bo/user/employee/EmployeeRegRequest';
import {argumentMap, JavaString, JavaList} from 'interpret-util';
import {TDubboCallResult, Dubbo} from 'dubbo2.js';

export interface IEmployeeProvider {
  addEvaluate(EmployeeEvaluate0: EmployeeEvaluate): TDubboCallResult<void>;
  updateEmployeeWarehouseRef(
    String0: JavaString,
    List1: JavaList,
  ): TDubboCallResult<void>;
  deleteExperience(
    String0: JavaString,
    String1: JavaString,
  ): TDubboCallResult<void>;
  cancelGroupLeader(
    String0: JavaString,
    String1: JavaString,
    String2: JavaString,
  ): TDubboCallResult<void>;
  queryEmployeeRoleId(
    String0: JavaString,
    String1: JavaString,
    String2: JavaString,
    String3: JavaString,
  ): TDubboCallResult<Array<string>>;
  delWarehouseRef(List0: JavaList): TDubboCallResult<void>;
  queryEmployeeExperienceById(
    String0: JavaString,
    String1: JavaString,
    String2: JavaString,
  ): TDubboCallResult<CentreResponse<EmployeeExperience>>;
  queryEmployeeListByRoleBnameList(
    EmployeeQueryRequest0: EmployeeQueryRequest,
  ): TDubboCallResult<CentreCutPageResponse<EmployeeInfo>>;
}
```

EmployeeProvider > cancelGroupLeader()

# Error > 命名冲突

The screenshot shows an IDE window with a project structure on the left and a code editor on the right. The project structure includes a folder named 'usercenter-api' under 'packages'. The code editor displays the file 'EmployeeProvider.ts' with a search for 'EmployeeInfo' resulting in 13 matches. The code includes several imports and an export interface definition.

```
1 import {EmployeeEvaluate} from '../../../../bo/user/employee/EmployeeEvaluate';
2 import {EmployeeWarehouse} from '../../../../bo/user/employee/EmployeeWarehouse';
3 import {CentreResponse} from '../../../../base/response/CentreResponse';
4 import {EmployeeExperience} from '../../../../bo/user/employee/EmployeeExperience';
5 import {EmployeeQueryRequest} from '../../../../bo/user/employee/EmployeeQueryRequest';
6 import {CentreCutPageResponse} from '../../../../base/response/CentreCutPageResponse';
7 import {EmployeeInfo} from '../../../../bo/user/employee/EmployeeInfo';
8 import {EmployeeEvaluateQueryRequest} from '../../../../bo/user/employee/EmployeeEvaluateQueryRequest';
9 import {WareHouseEmployeeQueryRequest} from '../../../../bo/user/employee/WareHouseEmployeeQueryRequest';
10 import {EmployeeInviteRequest} from '../../../../bo/user/employee/EmployeeInviteRequest';
11 import {EnumRole} from '../../../../bo/user/util/EnumRole';
12 import {EmployeeInfo as EmployeeInfo0} from '../../../../bo/user/message/EmployeeInfo';
13 import {EmployeeRoleCountInfo} from '../../../../bo/user/employee/EmployeeRoleCountInfo';
14 import {EmployeeBaseInfo} from '../../../../bo/user/employee/EmployeeBaseInfo';
15 import {CentreListResponse} from '../../../../base/response/CentreListResponse';
16 import {EmployeeDetailInfo} from '../../../../bo/user/employee/EmployeeDetailInfo';
17 import {EmployeeRegRequest} from '../../../../bo/user/employee/EmployeeRegRequest';
18 import {ArgumentMap, JavaString, JavaList} from 'interpret-util';
19 import {TDubboCallResult, Dubbo} from 'dubbo2.js';
20
21 export interface IEmployeeProvider {
22   addEvaluate(EmployeeEvaluate0: EmployeeEvaluate): TDubboCallResult<void>;
23   updateEmployeeWarehouseRef(
24     String0: JavaString,
25     List1: JavaList,
26   ): TDubboCallResult<void>;
27   deleteExperience(
28     String0: JavaString,
29     String1: JavaString,
30   ): TDubboCallResult<void>;
31   cancelGroupLeader(
32     String0: JavaString,
33     String1: JavaString,
34     String2: JavaString,
35   ): TDubboCallResult<void>;
36   queryEmployeeRoleId(
37     String0: JavaString,
38     String1: JavaString,
39     String2: JavaString,
40     String3: JavaString,
41   ): TDubboCallResult<Array<string>>;
42   deleteWarehouseRef(
43     String0: JavaString,
44     List1: JavaList,
45   ): TDubboCallResult<void>;
46 }
```

Error >override

**OverRide**

# Error > override

The screenshot shows an IDE window for the file `SpuProvider.java`. The project structure on the left includes `pc-micro-item`, `pc-item-api`, and `pc-item-app`. The main editor displays the following code:

```
108 * </p>
109 * 场景: 业务线设置商品类目
110 *
111 * @param request {@link SpuBindingCategoriesRequest}
112 * @throws ProductException <br/>
113 *           002500000002 : 参数异常<br>
114 *           002500009999 : 其他异常<br>
115 *           002500000000 : 系统异常<br>
116 */
117 CentreResponse<Boolean> bindingCategories(SpuBindingCategoriesRequest request) throws ProductException;
118
119 /**
120 * <p>
121 * 渠道商品绑定展示类目
122 * </p>
123 * 场景: 业务线设置商品类目
124 *
125 * @param request {@link SpuBindingCategoriesRequest}
126 * @throws ProductException <br/>
127 *           002500000002 : 参数异常<br>
128 *           002500009999 : 其他异常<br>
129 *           002500000000 : 系统异常<br>
130 */
131 CentreResponse<Boolean> bindingCategories(BatchSpuBindingCategoriesRequest request) throws ProductException;
132
133 /**
134 * 保存SPU属性参数
135 *
136 * @param request {@link SpuAttributeSaveRequest}
137 * @return true:成功
138 */
139 CentreResponse<Boolean> saveAttribute(SpuAttributeSaveRequest request);
140
141 /**
142 * 保存商品描述信息
143 *
144 * @param request {@link SpuDescriptionSaveRequest}
145 * @return true:成功
146 */
147 CentreResponse<Boolean> saveDescription(SpuDescriptionSaveRequest request);
148
149 /**
150 * 保存手机端详情
151 *
152 * @param request {@link SpuDescriptionSaveRequest}
```

The error message "Override method" is highlighted in a red box, indicating that the two `bindingCategories` methods have the same signature and are defined in the same class, which is not allowed in Java.

# Error >override

```
SpuProvider.ts x Makefile x SpuBindingCategoriesRequest.ts x
bindingCategories
19 import {ResponseResult} from '../../../base/api/respons
20 import {SpuShieldAreaIdsRequest} from './request/SpuShiel
21 import {BatchSpuBindingCategoriesRequest} from './request
22 import {argumentMap} from 'interpret-util';
23 import {TDubboCallResult, Dubbo} from 'dubbo2.js';
24
25 export interface ISpuProvider {
26   add(request: SpuAddRequest): TDubboCallResult<CentreRes
27   setShieldUsers(
28     request: SpuShieldUserIdsRequest,
29   ): TDubboCallResult<CentreResponse<boolean>>;
30   bindingCategories(
31     request: SpuBindingCategoriesRequest,
32   ): TDubboCallResult<CentreResponse<boolean>>;
33   bindingCategories(
34     request: BatchSpuBindingCategoriesRequest,
35   ): TDubboCallResult<CentreResponse<boolean>>;
36   addStick(
37     request: SpuAddStickRequest,
38   ): TDubboCallResult<CentreResponse<boolean>>;
39   update(
40     productModifyRequest: SpuUpdateRequest,
41   ): TDubboCallResult<CentreResponse<boolean>>;
42   recycling(
43     request: SpuRecyclingRequest,
44   ): TDubboCallResult<CentreResponse<Array<string>>>;
45   delisting(request: SpuDelistingRequest): TDubboCallResu
46   recovery(
47     request: SpuRecoveryRequest,
48   ): TDubboCallResult<CentreResponse<Array<string>>>;
16 import {SpuAttributeSaveRequest} from './request/SpuAttributeSaveReque
17 import {BindingSupplierRequest} from './request/BindingSupplierRequest
18 import {SpuListingRequest} from './request/SpuListingRequest';
19 import {ResponseResult} from '../../../base/api/response/ResponseRes
20 import {SpuShieldAreaIdsRequest} from './request/SpuShieldAreaIdsReque
21 import {argumentMap} from 'interpret-util';
22 import {TDubboCallResult, Dubbo} from 'dubbo2.js';
23
24 export interface ISpuProvider ...
85
86 export const SpuProviderWrapper = {
87   add: argumentMap,
88   setShieldUsers: argumentMap,
89   addStick: argumentMap,
90   update: argumentMap,
91   recycling: argumentMap,
92   delisting: argumentMap,
93   recovery: argumentMap,
94   saveDescription: argumentMap,
95   delete: argumentMap,
96   cancelSticky: argumentMap,
97   setShieldLevels: argumentMap,
98   setOverSold: argumentMap,
99   bindingCategories: argumentMap,
100   saveMobileDescription: argumentMap,
101   dragStick: argumentMap,
102   saveAttribute: argumentMap,
103   bindingOfflineSupplier: argumentMap,
104   listing: argumentMap,
105   setShieldAreas: argumentMap,
106   unBundlingOfflineSupplier: argumentMap,
107 };
108
109 export function SpuProvider(dubbo: Dubbo): ISpuProvider {
110   return dubbo.proxyService<ISpuProvider>({
111     dubboInterface: 'com.gianmi.pc.item.api.spu.SpuProvider',
112     methods: SpuProviderWrapper,
113   });
114 }
115
116 //generate by interpret-cli dubbo2.js
```

## Error >override

```
SpuProvider.ts x SpuBindingCategoriesRequest.ts x
5   chainMasterId?: string;
6   categoryIds?: Array<string>;
7   channelEnum?: ChannelEnum;
8   optUserCode?: string;
9   spuId?: string;
10  optUserName?: string;
11  }
12
13  export class SpuBindingCategoriesRequest {
14  constructor(params: ISpuBindingCategoriesRequest) {
15    this.chainMasterId = params.chainMasterId;
16    this.categoryIds = params.categoryIds;
17    this.channelEnum = params.channelEnum;
18    this.optUserCode = params.optUserCode;
19    this.spuId = params.spuId;
20    this.optUserName = params.optUserName;
21  }
22
23  chainMasterId?: string;
24  categoryIds?: Array<string>;
25  channelEnum?: ChannelEnum;
26  optUserCode?: string;
27  spuId?: string;
28  optUserName?: string;
29
30  __fields2java() {
31    return {
32      $class: 'com.gianmi.pc.item.api.spu.request.SpuBindingCategoriesRequest',
33      $: {
34        chainMasterId: java.String(this.chainMasterId),
35        categoryIds: java.Set(
36          (this.categoryIds || []).map(paramItem => {
37            return java.String(paramItem);
38          }),
39        ),
40        channelEnum: java['enum'](
41          'com.gianmi.pc.base.api.constants.ChannelEnum',
42          ChannelEnum[this.channelEnum],
43        ),
44        optUserCode: java.String(this.optUserCode),
45        spuId: java.String(this.spuId),
46        optUserName: java.String(this.optUserName),
47      },
48    };
49  }
```

Error >override

**当基本类型遇上overRide**

# Error > override

The screenshot shows an IDE with a project structure on the left and a TypeScript file on the right. The project structure includes a folder named 'lib' under 'acct-api'. The TypeScript file, 'BankBatchTransProvider.ts', defines an interface 'IBankBatchTransProvider' and a function 'BankBatchTransProvider'.

```
1  import ...
2
3  export interface IBankBatchTransProvider {
4    modifyBatchOrdersTask(
5      BankBatchQueueBO0: BankBatchQueueBO,
6    ): TDubboCallResult<boolean>;
7    batchTransfer(
8      String0: JavaString,
9      List1: JavaList,
10   ): TDubboCallResult<Array<BankTransReturnBO>>;
11   batchTransfer(
12     String0: JavaString,
13     List1: JavaList,
14   ): TDubboCallResult<Array<BankTransReturnBO>>;
15   getBatchQueues(
16     BankBatchQueueBO0: BankBatchQueueBO,
17   ): TDubboCallResult<Array<BankBatchQueueBO>>;
18   removeBankBatchQueue(String0: JavaString): TDubboCallResult<boolean>;
19   modifyBatchOrdersTask(
20     BankBatchQueueBO0: BankBatchQueueBO,
21   ): TDubboCallResult<boolean>;
22 }
23
24 export const BankBatchTransProviderWrapper = {
25   modifyBatchOrdersTask: argumentMap,
26   batchTransfer: argumentMap,
27   getBatchQueues: argumentMap,
28   removeBankBatchQueue: argumentMap,
29 };
30
31 export function BankBatchTransProvider(dubbo: Dubbo): IBankBatchTransProvider {
32   return dubbo.proxyService<IBankBatchTransProvider>({
33     dubboInterface: 'com.qipay.acct.batch.bank.provider.BankBatchTransProvider',
34     methods: BankBatchTransProviderWrapper,
35   });
36 }
```



# Error > override

The screenshot shows an IDE window with the following breadcrumb: `dubbo-providers > packages > acct-api > src > com > ofpay > acct > pay > provider > PayGateWayProvider.ts`. The left sidebar shows a project tree with the following structure:

- dubbo-providers (~/.workbench/qmfe)
  - node\_modules library root
  - packages
    - acct-api
    - cloudshop-platform-consumer
    - d2p-cart-api
    - d2p-common-service-api
    - mc-api
    - pc-cats-api
    - pc-item-api
    - pc-price-api
    - qianmi-pc-api
    - uc-qstore-api
    - uc-quser-api
    - usercenter-api
    - yxtc-consumer
  - .gitignore
  - lerna.json
  - lerna-debug.log
  - Makefile
  - npm-debug.log
  - package.json
  - package-lock.json
  - README.md
  - tsconfig.json
  - yarn-error.log
- External Libraries
- Scratches and Consoles

The main editor displays the following TypeScript code:

```
1  import ...
7
8  export interface IPayGateWayProvider {
9
10     calculateGateCommission(
11         String0: JavaString,
12     ): TDubboCallResult<PayMoneyInfoB0>;
13     calculateGateCommission(
14         String0: JavaString,
15         String1: JavaString,
16     ): TDubboCallResult<PayMoneyInfoB0>;
17     bankPayByOrderNo(
18         String0: JavaString,
19         String1: JavaString,
20         String2: JavaString,
21     ): TDubboCallResult<string>;
22     getBankHtml(
23         String0: JavaString,
24         String1: JavaString,
25         String2: JavaString,
26     ): TDubboCallResult<string>;
27     getBankHtml(TransBankB10: TransBankB0): TDubboCallResult<string>;
28     getBankHtml(
29         String0: JavaString,
30         String1: JavaString,
31         String2: JavaString,
32         String3: JavaString,
33     ): TDubboCallResult<string>;
34     payment2Bank(
35         String0: JavaString,
36         String1: JavaString,
37         String2: JavaString,
38         String3: JavaString,
39     ): TDubboCallResult<string>;
40     tradeReverse(String0: JavaString): TDubboCallResult<string>;
41     getChargeBankHtml(
42         String0: JavaString,
43         String1: JavaString,
44     ): TDubboCallResult<string>;
45     getTradeBankHtml(
46         String0: JavaString,
47         String1: JavaString,
48         String2: JavaString,
49     ): TDubboCallResult<string>;
50     tradeQuery(String0: JavaString): TDubboCallResult<string>;
51     bankPayByOrderNo(
52         String0: JavaString,
53         String1: JavaString,
54     ): TDubboCallResult<string>;
```

# Error > override

The screenshot shows the VS Code editor interface. The left sidebar displays the project structure for 'd2p-visitor', with the file 'pay.ts' selected under the path 'packages/bff/ts/app/service'. The main editor area shows the following TypeScript code:

```
75     value: '0.00'  
76   },  
77   payType: EnumPayType.OLP,  
78   bankCode: '0850'  
79   });  
80  
81   // 先调担保接口 (账务要求)  
82   await this.app.dubbo.paymentProvider.guaranteePayment(  
83     paymentB0,  
84     paymentMethod,  
85     java.String(userId),  
86     java.enum(  
87       'com.guiguang.pc.base.api.constants.ChannelEnum',  
88       'YX'),  
89   );  
90  
91   // 获取拉起微信支付的参数 0850表示小程序支付  
92   const {err, res} = await this.app.dubbo.payGatewayProvider.payment2Bank(  
93     java.String(tid),  
94     java.String("0850"),  
95     java.String(openId),  
96     java.String("2")  
97   );  
98  
99   return res  
100 }  
101 }  
102
```

Error >override

为什么不翻译

## Error >override

```
OverrideProvider.java x
1 package com.qianmi.pc.price.api;
2
3 import com.qianmi.pc.base.api.exception.ProductException;
4 import com.qianmi.pc.price.api.request.*;
5
6 import java.util.Collection;
7 import java.util.List;
8 import java.util.Map;
9 import java.util.Set;
10
11 public interface OverrideProvider {
12
13     void hello0(BasePriceSaveRequest request) throws ProductException;
14     void hello0(List<BasePriceSaveRequest> request) throws ProductException;
15
16     void hello2(String request,String num) throws ProductException;
17     void hello2(String request,Integer num) throws ProductException;
18
19     //Error: 不能翻译了 对以下情况做不了区分.
20     void hello3(String request,Integer num) throws ProductException;
21     void hello3(String request,Float num) throws ProductException;
22
23     void hello1(BasePriceSaveRequest priceRequest,List<BasePriceSaveRequest> request) throws ProductException;
24     void hello1(BasePriceSaveRequest priceRequest,Collection<BasePriceSaveRequest> request) throws ProductException;
25     void hello1(BasePriceSaveRequest priceRequest,Set<BasePriceSaveRequest> request) throws ProductException;
26     void hello1(BasePriceSaveRequest priceRequest,Map<String,BasePriceSaveRequest> request) throws ProductException;
27 }
28
```

# 抽象类参数

## Error > 抽象类参数

```
PaymentProvider.java x PaymentMethod.java x
1 package com.ofpay.acct.trade.provider;
2
3 import ...
12
13 public interface PaymentProvider {
14
15     /**
16      * sale订单付款, dubbo接口仅提供信用点、积分等欧飞余额支付方式的组合付款或单类型付款<br/>
17      * 积分不同步产生, 由后置任务异步产生积分。
18      *
19      * @param paymentBO 付款主体
20      * @param paymentMethod 付款方式、金额信息
21      * @return
22      * @throws BaseException
23      * @author Miner
24      * @date 2013-6-18 下午09:23:18
25      */
26     PayResultBO payment(PaymentBO paymentBO, PaymentMethod paymentMethod) throws BaseException;
27
28
29     /**
30      * 函数功能: esup回款流程
31      *
32      * @param orderNo
33      * @param orderSys
34      * @return
35      * @throws BaseException
36      * @author Miner 2012-10-30 下午01:54:18
37      */
38     boolean esupCreditBack(String orderNo, String orderSys) throws BaseException;
39
40     /**
41      * 函数功能: 撤销esup回款
42      *
43      * @param orderNo
44      * @param orderSys
45      * @return
46      * @throws BaseException
47      * @author Miner 2012-11-21 下午03:04:45
48      */
49     boolean reverseEsupCreditBack(String orderNo, String orderSys) throws BaseException;
50
```

# Error > 抽象类参数

The screenshot displays an IDE window with the following components:

- Project Structure (Left):** A tree view showing the project hierarchy: `trade` > `bo` > `PaymentMethod`. The `PaymentMethod` class is highlighted in blue.
- Code Editor (Center):** The source code for `PaymentMethod.java`. The code is as follows:

```
1 package com.ofpay.acct.trade.bo.payment;
2
3 import ...
4
5 /**
6  * 付款方式信息抽象类, 包括基础付款方式信息
7  *
8  * @author Miner
9  * @date 2013-6-18 下午07:18:01
10 */
11 public abstract class PaymentMethod extends BaseBean implements Cloneable {
12
13     /**
14      * 付款类型
15      */
16     protected EnumPayType payType;
17
18     /**
19      * 支付金额
20      */
21     private BigDecimal amount;
22
23     /**
24      * 使用的银行编码
25      */
26     private String bankCode;
27
28     public BigDecimal getAmount() { return amount; }
29
30     public void setAmount(BigDecimal amount) { this.amount = amount; }
31
32     public EnumPayType getPayType() { return payType; }
33
34     @Override
35     public void formatAmount() {
36         if (!isFormat()) {
37             amount = MoneyUtil.YuanToSysUnit(amount);
38             setFormat(true);
39         }
40     }
41
42     public boolean validate() throws BaseException {
43         if (amount == null) {
44             throw new BaseException(BaseErrorCode.ARGUMENTS_NULL);
45         }
46     }
47 }
```
- Bottom Panel:** Shows the class name `PaymentMethod`.

Error > 泛型

泛型?



## Error > 泛型

```
D2PTradeModifyRequest.java x TradeV2ModifyRequest.java x ItemCreateRequest.java x
30
31 /**
32  * 物流总额
33  */
34 private BigDecimal logisticsFee;
35
36 /**
37  * 收货人地址
38  *
39  * @see ConsigneeInfo
40  */
41 private ConsigneeInfo consigneeInfo;
42
43 /**
44  * 期望交货日期
45  */
46 private Date expectDeliverDate;
47
48 /**
49  * 申请特价
50  */
51 private BigDecimal bargainPrice;
52
53 /**
54  * 订单附件
55  */
56 private List<String> attachFile;
57
58 /**
59  * 新的商品单列表
60  */
61 private List<ItemCreateRequest> itemRequestList;
62
63 /**
64  * 开发票时收取的费用
65  */
66 private BigDecimal invoiceFee;
67
68 private String payTypeId;
69
70 @Override
71 public boolean checkParam() {...}
93
94 public String getLeaveMessage() { return leaveMessage; }
97
98 public void setLeaveMessage(String leaveMessage) { this.leaveMessage = leaveMessage; }
101
```

## Error > 泛型

```
D2PTradeModifyRequest.java x TradeV2ModifyRequest.java x ItemCreateRequest.java x
1 package com.qianmi.yxtc.api.request;
2
3 import ...
9
10 /**
11  * <p>
12  * 订单商品单信息
13  * </p>
14  *
15  * @author OF046 Liuweibin
16  * @since 2.0
17  * @version 2.0
18  *
19 public class ItemCreateRequest<T extends MarketingCreateRequest> extends CentreRequest {
20
21     private static final long serialVersionUID = -2910911811932097113L;
22
23     /**
24      * 商品单编号
25      */
26     private String oid;
27     /**
28      * 商品名称
29      */
30     private String itemName;
31
32     /**
33      * sku编号
34      */
35     private String skuId;
36
37     /**
38      * spu编号
39      */
40     private String spuId;
41
42     /**
43      * 商品数量
44      */
45     private BigDecimal itemNumber;
46     /**
47      * 真实价(即出参的pervalue)
48      */
49     private BigDecimal levelPrice;
50     /**
51      * 成本价
52      */
```

## Error > 泛型

```
D2PTradeModifyRequest.ts
Q-
Match Case

35     this.payTypeId = params.payTypeId;
36     this.leaveMessage = params.leaveMessage;
37     this.consigneeInfo = params.consigneeInfo;
38     this.optUserCode = params.optUserCode;
39     this.logisticsFee = params.logisticsFee;
40     this.expectDeliverDate = params.expectDeliverDate;
41     this.itemRequestList = params.itemRequestList;
42 }
43
44 requestUserCode?: string;
45 attachFile?: Array<string>;
46 chainMasterCode?: string;
47 requestIp?: string;
48 optUserName?: string;
49 invoiceFee?: {value: string};
50 tid?: string;
51 bargainPrice?: {value: string};
52 serialVersionUID?: number;
53 payTypeId?: string;
54 leaveMessage?: string;
55 consigneeInfo?: ConsigneeInfo;
56 optUserCode?: string;
57 logisticsFee?: {value: string};
58 expectDeliverDate?: Date;
59 itemRequestList?: Array<ItemCreateRequest>;
60
61 __fields2java() {
62     return {
63         $class: 'com.gianmi.yxtc.api.d2pv2.request.D2PTradeModifyRequest',
64         $: {
65             requestUserCode: java.String(this.requestUserCode),
66             attachFile: java.List(
67                 (this.attachFile || []).map(paramItem => {
68                     return java.String(paramItem);
69                 }),
70             ),
71             chainMasterCode: java.String(this.chainMasterCode),
```

## Error > 泛型

```
BaseListBO.java x
1 package com.ofpay.comm.base;
2
3 import ...
4
5
6
7
8
9 public class BaseListBO implements Serializable {
10
11     private static final long serialVersionUID = -7845365502476716789L;
12     // 当前页序号
13     private int pageno = 0;
14     // 每页显示数量
15     private int pagesize = 10;
16     // 总条数
17     private int totalcount = 0;
18     // 列表
19     private List<?> datalist = null;
20     // map
21     private Map<?, ?> datamap = null;
22
23     /**
24      * pageno
25      *
26      * @return the pageno
27      * @since CodingExample Ver(编码范例查看) 1.0
28      */
29
30     public int getPageno() { return pageno; }
31
32
33
34     /**
35      * @param pageno the pageno to set
36      */
37     public void setPageno(int pageno) { this.pageno = pageno; }
38
39
40
41     /**
42      * pagesize
43      *
44      * @return the pagesize
45      * @since CodingExample Ver(编码范例查看) 1.0
46      */
47
48     public int getPagesize() { return pagesize; }
49
50
51
```

## Error > 泛型

```
BaseListBO.ts x
2
3 export interface IBaseListBO {
4     totalcount?: number;
5     datalist?: Array<any>;
6     pageno?: number;
7     datamap?: {[name: any]: any};
8     pagesize?: number;
9 }
10
11 export class BaseListBO {
12     constructor(params: IBaseListBO) {
13         this.totalcount = params.totalcount;
14         this.datalist = params.datalist;
15         this.pageno = params.pageno;
16         this.datamap = params.datamap;
17         this.pagesize = params.pagesize;
18     }
19
20     totalcount?: number;
21     datalist?: Array<any>;
22     pageno?: number;
23     datamap?: {[name: any]: any};
24     pagesize?: number;
25
26     __fields2java() {
27         return {
28             $class: 'com.qfpay.comm.base.BaseListBO',
29             $: {
30                 totalcount: java.Integer(this.totalcount),
31                 pageno: java.Integer(this.pageno),
32                 pagesize: java.Integer(this.pagesize),
33             },
34         };
35     }
36 }
37
38 //generate by interpret-cli dubbo2.js
39
```

Error > Enum类型

# Enum类型

## how > 翻译流程 > 转换 > Java to Ts > Enum转换实例

← → × 🏠 <https://www.npmjs.com/package/js-to-java> 🔍 ☆ 🏠

★ Bookmarks 🌐 🖨️ 📄 千米 📁 kindle 📁 frontend 📁 online 📁 OnlineTools 📁 QMTools 🔄 CNode: Ncde.js专...

```
java.BigDecimal('100.06');  
// Or java.BigDecimal({value: '100.06'});  
// => { $class: 'java.math.BigDecimal', $: { value: '100.06' } }
```

### java.enum(classname, value)

```
java.enum('hessian.demo.Color', 'RED');  
or  
java.enum('hessian.demo.Color', {name: 'RED'});  
// => { $class: 'hessian.demo.Color', $: { name: 'RED' } }
```

### java.array(classname, values)

```
java.array('Boolean', [true, false]);  
// => { $class: '[java.lang.Boolean]', $: [true, false] }
```

Available built-in classes shortcuts:

- java.array.Boolean(values)
- java.array.boolean(values)
- java.array.Integer(values)
- java.array.int(values)

## how > 翻译流程 > 转换 > Java to Ts > Enum转换实例

```
TagEnum.java x
1 package com.qianmi.pc.item.constant;
2
3 /**
4  * 商品标签枚举
5  * <p>Filename: com.qianmi.pc.item.constant.TagEnum.java</p>
6  * <p>Date: 2017-04-14 13:46.</p>
7  *
8  * @author <a href="mailto:haojingguo@qianmi.com">0F2602-郝敬国</a>
9  * @version v1.0.0
10  *
11  */
12 public enum TagEnum{
13     COMMEND(1),
14     NEW(2),
15     PROMOTION(3),
16     HOT(4);
17
18     private int value;
19
20     private TagEnum(int value) { this.value = value; }
21
22
23     public int getValue() { return this.value; }
24
25
26
27 }
28
```

```
TagEnum.ts x
1 export enum TagEnum {
2     NEW = 1,
3     PROMOTION = 2,
4     HOT = 3,
5     COMMEND = 4,
6 }
7 //generate by interpret-cli dubbo2.js
8
```



# how > 翻译流程 > 转换 > Java to Ts > Enum转换实例

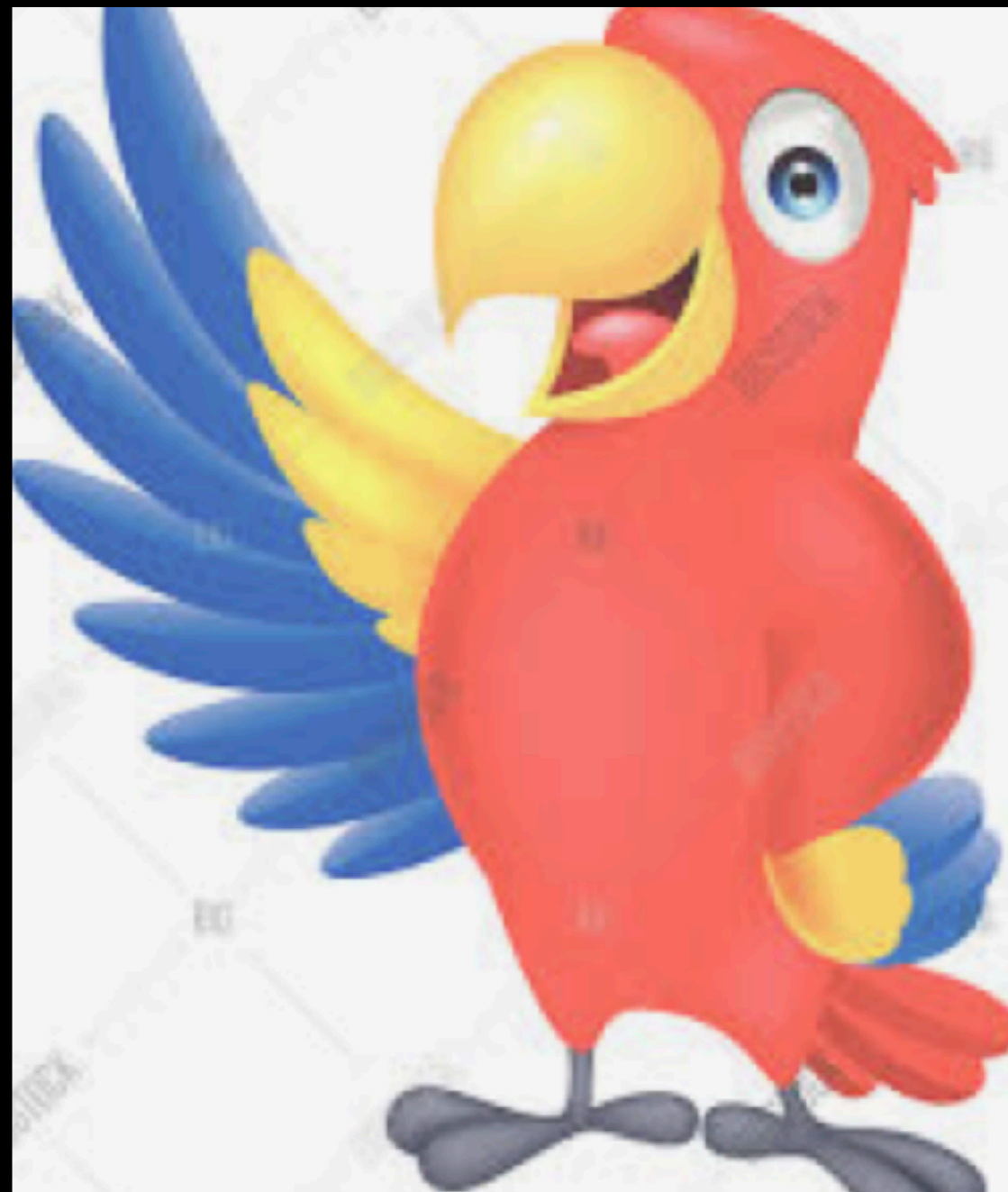
```
ActivityTypeEnum.java x ConsoleDataAnalysisProvider.java x ConsoleGiftCardQueryProvider.java x ConsoleGiftCardQueryProviderImpl.java x
PAY_REWARD 2 matches
15 */
16 @Getter
17 public enum ActivityTypeEnum {
18
19     // 可赠送的物品, 优先级最低
20     INTEGRAL("积分", "INTEGRAL", 999),
21     COUPON("优惠券", "COUPON", 999),
22     PRESENT("赠品", "PRESENT", 999),
23
24     // 单个商品
25     SEC_KILL("秒杀", "SEC_KILL", 1),
26     GROUPOUN("多人拼团", "GROUPOUN", 1),
27     MULTI_BUY_FEEDBACK("团购返现", "MULTI_BUY_FEEDBACK", 1),
28     TIME_LIMITED_DISCOUNT("限时抢购", "TIME_LIMITED_DISCOUNT", 10),
29     DAILY_DISCOUNT("天天特价", "DAILY_DISCOUNT", 10),
30
31     // 商品组合
32     COLLOCATION_GROUP("搭配宝", "COLLOCATION_GROUP", 20),
33     FIXED_GROUP("固定套装", "FIXED_GROUP", 20),
34     PACKAGE_ONE_PRICE("打包一口价", "PACKAGE_ONE_PRICE", 20),
35
36     // 订单级
37     FULL_PRIVILEGE("商品满减送", "FULL_PRIVILEGE", 30),
38     ORDER_CASH_BACK("订单返现", "ORDER_CASH_BACK", 40),
39
40     // xx有礼
41     REGISTER_REWARD("注册有礼", "REGISTER_REWARD", 1),
42     RECOMMAND_REWARD("推荐有礼", "RECOMMEND_REWARD", 1),
43     PAY_REWARD("支付有礼", "PAY_REWARD", 1),
44
45     // 幸运xx
46     LUCKY_LOTTERY("刮刮卡", "LUCKY_LOTTERY", 1),
47     WHEEL("幸运大转盘", "WHEEL", 1),
48     GOLDEN_EGGS("砸金蛋", "GOLDEN_EGGS", 1),
49
50     // 裂变券
51     FISSION_COUPON("裂变券", "FISSION_COUPON", 1),
52
53     // 签到
54     SIGN_IN("签到", "SIGN_IN", 1);
55
56 /**
57  * 商品绑定级别为SPU的活动类型列表
58  */
```

## how > 翻译流程 > 转换 > Java to Ts > Enum转换实例

```
BaseListBO.ts × Makefile × ActivityTypeEnum.ts ×
1 export enum ActivityTypeEnum {
2   ORDER_CASH_BACK = 'ORDER_CASH_BACK',
3   RECOMMAND_REWARD = 'RECOMMAND_REWARD',
4   PRESENT = 'PRESENT',
5   INTEGRAL = 'INTEGRAL',
6   WHEEL = 'WHEEL',
7   SIGN_IN = 'SIGN_IN',
8   COLLOCATION_GROUP = 'COLLOCATION_GROUP',
9   PAY_REWARD = 'PAY_REWARD',
10  SEC_KILL = 'SEC_KILL',
11  FIXED_GROUP = 'FIXED_GROUP',
12  PACKAGE_ONE_PRICE = 'PACKAGE_ONE_PRICE',
13  FISSION_COUPON = 'FISSION_COUPON',
14  COUPON = 'COUPON',
15  MULTI_BUY_FEEDBACK = 'MULTI_BUY_FEEDBACK',
16  GROUPON = 'GROUPON',
17  LUCKY_LOTTERY = 'LUCKY_LOTTERY',
18  REGISTER_REWARD = 'REGISTER_REWARD',
19  FULL_PRIVILEGE = 'FULL_PRIVILEGE',
20  TIME_LIMITED_DISCOUNT = 'TIME_LIMITED_DISCOUNT',
21  GOLDEN_EGGS = 'GOLDEN_EGGS',
22 }
23 //generate by interpret-cli dubbo2.js
24
```

# TODO

- 代码注释;
- Long BigDecimal失真!
- overRide处理;
- 参数是否需要不明确
- 多级泛型;
- 各中心代码自动化管理



**thanks all !**

**Q & A**