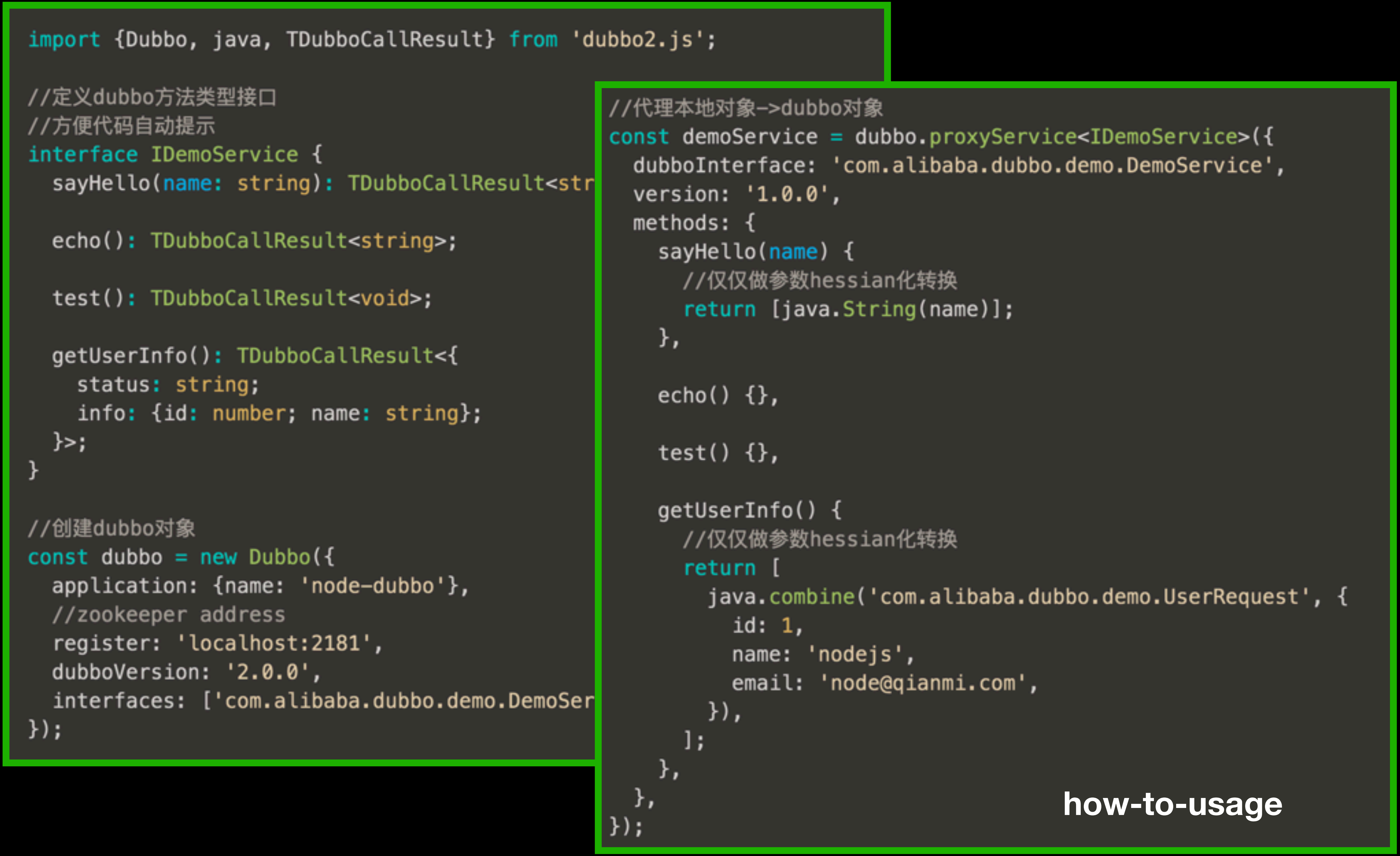
翻译师 => What?

what > dubbo2.js 使用示例



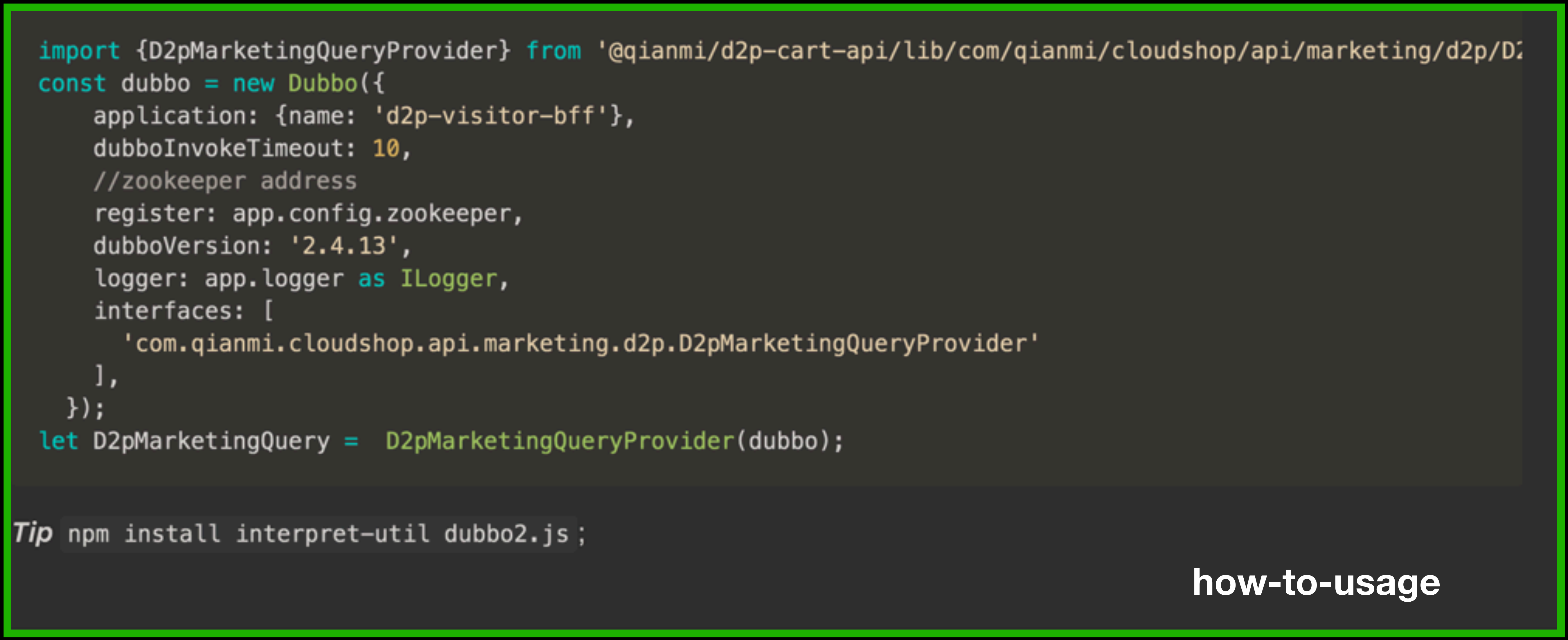
```
import {Dubbo, java, TDubboCallResult} from 'dubbo2.js';

//定义dubbo方法类型接口
//方便代码自动提示
interface IDemoService {
  sayHello(name: string): TDubboCallResult<str

  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.DemoSer
});
```

```
//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！

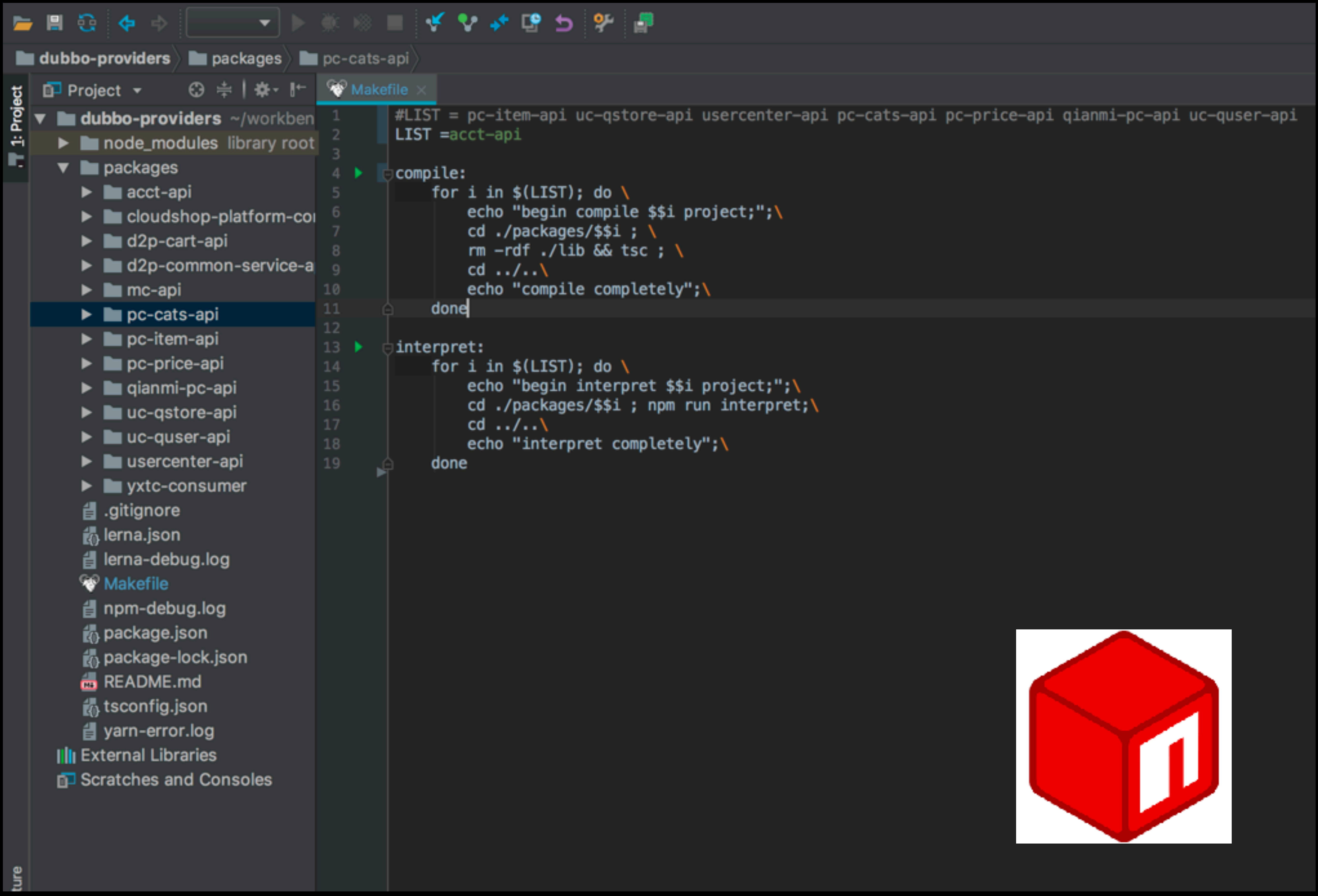


```
import {D2pMarketingQueryProvider} from '@qianmi/d2p-cart-api/lib/com/qianmi/cloudshop/api/marketing/d2p/D2
const dubbo = new Dubbo({
    application: {name: 'd2p-visitor-bff'},
    dubboInvokeTimeout: 10,
    //zookeeper address
    register: app.config.zookeeper,
    dubboVersion: '2.4.13',
    logger: app.logger as ILogger,
    interfaces: [
      'com.qianmi.cloudshop.api.marketing.d2p.D2pMarketingQueryProvider'
    ],
  });
let D2pMarketingQuery =  D2pMarketingQueryProvider(dubbo);
```

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

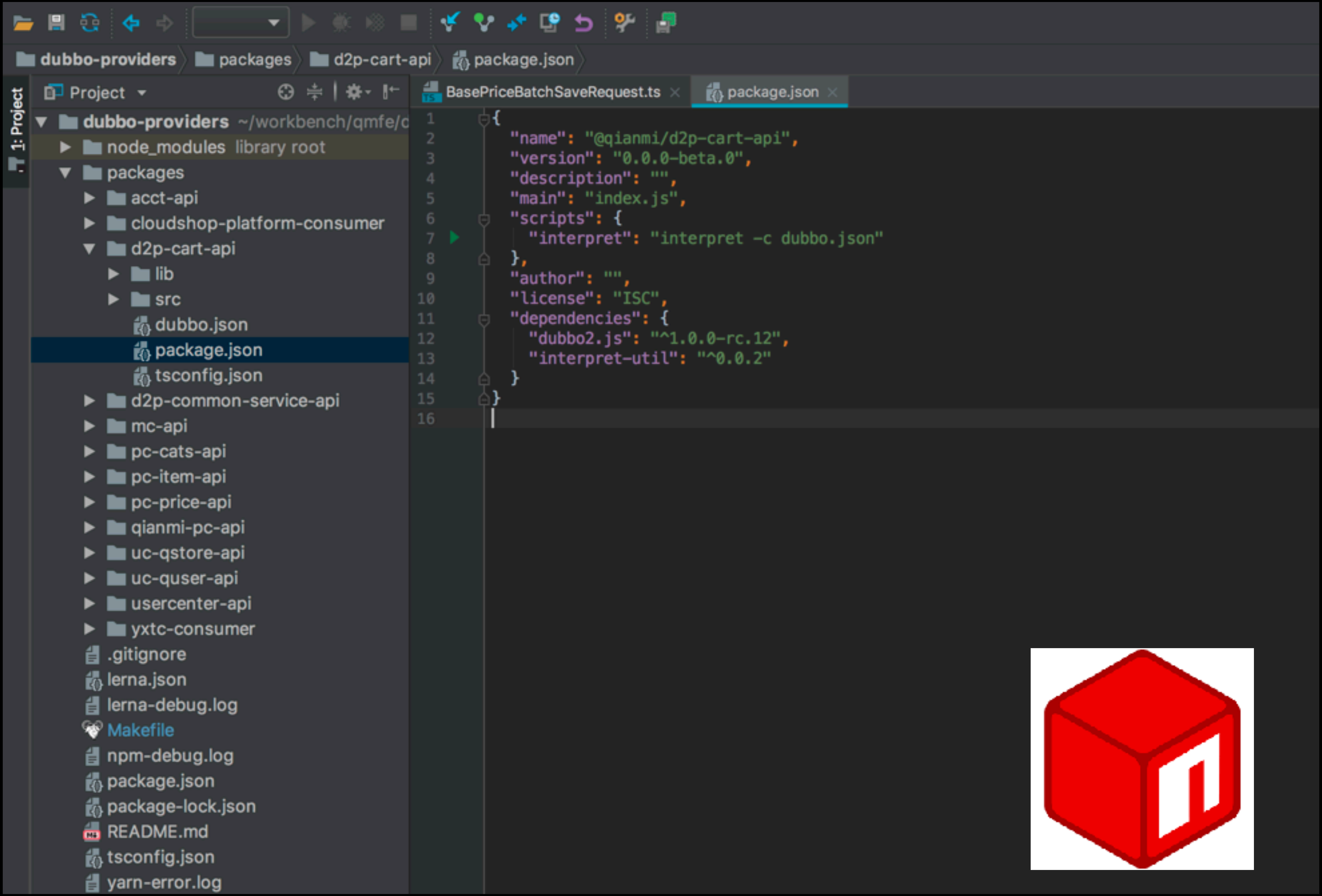
how-to-usage

what > dubbo2.js 使用示例



```
#LIST = pc-item-api uc-qstore-api usercenter-api pc-cats-api pc-price-api qianmi-pc-api uc-quser-api
LIST =acct-api

compile:
	for i in $(LIST); do \
		echo "begin compile $$i project;";\
		cd ./packages/$$i ; \
		rm -rdf ./lib && tsc ; \
		cd ../..\
		echo "compile completely";\
	done

interpret:
	for i in $(LIST); do \
		echo "begin interpret $$i project;";\
		cd ./packages/$$i ; npm run interpret;\
		cd ../..\
		echo "interpret completely";\
	done
```

what > dubbo2.js 使用示例

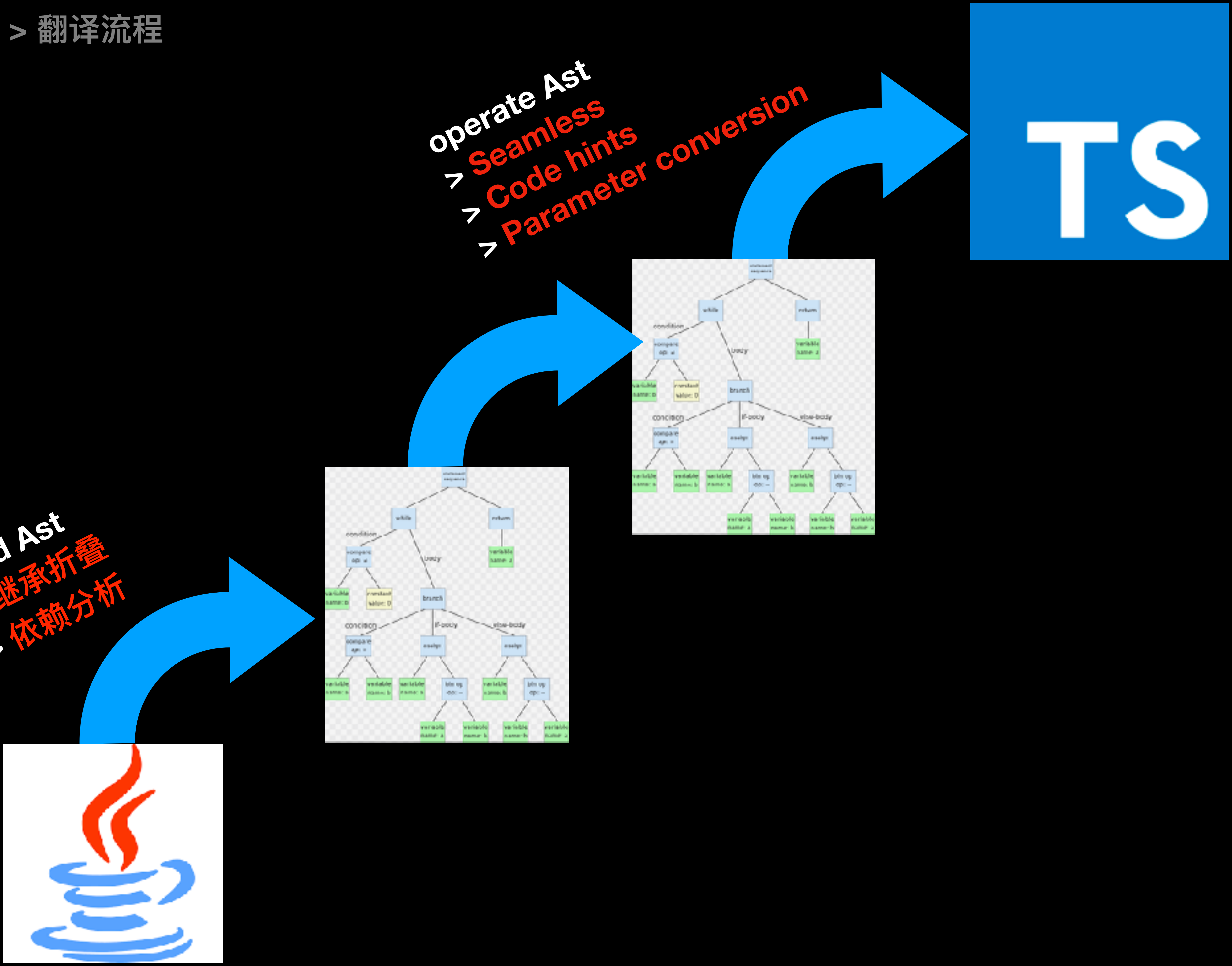


```
▼ dubbo-providers ~/workbench/qmfe/d
  ▶ node_modules library root
  ▼ packages
    ▶ acct-api
    ▶ cloudshop-platform-consumer
    ▼ d2p-cart-api
      ▶ lib
      ▶ src
        dubbo.json
        package.json
        tsconfig.json
    ▶ d2p-common-service-api
    ▶ mc-api
    ▶ pc-cats-api
    ▶ 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
```

```json
{
  "name": "@qianmi/d2p-cart-api",
  "version": "0.0.0-beta.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "interpret": "interpret -c dubbo.json"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "dubbo2.js": "^1.0.0-rc.12",
    "interpret-util": "^0.0.2"
  }
}

```

翻译师 => How ?

how > 翻译流程

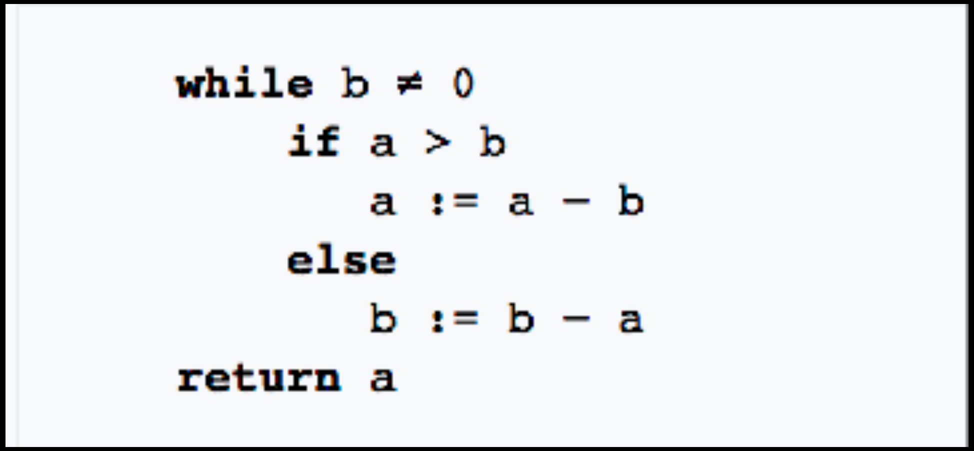


read Ast
> 继承折叠
> 依赖分析

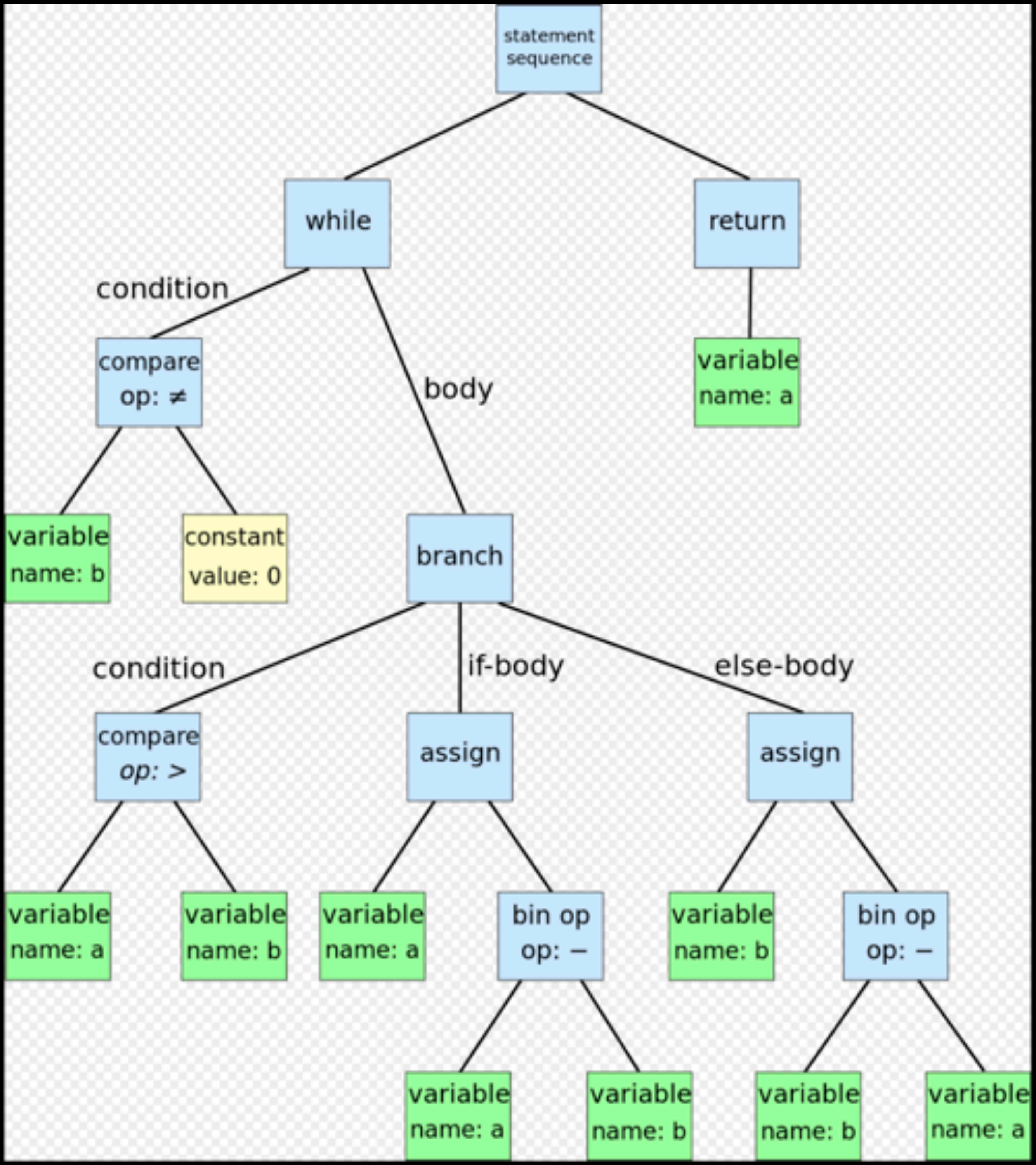
operate Ast
> Seamless
> Code hints
> Parameter conversion

Extract Ast

how > ast抽象语法树

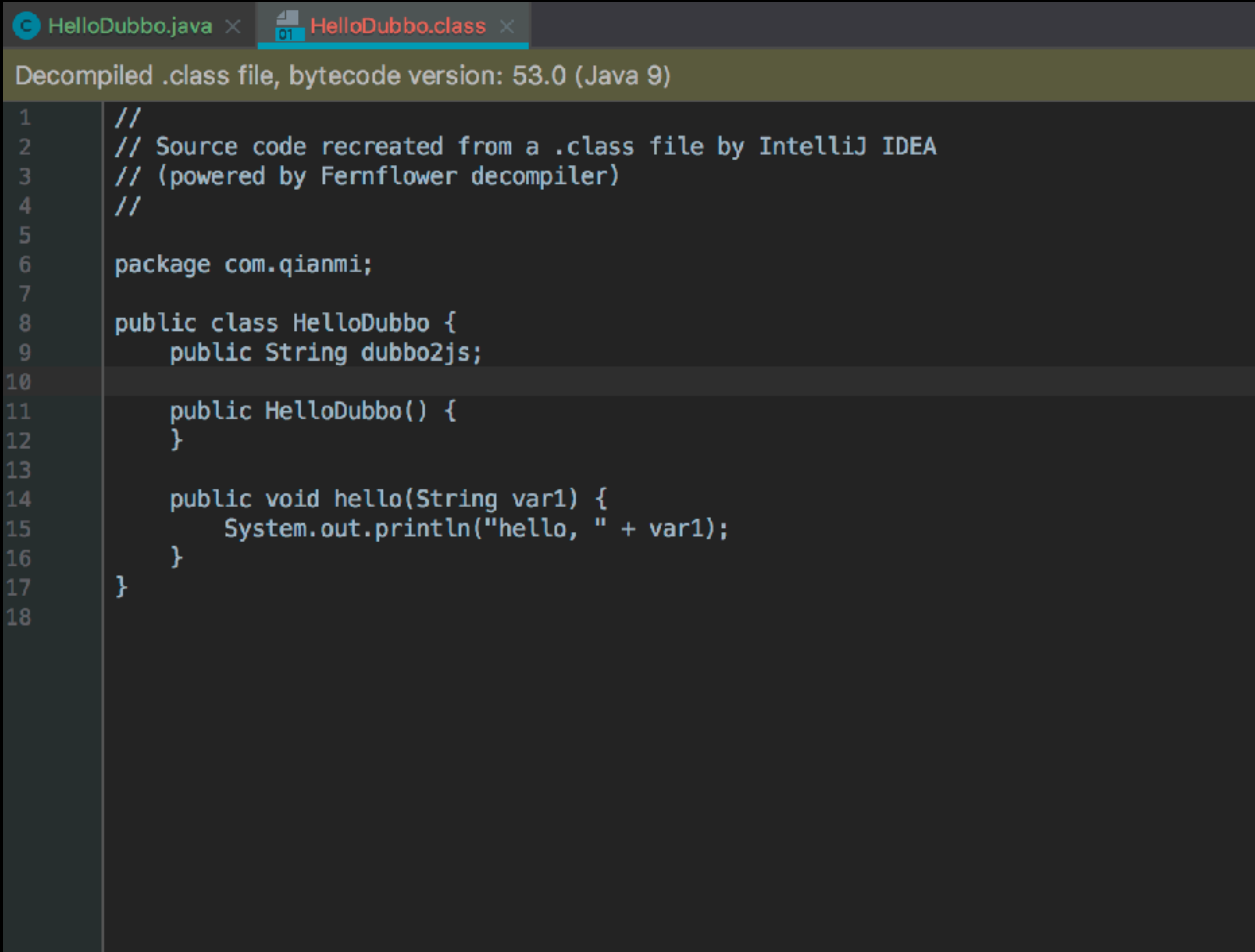


```
while b ≠ 0
    if a > b
        a := a − b
    else
        b := b − a
return a
```



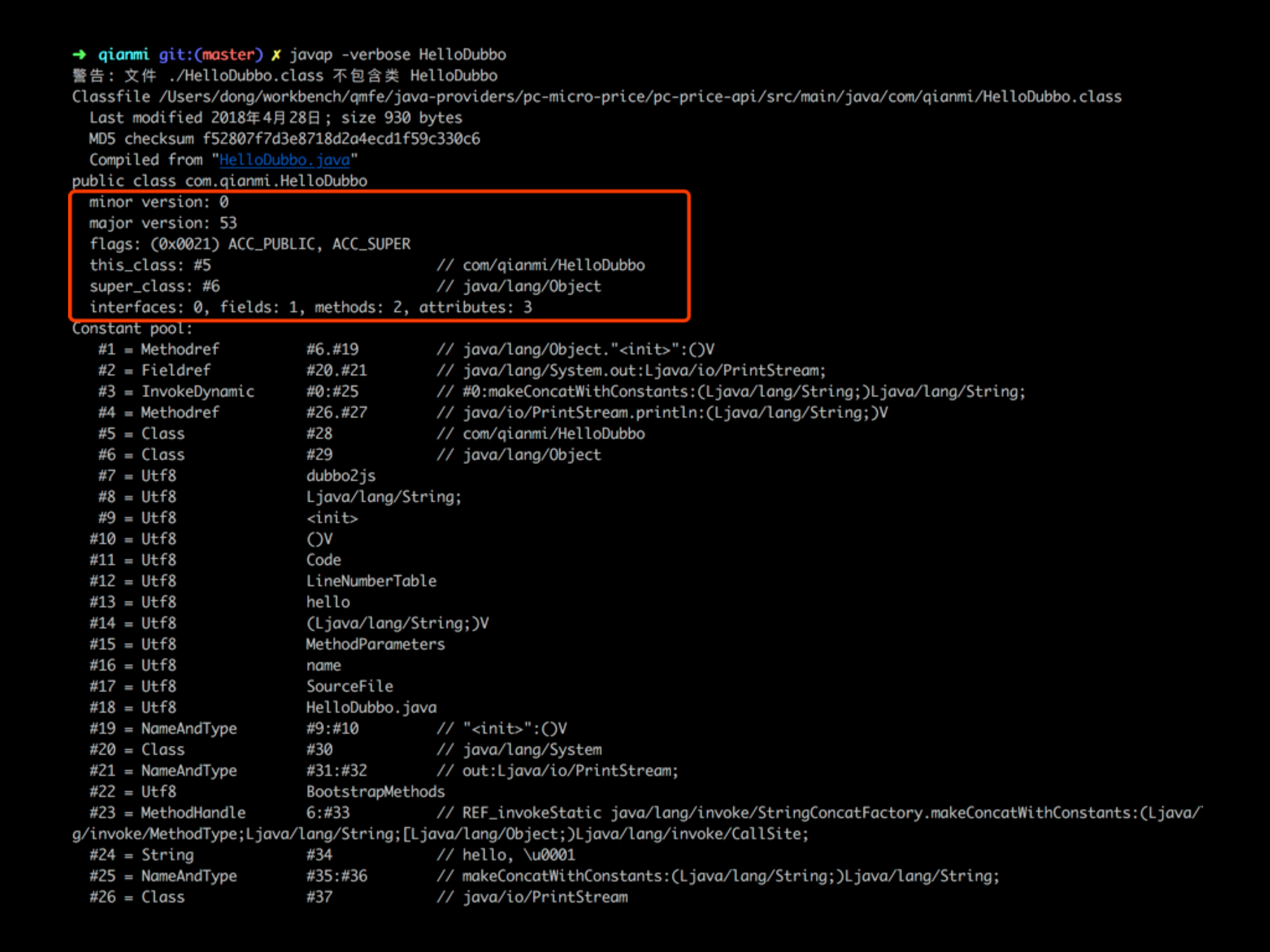
```java
package com.qianmi;

public class HelloDubbo {

    public String dubbo2js;

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

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

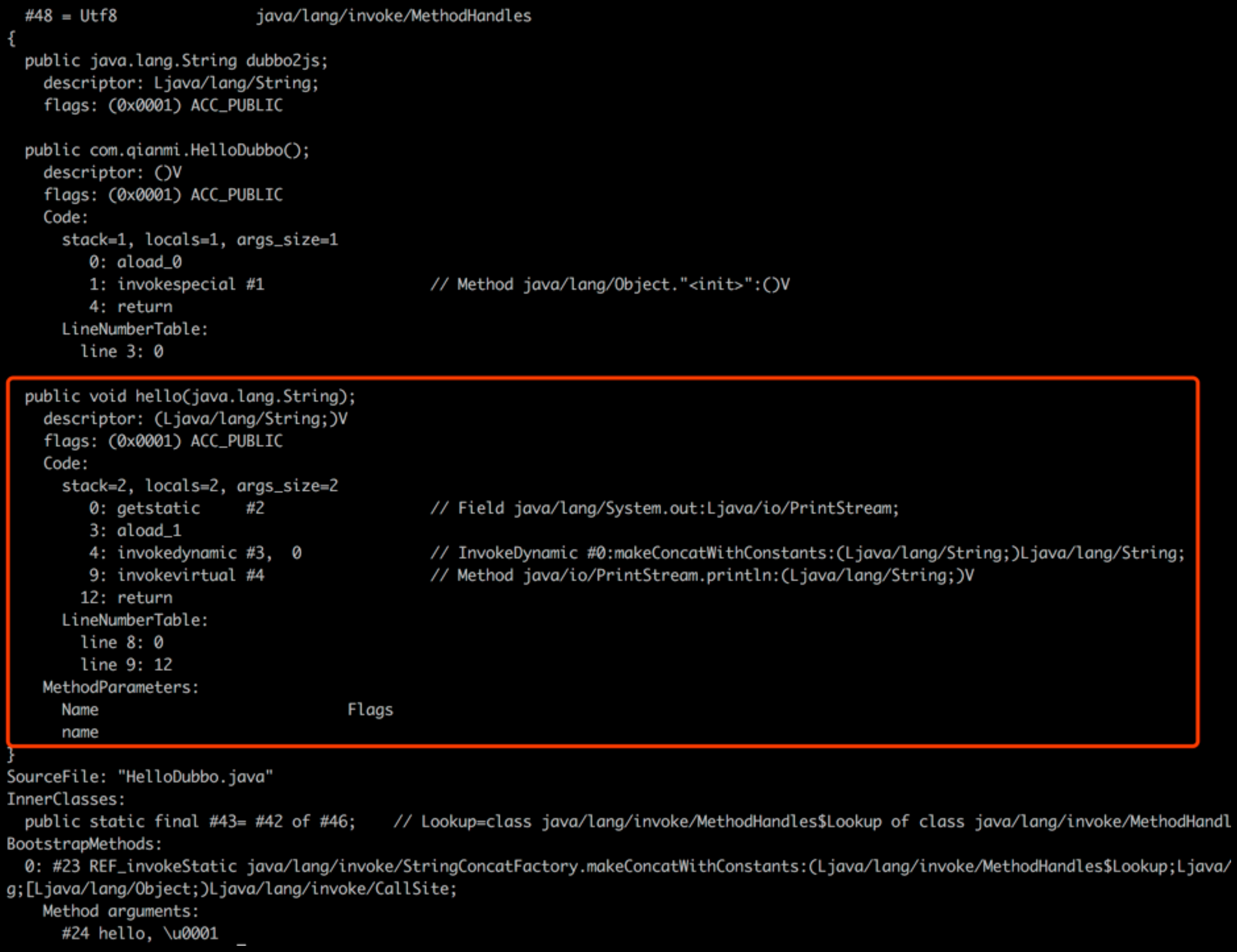


Decompiled .class file, bytecode version: 53.0 (Java 9)

```
//
// Source code recreated from a .class file by IntelliJ IDEA
// (powered by Fernflower decompiler)
//

package com.qianmi;

public class HelloDubbo {
    public String dubbo2js;

    public HelloDubbo() {
    }

    public void hello(String var1) {
        System.out.println("hello, " + var1);
    }
}
```



```
➜ 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~~

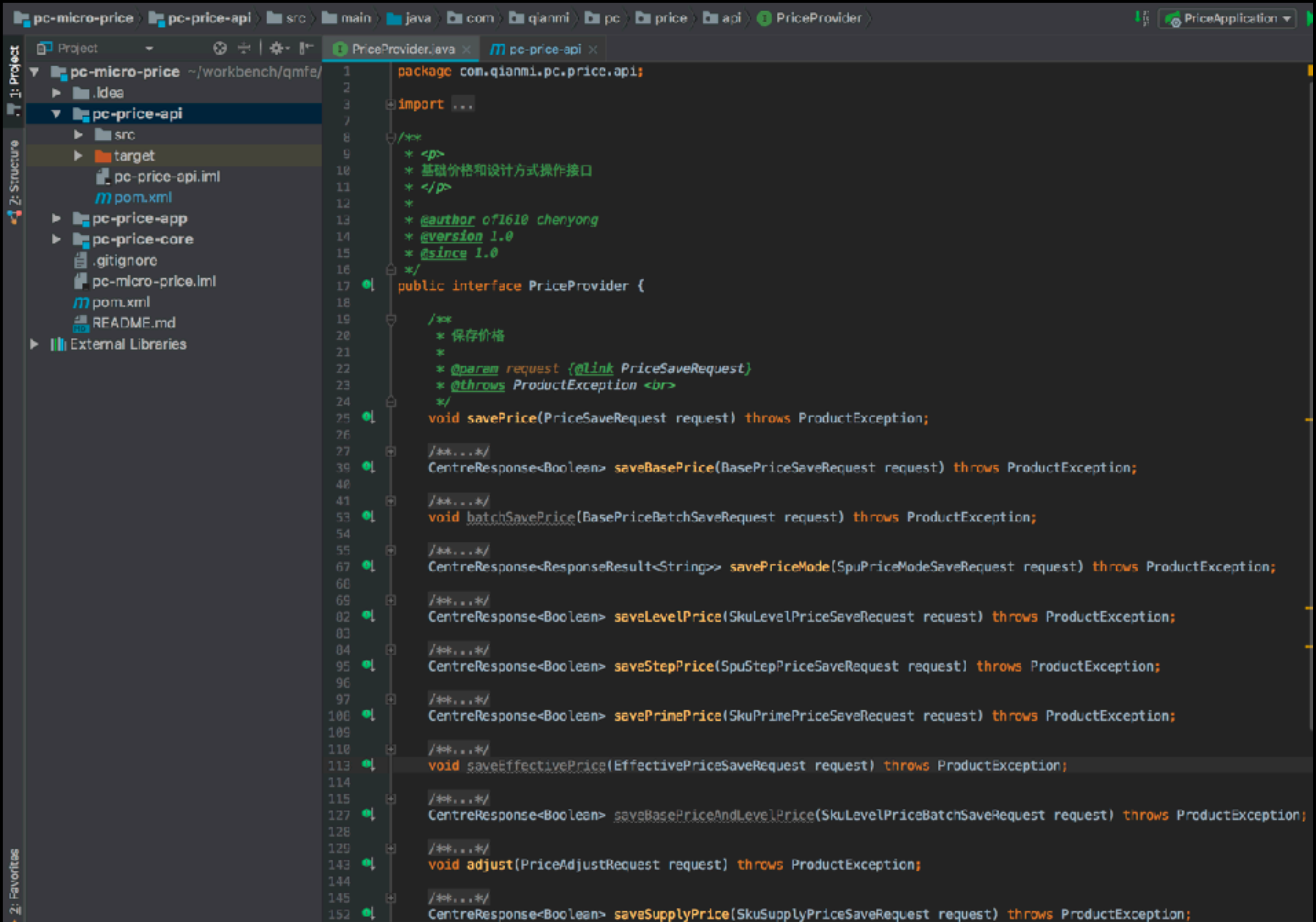
java to Ts

- 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

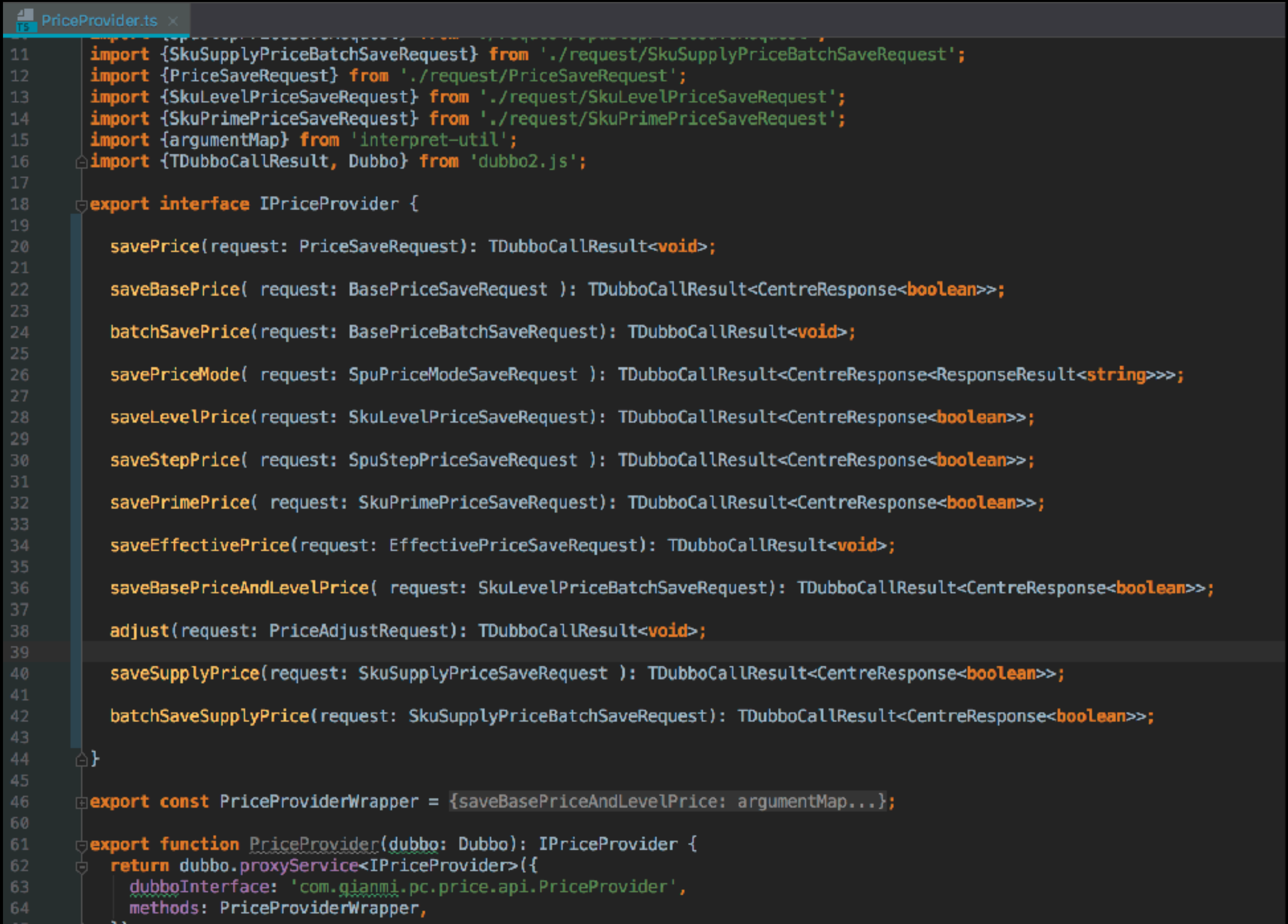
Java类型	Typescript类型
java.lang.String	string
java.lang.Object	Object
java.lang.Integer/java.lang.int	number
java.lang.short /java.lang.Short	number
java.lang.long/java.lang.Long	number
java.lang.double/java.lang.Double	number
java.lang.float/java.lang.Float	number
java.lang.Void	void
java.lang.boolean/java.lang.Boolean	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



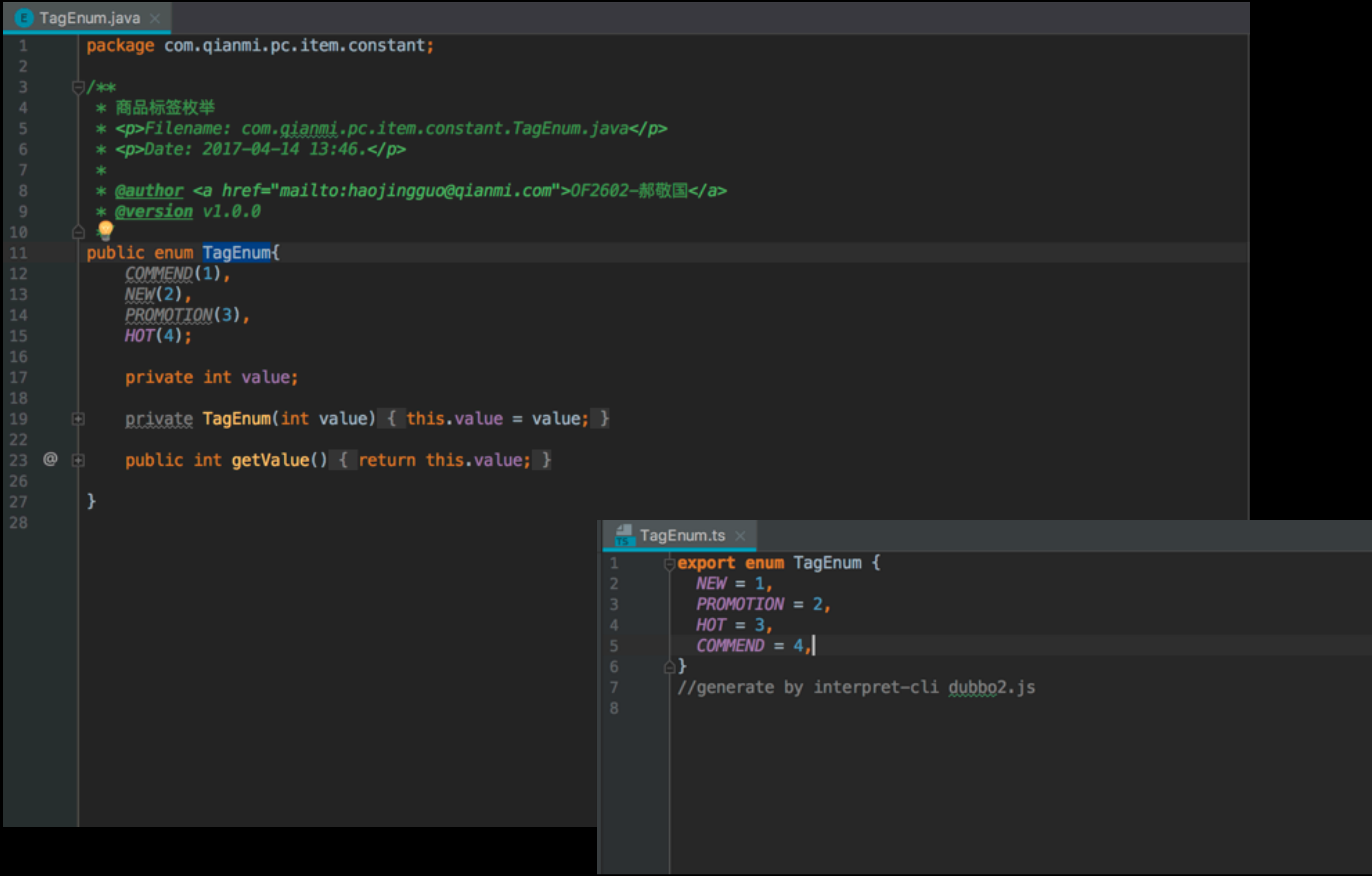
```
package com.qianmi.pc.price.api;

import ...

/**
 * <p>
 * 基础价格和设计方式操作接口
 * </p>
 *
 * @author of1610 chenyong
 * @version 1.0
 * @since 1.0
 */
public interface PriceProvider {

    /**
     * 保存价格
     *
     * @param request {@link PriceSaveRequest}
     * @throws ProductException <br>
     */
    void savePrice(PriceSaveRequest request) throws ProductException;

    /**...*/
    CentreResponse<Boolean> saveBasePrice(BasePriceSaveRequest request) throws ProductException;

    /**...*/
    void batchSavePrice(BasePriceBatchSaveRequest request) throws ProductException;

    /**...*/
    CentreResponse<ResponseResult<String>> savePriceMode(SpuPriceModeSaveRequest request) throws ProductException;

    /**...*/
    CentreResponse<Boolean> saveLevelPrice(SkuLevelPriceSaveRequest request) throws ProductException;

    /**...*/
    CentreResponse<Boolean> saveStepPrice(SpuStepPriceSaveRequest request) throws ProductException;

    /**...*/
    CentreResponse<Boolean> savePrimePrice(SkuPrimePriceSaveRequest request) throws ProductException;

    /**...*/
    void saveEffectivePrice(EffectivePriceSaveRequest request) throws ProductException;

    /**...*/
    CentreResponse<Boolean> saveBasePriceAndLevelPrice(SkuLevelPriceBatchSaveRequest request) throws ProductException;

    /**...*/
    void adjust(PriceAdjustRequest request) throws ProductException;

    /**...*/
    CentreResponse<Boolean> saveSupplyPrice(SkuSupplyPriceSaveRequest request) throws ProductException;
```

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



```typescript
import {SkuSupplyPriceBatchSaveRequest} from './request/SkuSupplyPriceBatchSaveRequest';
import {PriceSaveRequest} from './request/PriceSaveRequest';
import {SkuLevelPriceSaveRequest} from './request/SkuLevelPriceSaveRequest';
import {SkuPrimePriceSaveRequest} from './request/SkuPrimePriceSaveRequest';
import {argumentMap} from 'interpret-util';
import {TDubboCallResult, Dubbo} from 'dubbo2.js';

export interface IPriceProvider {

  savePrice(request: PriceSaveRequest): TDubboCallResult<void>;

  saveBasePrice( request: BasePriceSaveRequest ): TDubboCallResult<CentreResponse<boolean>>;

  batchSavePrice(request: BasePriceBatchSaveRequest): TDubboCallResult<void>;

  savePriceMode( request: SpuPriceModeSaveRequest ): TDubboCallResult<CentreResponse<ResponseResult<string>>>;

  saveLevelPrice(request: SkuLevelPriceSaveRequest): TDubboCallResult<CentreResponse<boolean>>;

  saveStepPrice( request: SpuStepPriceSaveRequest ): TDubboCallResult<CentreResponse<boolean>>;

  savePrimePrice( request: SkuPrimePriceSaveRequest): TDubboCallResult<CentreResponse<boolean>>;

  saveEffectivePrice(request: EffectivePriceSaveRequest): TDubboCallResult<void>;

  saveBasePriceAndLevelPrice( request: SkuLevelPriceBatchSaveRequest): TDubboCallResult<CentreResponse<boolean>>;

  adjust(request: PriceAdjustRequest): TDubboCallResult<void>;

  saveSupplyPrice(request: SkuSupplyPriceSaveRequest ): TDubboCallResult<CentreResponse<boolean>>;

  batchSaveSupplyPrice(request: SkuSupplyPriceBatchSaveRequest): TDubboCallResult<CentreResponse<boolean>>;

}

export const PriceProviderWrapper = {saveBasePriceAndLevelPrice: argumentMap...};

export function PriceProvider(dubbo: Dubbo): IPriceProvider {
  return dubbo.proxyService<IPriceProvider>({
    dubboInterface: 'com.qianmi.pc.price.api.PriceProvider',
    methods: PriceProviderWrapper,
```

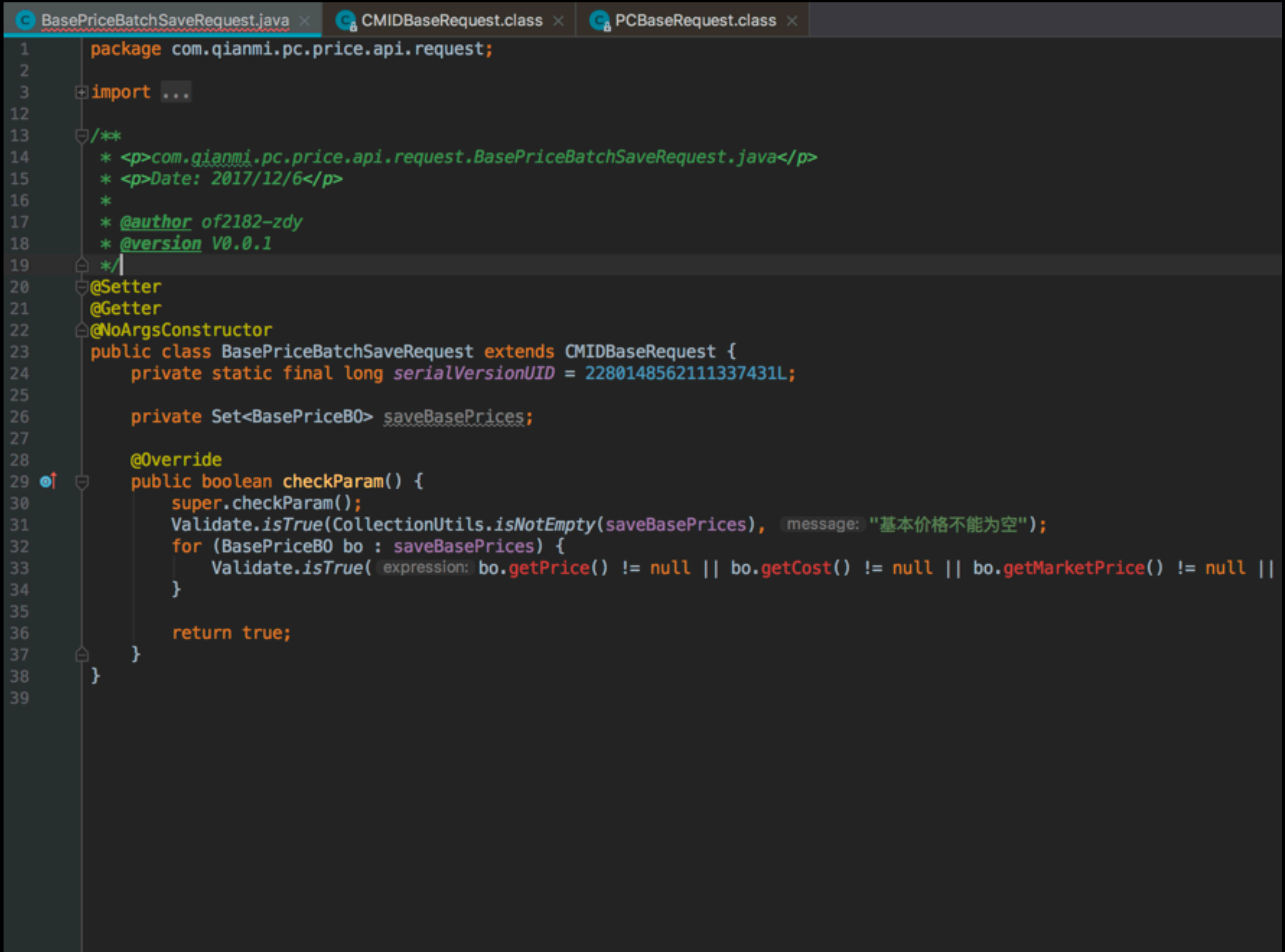
how > 翻译流程 > 转换 > Java to Ts > Enum转换示例



```java
package com.qianmi.pc.item.constant;

/**
 * 商品标签枚举
 * <p>Filename: com.qianmi.pc.item.constant.TagEnum.java</p>
 * <p>Date: 2017-04-14 13:46.</p>
 *
 * @author <a href="mailto:haojingguo@qianmi.com">OF2602-郝敬国</a>
 * @version v1.0.0
 */
public enum TagEnum{
    COMMEND(1),
    NEW(2),
    PROMOTION(3),
    HOT(4);

    private int value;

    private TagEnum(int value) { this.value = value; }

    public int getValue() { return this.value; }

}
```

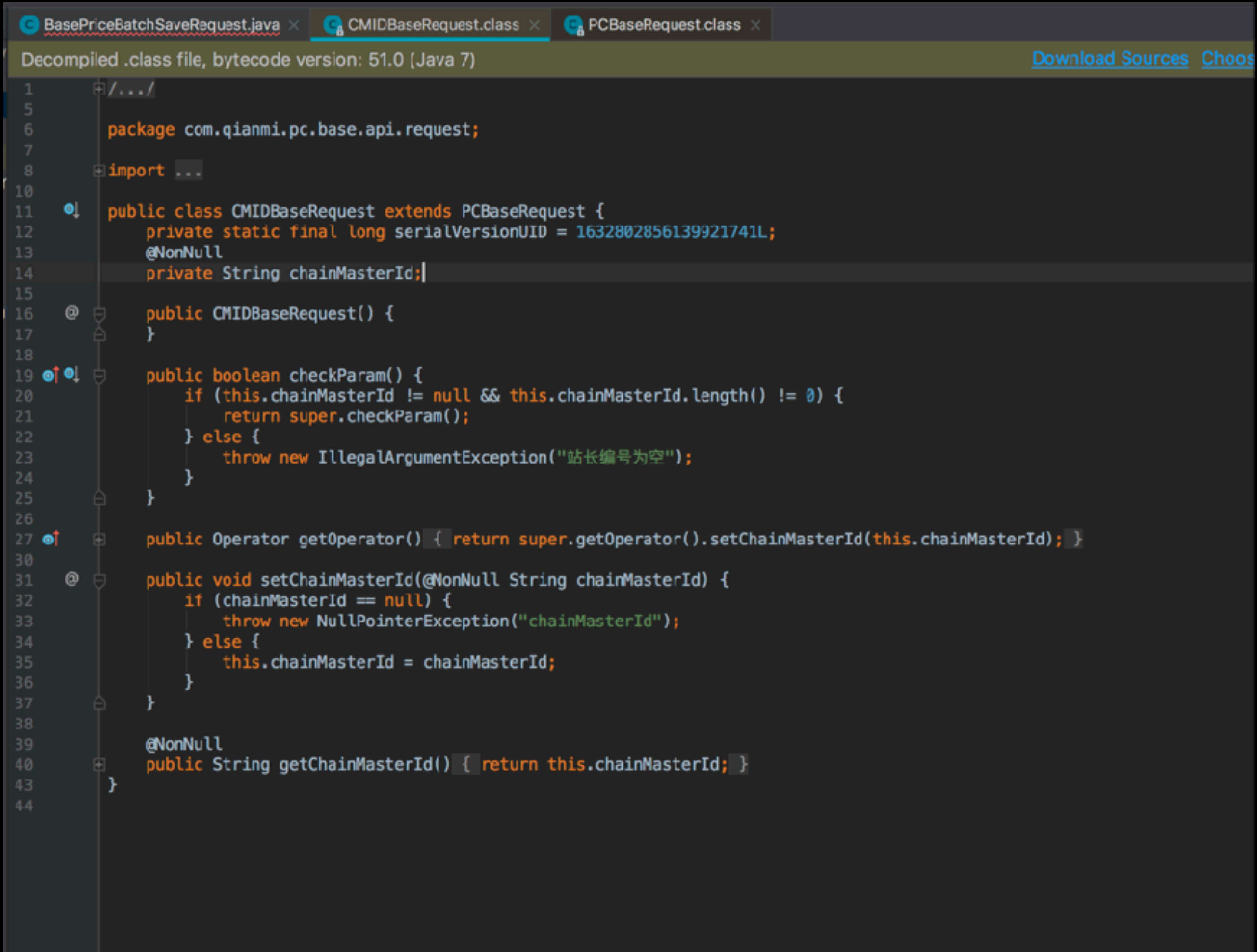
```ts
export enum TagEnum {
  NEW = 1,
  PROMOTION = 2,
  HOT = 3,
  COMMEND = 4,
}
//generate by interpret-cli dubbo2.js
```

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



```
package com.qianmi.pc.price.api.request;

import ...

/**
 * <p>com.qianmi.pc.price.api.request.BasePriceBatchSaveRequest.java</p>
 * <p>Date: 2017/12/6</p>
 *
 * @author of2182-zdy
 * @version V0.0.1
 */
@Setter
@Getter
@NoArgsConstructor
public class BasePriceBatchSaveRequest extends CMIDBaseRequest {
    private static final long serialVersionUID = 2280148562111337431L;

    private Set<BasePriceBO> saveBasePrices;

    @Override
    public boolean checkParam() {
        super.checkParam();
        Validate.isTrue(CollectionUtils.isNotEmpty(saveBasePrices),  message: "基本价格不能为空");
        for (BasePriceBO bo : saveBasePrices) {
            Validate.isTrue( expression: bo.getPrice() != null || bo.getCost() != null || bo.getMarketPrice() != null ||
        }

        return true;
    }
}
```

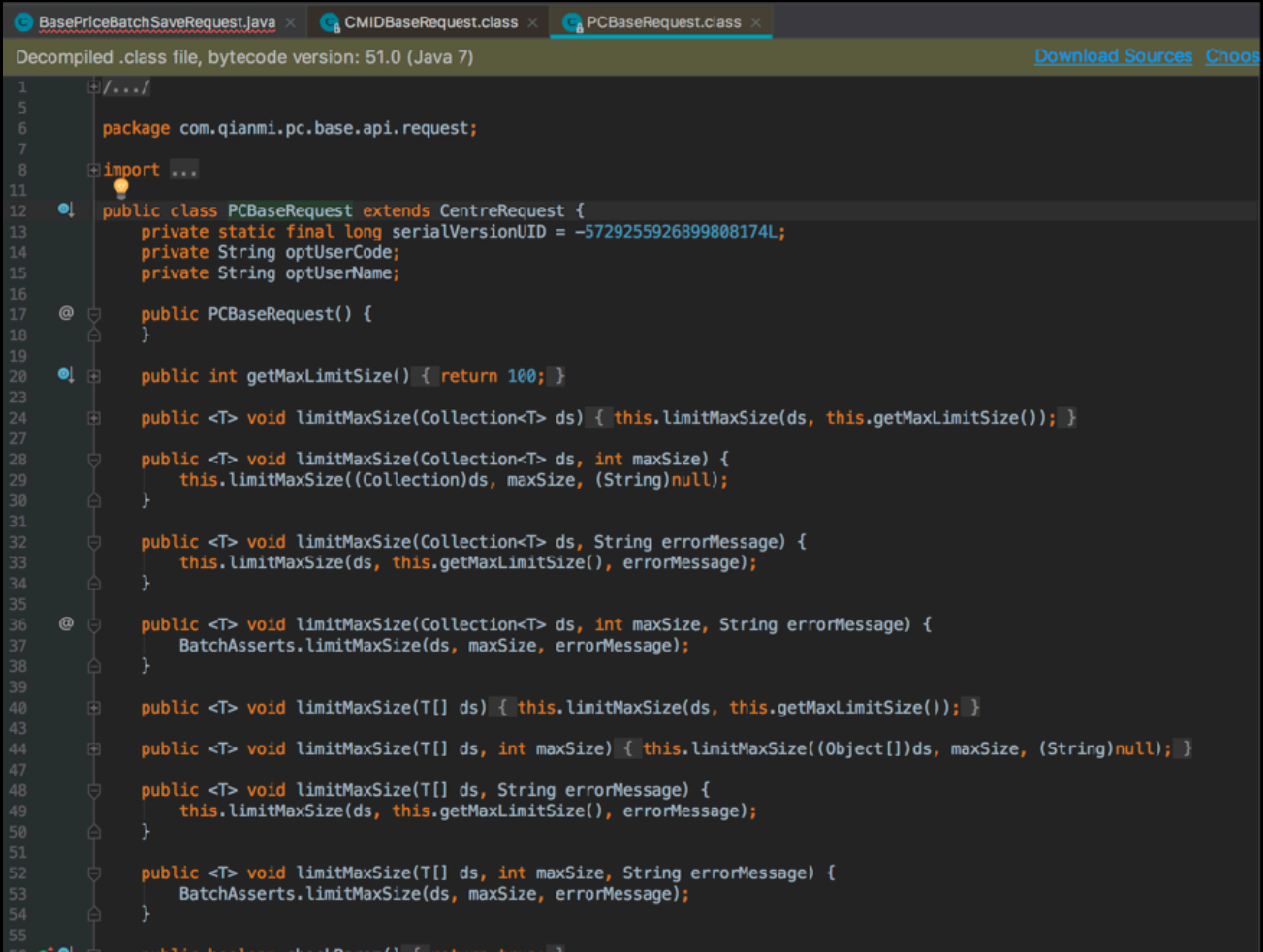
how > 翻译流程 > 转换 > Java to Ts > Bean转换示例



Decompiled .class file, bytecode version: 51.0 (Java 7)

```java
/.../

package com.qianmi.pc.base.api.request;

import ...

public class CMIDBaseRequest extends PCBaseRequest {
    private static final long serialVersionUID = 1632802856139921741L;
    @NonNull
    private String chainMasterId;

    public CMIDBaseRequest() {
    }

    public boolean checkParam() {
        if (this.chainMasterId != null && this.chainMasterId.length() != 0) {
            return super.checkParam();
        } else {
            throw new IllegalArgumentException("站长编号为空");
        }
    }

    public Operator getOperator() { return super.getOperator().setChainMasterId(this.chainMasterId); }

    public void setChainMasterId(@NonNull String chainMasterId) {
        if (chainMasterId == null) {
            throw new NullPointerException("chainMasterId");
        } else {
            this.chainMasterId = chainMasterId;
        }
    }

    @NonNull
    public String getChainMasterId() { return this.chainMasterId; }
}
```

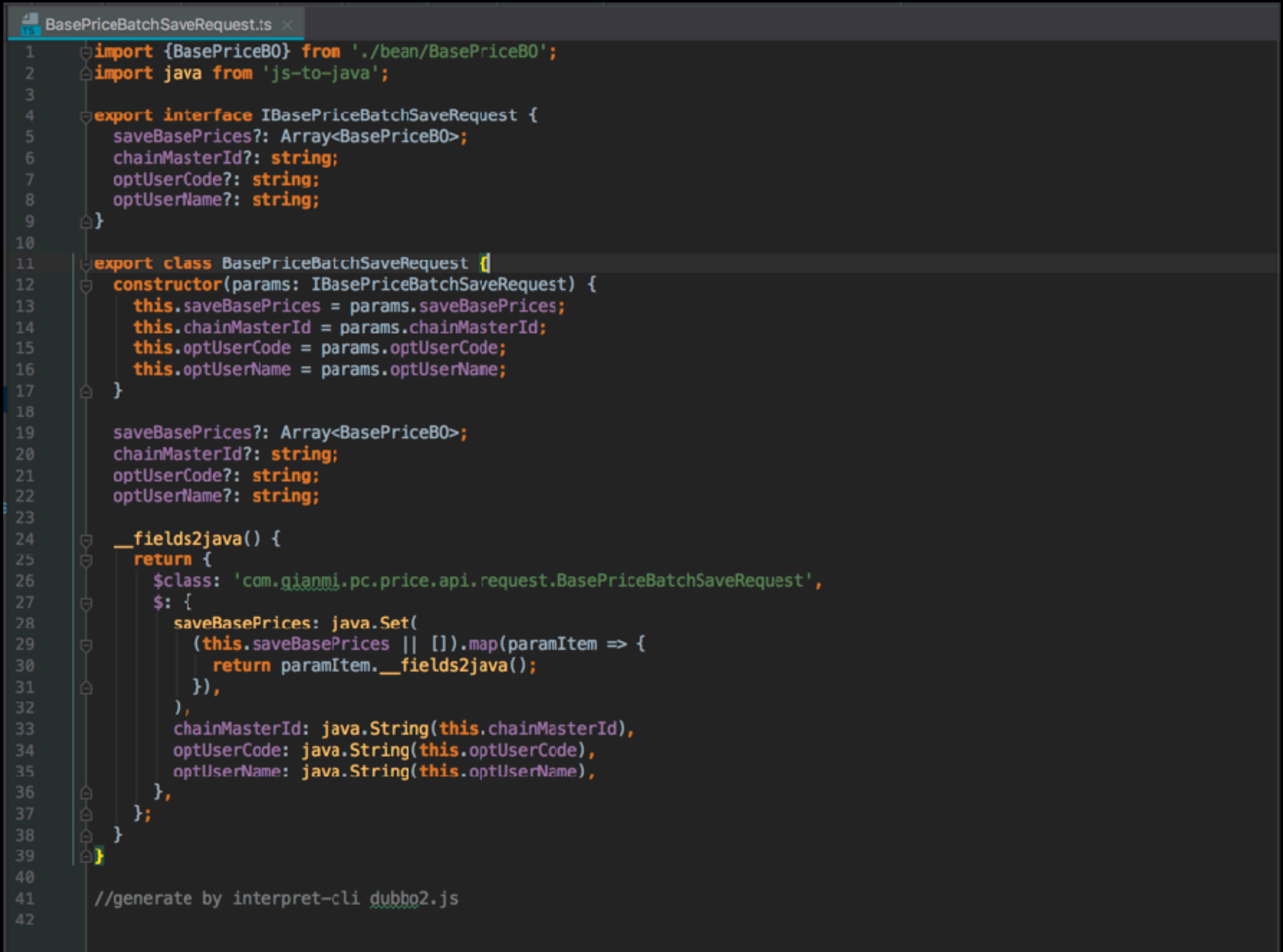
how > 翻译流程 > 转换 > Java to Ts > Bean转换示例



Decompiled .class file, bytecode version: 51.0 (Java 7)

```java
/.../

package com.qianmi.pc.base.api.request;

import ...

public class PCBaseRequest extends CentreRequest {
    private static final long serialVersionUID = -5729255926899808174L;
    private String optUserCode;
    private String optUserName;

    public PCBaseRequest() {
    }

    public int getMaxLimitSize() { return 100; }

    public <T> void limitMaxSize(Collection<T> ds) { this.limitMaxSize(ds, this.getMaxLimitSize()); }

    public <T> void limitMaxSize(Collection<T> ds, int maxSize) {
        this.limitMaxSize((Collection)ds, maxSize, (String)null);
    }

    public <T> void limitMaxSize(Collection<T> ds, String errorMessage) {
        this.limitMaxSize(ds, this.getMaxLimitSize(), errorMessage);
    }

    public <T> void limitMaxSize(Collection<T> ds, int maxSize, String errorMessage) {
        BatchAsserts.limitMaxSize(ds, maxSize, errorMessage);
    }

    public <T> void limitMaxSize(T[] ds) { this.limitMaxSize(ds, this.getMaxLimitSize()); }

    public <T> void limitMaxSize(T[] ds, int maxSize) { this.limitMaxSize((Object[])ds, maxSize, (String)null); }

    public <T> void limitMaxSize(T[] ds, String errorMessage) {
        this.limitMaxSize(ds, this.getMaxLimitSize(), errorMessage);
    }

    public <T> void limitMaxSize(T[] ds, int maxSize, String errorMessage) {
        BatchAsserts.limitMaxSize(ds, maxSize, errorMessage);
    }

```

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

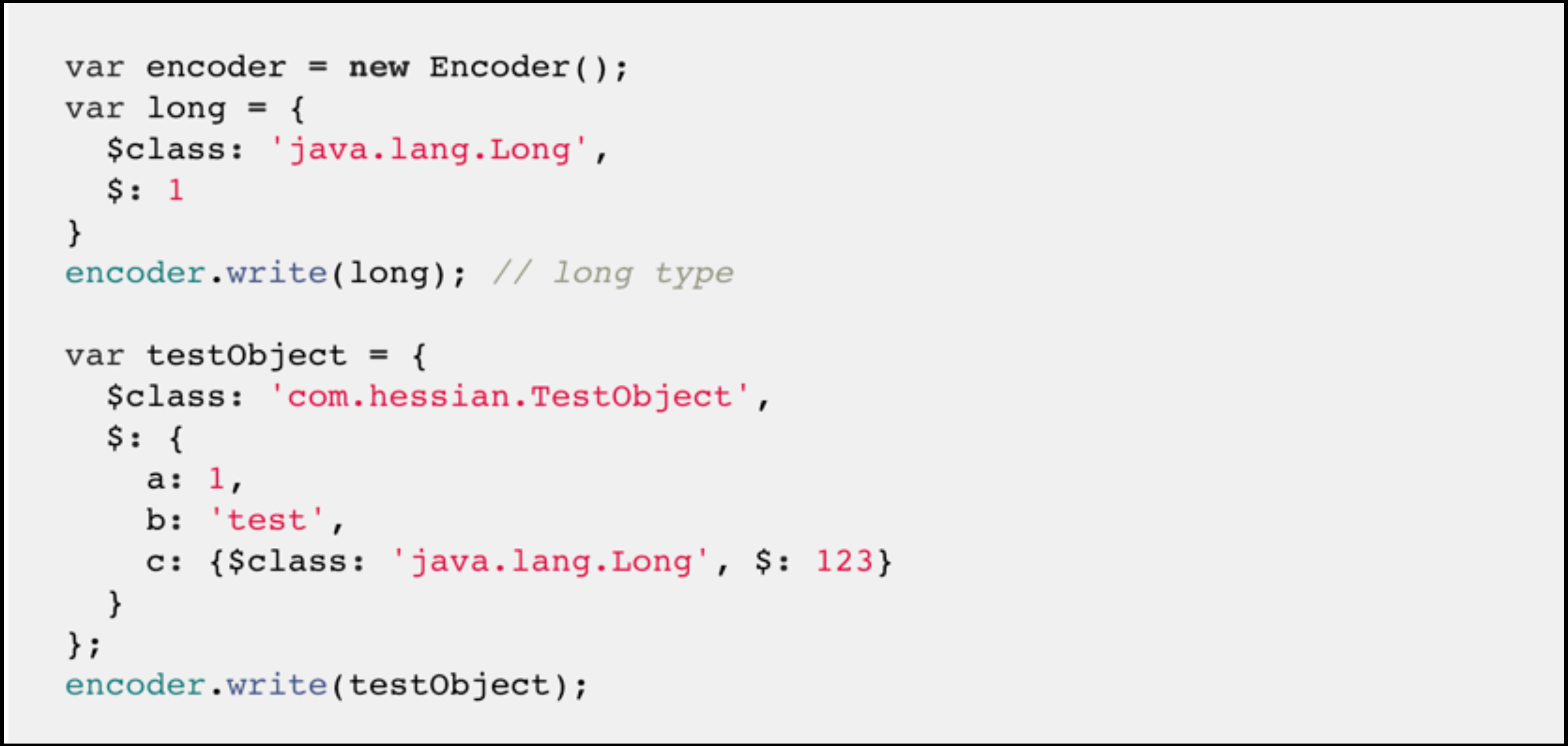


```
import {BasePriceBO} from './bean/BasePriceBO';
import java from 'js-to-java';

export interface IBasePriceBatchSaveRequest {
  saveBasePrices?: Array<BasePriceBO>;
  chainMasterId?: string;
  optUserCode?: string;
  optUserName?: string;
}

export class BasePriceBatchSaveRequest {
  constructor(params: IBasePriceBatchSaveRequest) {
    this.saveBasePrices = params.saveBasePrices;
    this.chainMasterId = params.chainMasterId;
    this.optUserCode = params.optUserCode;
    this.optUserName = params.optUserName;
  }

  saveBasePrices?: Array<BasePriceBO>;
  chainMasterId?: string;
  optUserCode?: string;
  optUserName?: string;

  __fields2java() {
    return {
      $class: 'com.qianmi.pc.price.api.request.BasePriceBatchSaveRequest',
      $: {
        saveBasePrices: java.Set(
          (this.saveBasePrices || []).map(paramItem => {
            return paramItem.__fields2java();
          }),
        ),
        chainMasterId: java.String(this.chainMasterId),
        optUserCode: java.String(this.optUserCode),
        optUserName: java.String(this.optUserName),
      },
    };
  }
}

//generate by interpret-cli dubbo2.js
```

参数-转换

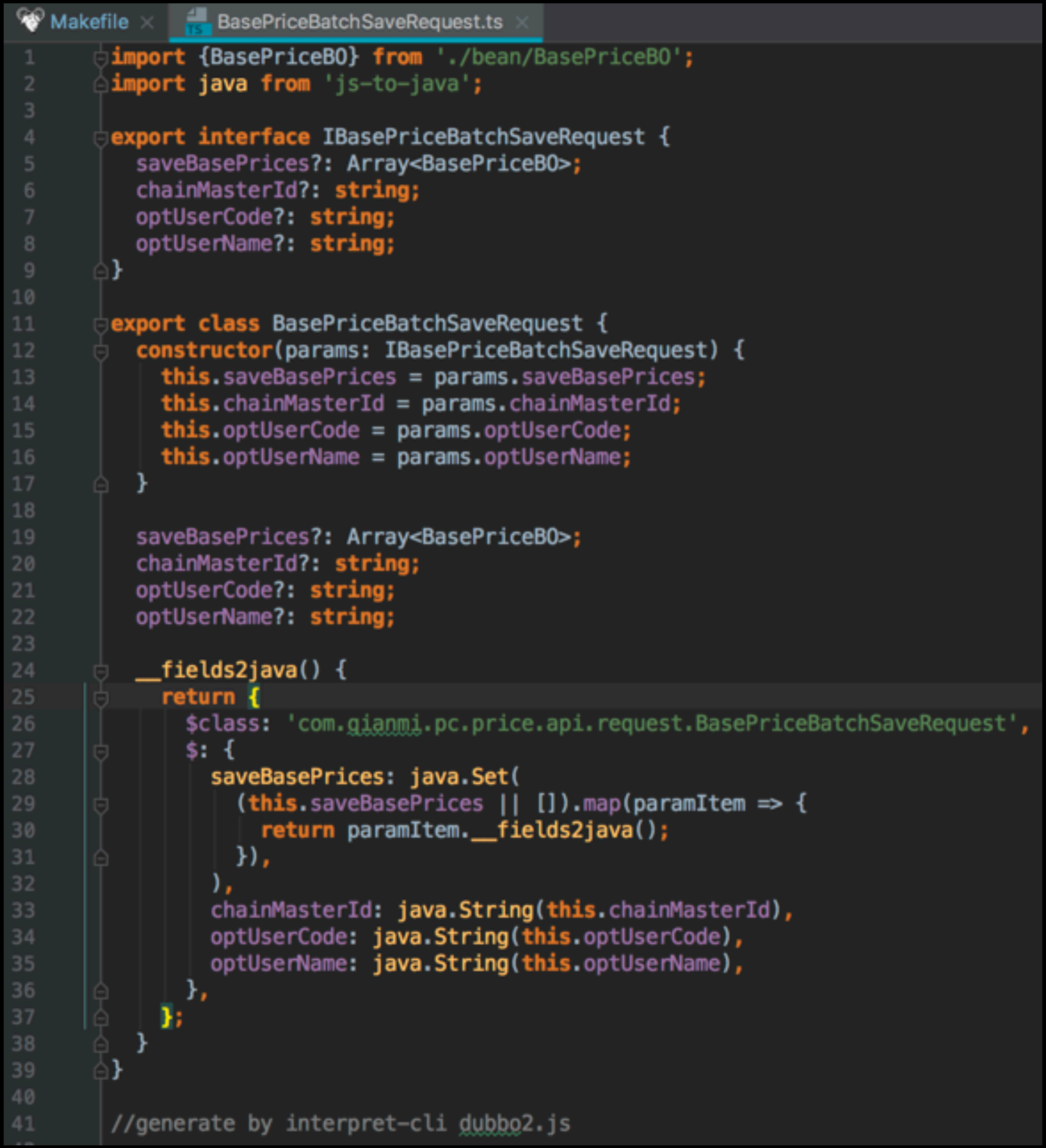
- bean对象转换为hessian Param

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



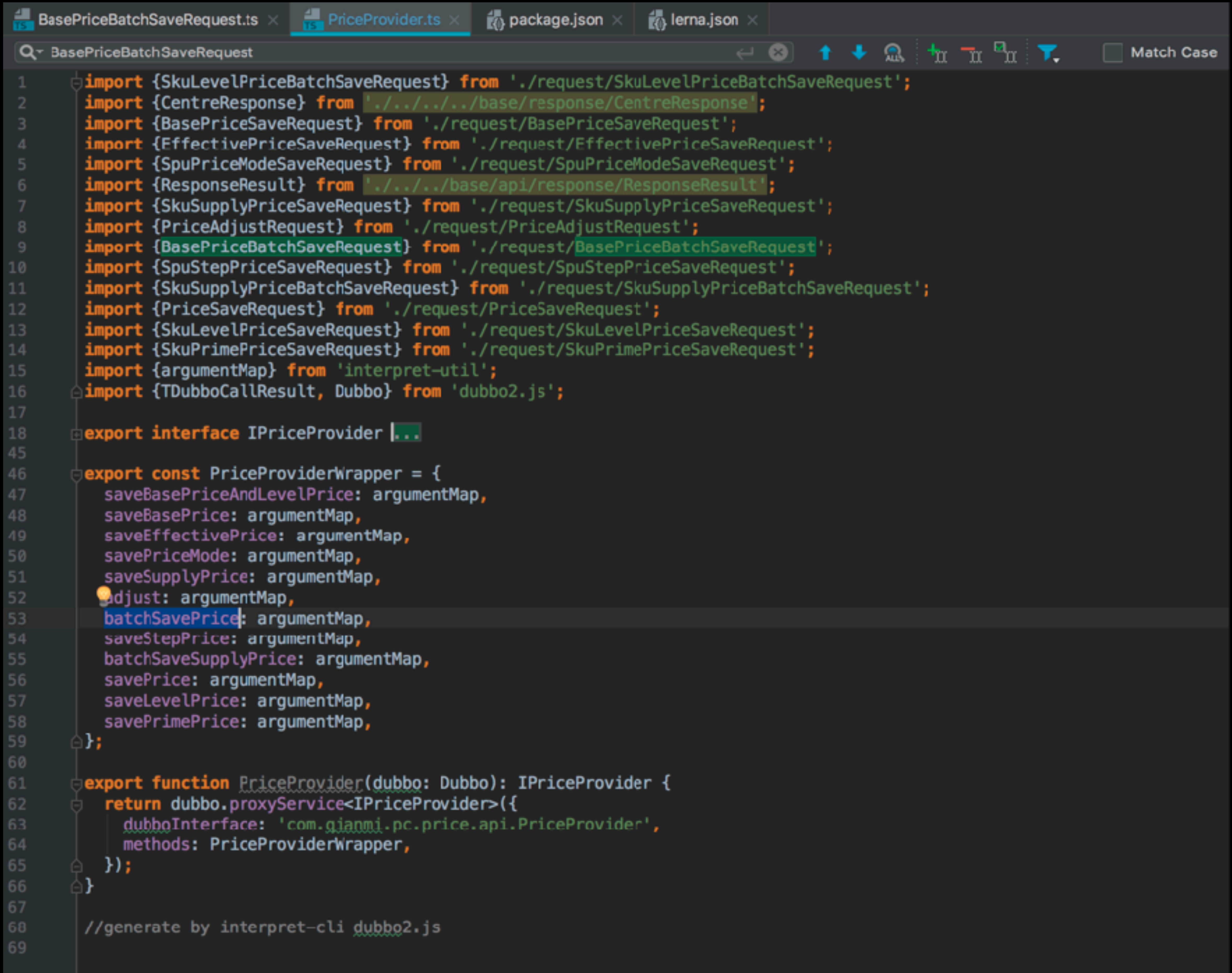
```
var encoder = new Encoder();
var long = {
  $class: 'java.lang.Long',
  $: 1
}
encoder.write(long); // long type

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



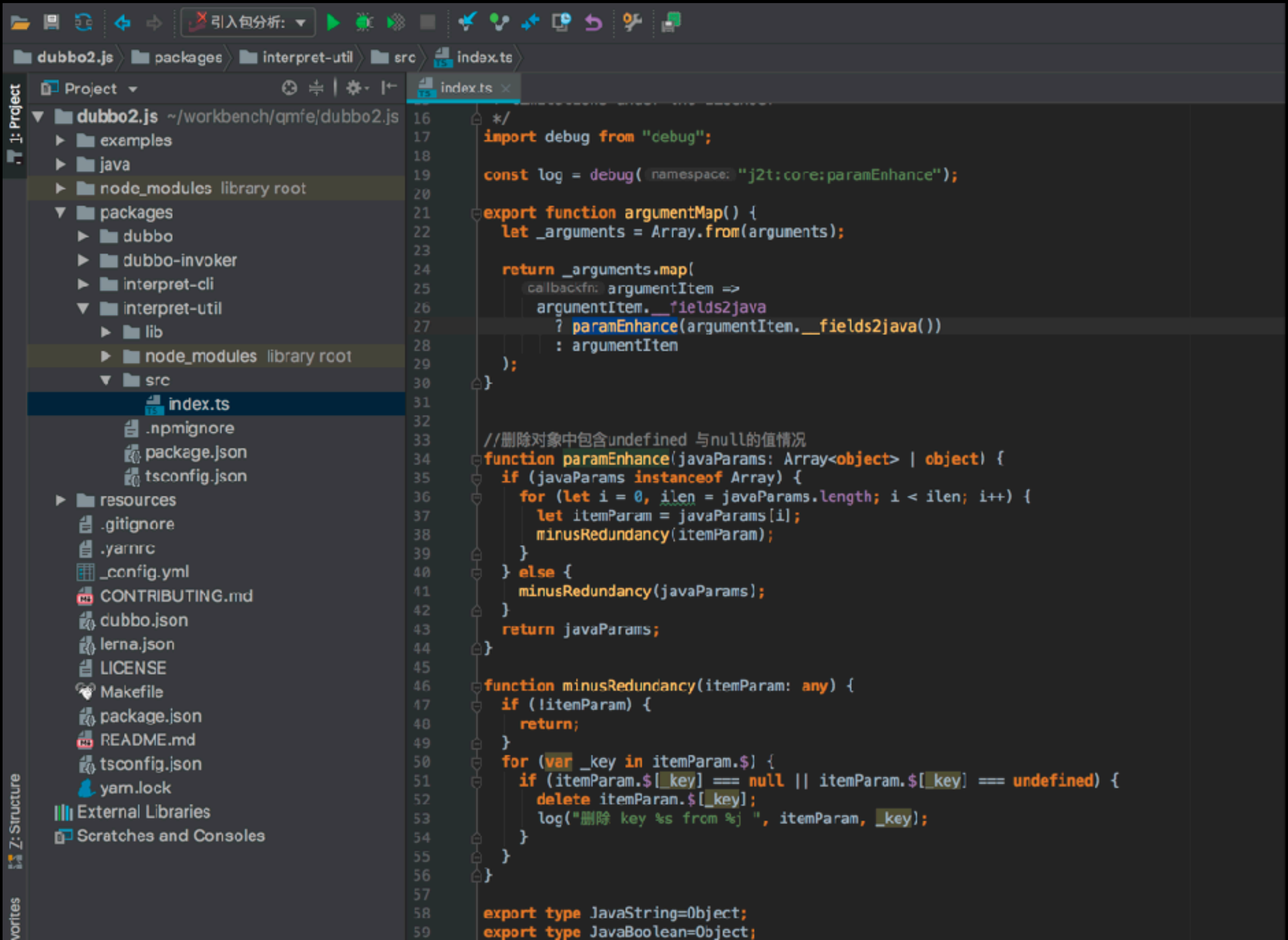
```
import {BasePriceBO} from './bean/BasePriceBO';
import java from 'js-to-java';

export interface IBasePriceBatchSaveRequest {
  saveBasePrices?: Array<BasePriceBO>;
  chainMasterId?: string;
  optUserCode?: string;
  optUserName?: string;
}

export class BasePriceBatchSaveRequest {
  constructor(params: IBasePriceBatchSaveRequest) {
    this.saveBasePrices = params.saveBasePrices;
    this.chainMasterId = params.chainMasterId;
    this.optUserCode = params.optUserCode;
    this.optUserName = params.optUserName;
  }

  saveBasePrices?: Array<BasePriceBO>;
  chainMasterId?: string;
  optUserCode?: string;
  optUserName?: string;

  __fields2java() {
    return {
      $class: 'com.qianmi.pc.price.api.request.BasePriceBatchSaveRequest',
      $: {
        saveBasePrices: java.Set(
          (this.saveBasePrices || []).map(paramItem => {
            return paramItem.__fields2java();
          }),
        ),
        chainMasterId: java.String(this.chainMasterId),
        optUserCode: java.String(this.optUserCode),
        optUserName: java.String(this.optUserName),
      },
    };
  }
}

//generate by interpret-cli dubbo2.js
```

how > 翻译流程 > 转换 > js to hessian



```
import {SkuLevelPriceBatchSaveRequest} from './request/SkuLevelPriceBatchSaveRequest';
import {CentreResponse} from './../../../base/response/CentreResponse';
import {BasePriceSaveRequest} from './request/BasePriceSaveRequest';
import {EffectivePriceSaveRequest} from './request/EffectivePriceSaveRequest';
import {SpuPriceModeSaveRequest} from './request/SpuPriceModeSaveRequest';
import {ResponseResult} from './../../base/api/response/ResponseResult';
import {SkuSupplyPriceSaveRequest} from './request/SkuSupplyPriceSaveRequest';
import {PriceAdjustRequest} from './request/PriceAdjustRequest';
import {BasePriceBatchSaveRequest} from './request/BasePriceBatchSaveRequest';
import {SpuStepPriceSaveRequest} from './request/SpuStepPriceSaveRequest';
import {SkuSupplyPriceBatchSaveRequest} from './request/SkuSupplyPriceBatchSaveRequest';
import {PriceSaveRequest} from './request/PriceSaveRequest';
import {SkuLevelPriceSaveRequest} from './request/SkuLevelPriceSaveRequest';
import {SkuPrimePriceSaveRequest} from './request/SkuPrimePriceSaveRequest';
import {argumentMap} from 'interpret-util';
import {TDubboCallResult, Dubbo} from 'dubbo2.js';

export interface IPriceProvider {...}

export const PriceProviderWrapper = {
  saveBasePriceAndLevelPrice: argumentMap,
  saveBasePrice: argumentMap,
  saveEffectivePrice: argumentMap,
  savePriceMode: argumentMap,
  saveSupplyPrice: argumentMap,
  adjust: argumentMap,
  batchSavePrice: argumentMap,
  saveStepPrice: argumentMap,
  batchSaveSupplyPrice: argumentMap,
  savePrice: argumentMap,
  saveLevelPrice: argumentMap,
  savePrimePrice: argumentMap,
};

export function PriceProvider(dubbo: Dubbo): IPriceProvider {
  return dubbo.proxyService<IPriceProvider>({
    dubboInterface: 'com.qianmi.pc.price.api.PriceProvider',
    methods: PriceProviderWrapper,
  });
}

//generate by interpret-cli dubbo2.js
```

how > 翻译流程 > 转换 > js to hessian



```
 */
import debug from "debug";

const log = debug( namespace: "j2t:core:paramEnhance");

export function argumentMap() {
  let _arguments = Array.from(arguments);

  return _arguments.map(
     callbackfn: argumentItem =>
      argumentItem.__fields2java
        ? paramEnhance(argumentItem.__fields2java())
        : argumentItem
  );
}


//删除对象中包含undefined 与null的值情况
function paramEnhance(javaParams: Array<object> | object) {
  if (javaParams instanceof Array) {
    for (let i = 0, ilen = javaParams.length; i < ilen; i++) {
      let itemParam = javaParams[i];
      minusRedundancy(itemParam);
    }
  } else {
    minusRedundancy(javaParams);
  }
  return javaParams;
}

function minusRedundancy(itemParam: any) {
  if (!itemParam) {
    return;
  }
  for (var _key in itemParam.$) {
    if (itemParam.$[_key] === null || itemParam.$[_key] === undefined) {
      delete itemParam.$[_key];
      log("删除 key %s from %j ", itemParam, _key);
    }
  }
}

export type JavaString=Object;
export type JavaBoolean=Object;
```

I come , i conquer !

0.抽取ast

from sourceCode

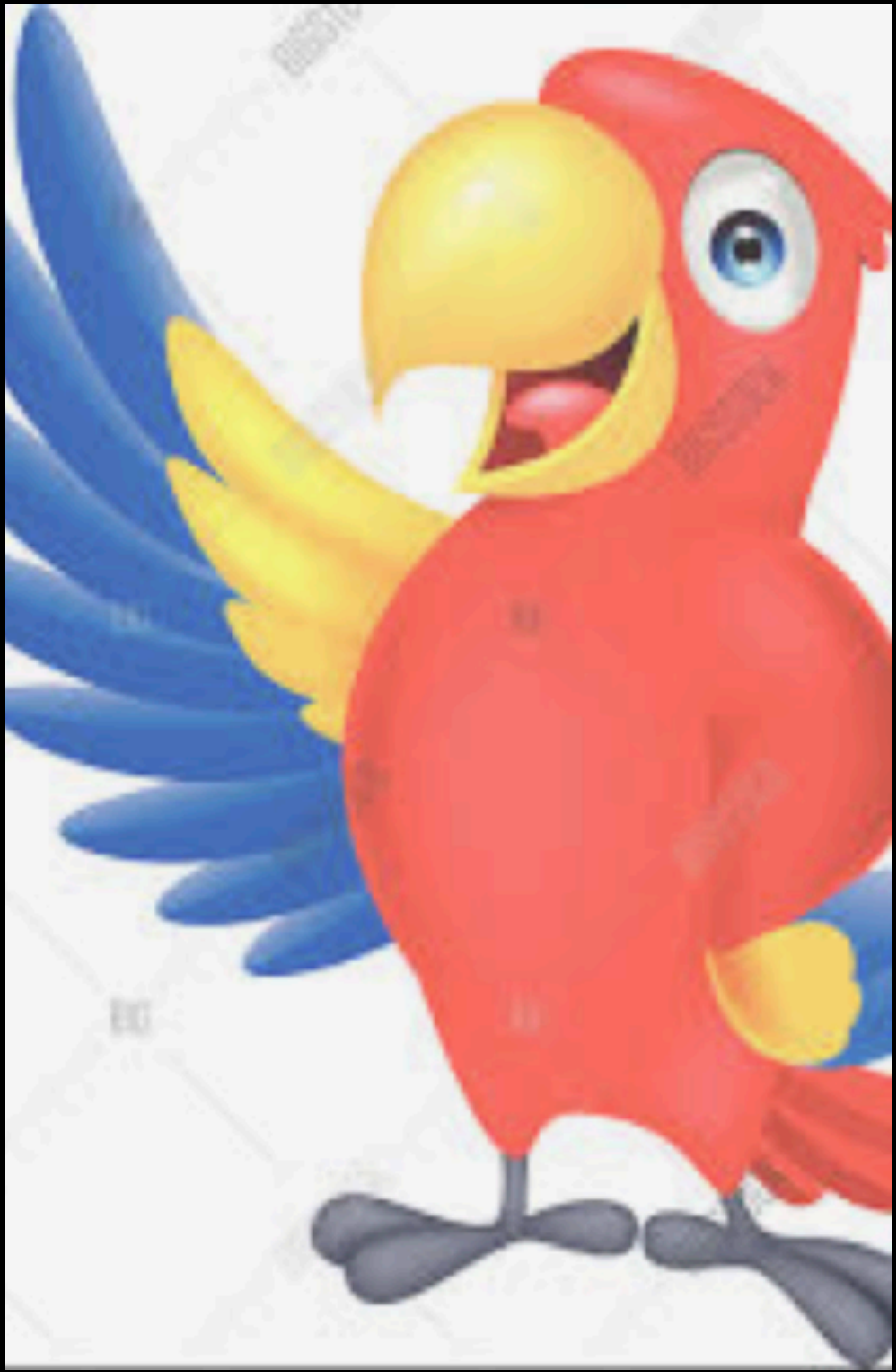
from byteCode

1.**java接口转换**

java => typescript

2.**参数转换**

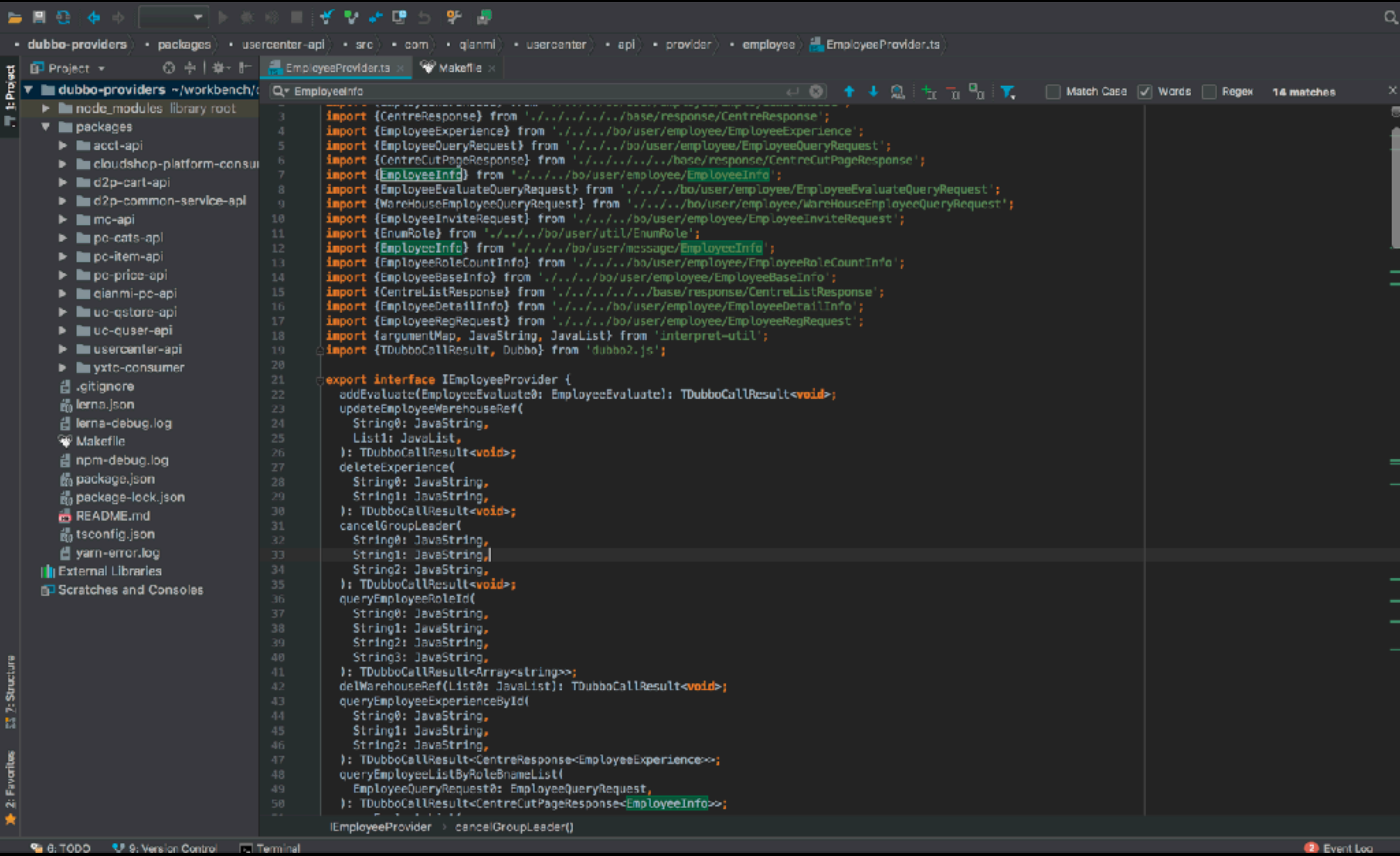
javascript => hession



翻译师 => Error

命名冲突

Error > 命名冲突

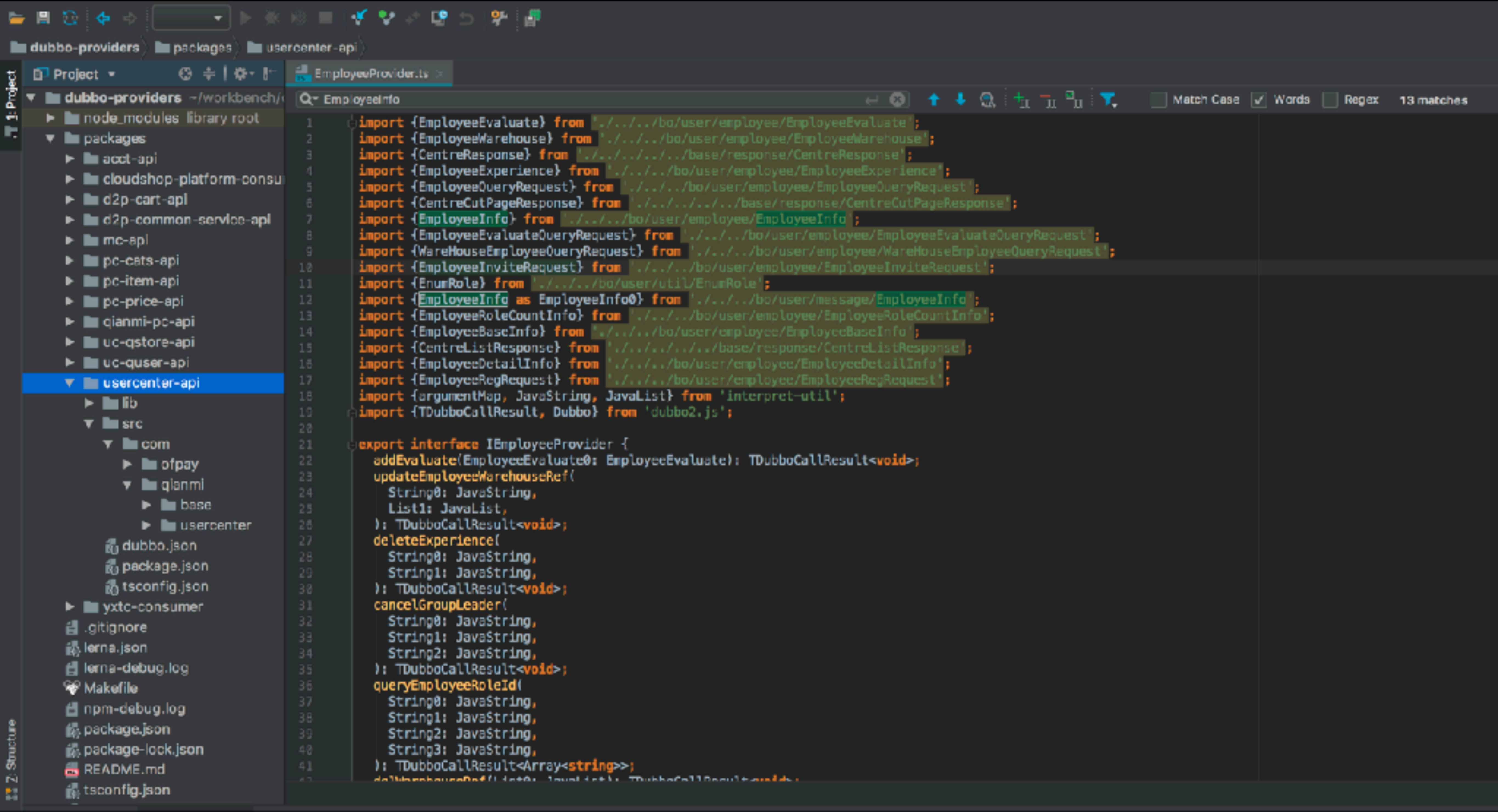


Q▾ EmployeeInfo

□ Match Case ☑ Words □ Regex 14 matches

```
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>>;
```

Error > 命名冲突

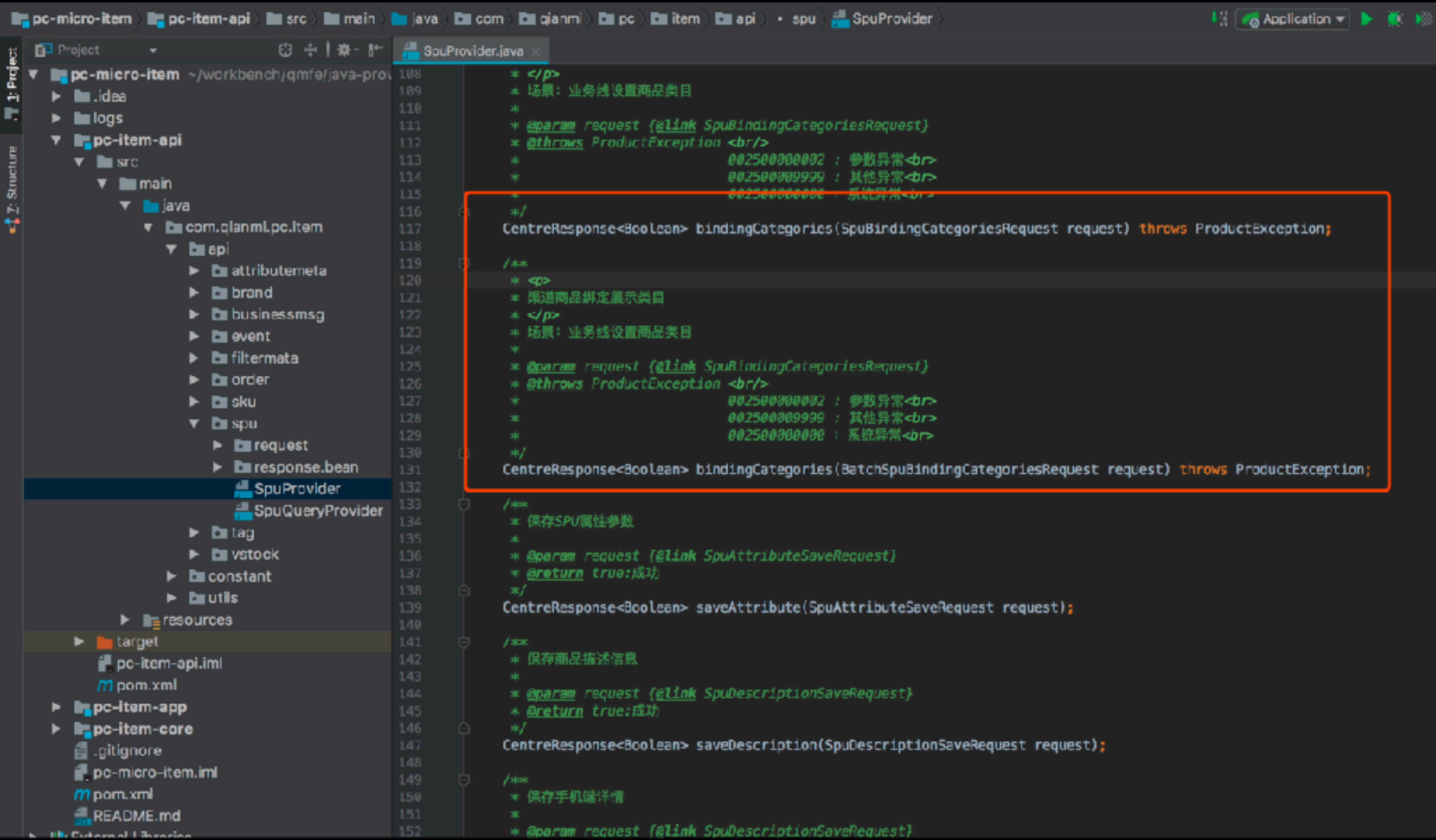


EmployeeInfo ☐ Match Case ☑ Words ☐ Regex 13 matches

```
import {EmployeeEvaluate} from './../../bo/user/employee/EmployeeEvaluate';
import {EmployeeWarehouse} from './../../bo/user/employee/EmployeeWarehouse';
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 as EmployeeInfo0} 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>;
```

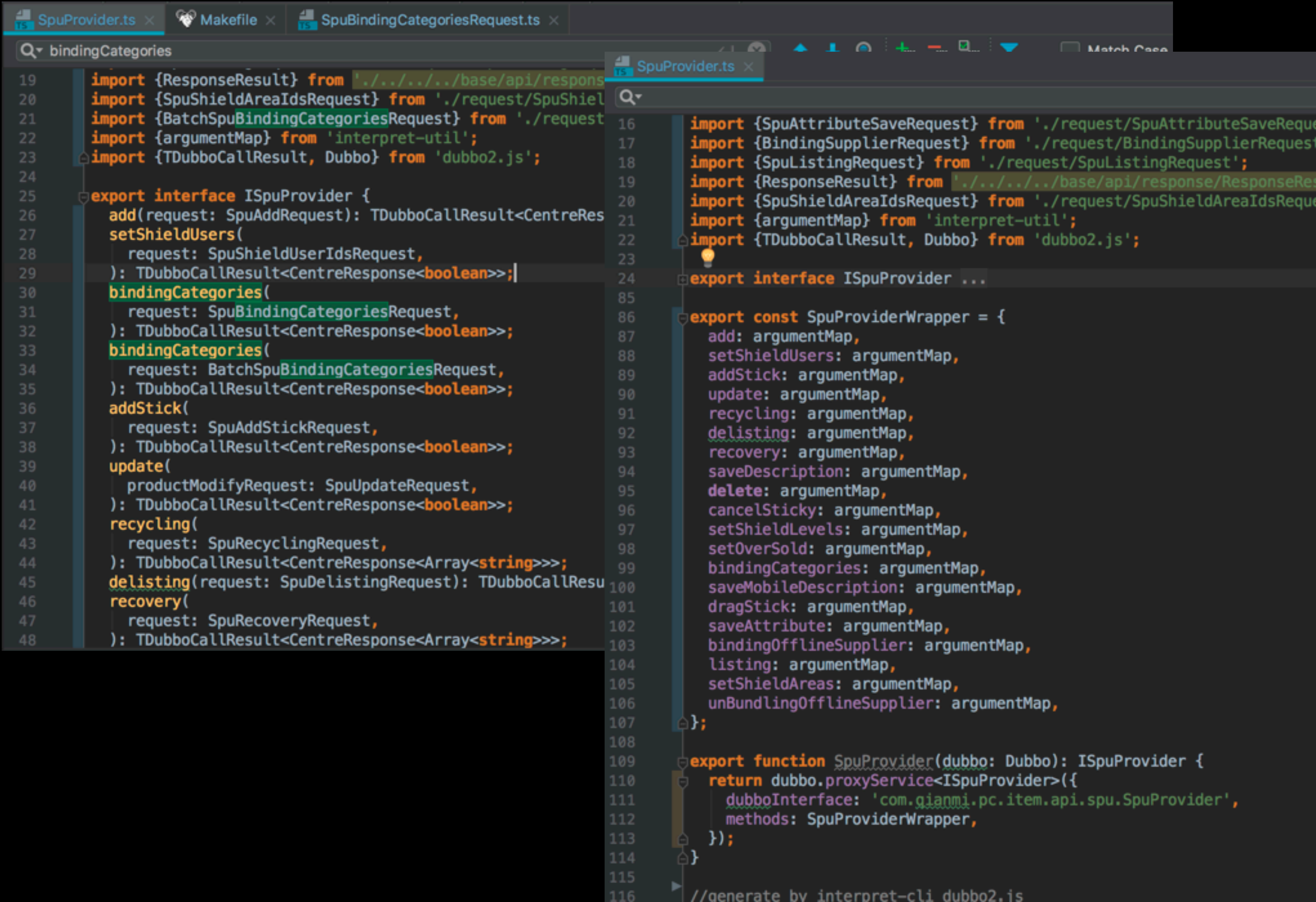
OverRide

Error >override



```java
     * </p>
     * 场景：业务线设置商品类目
     *
     * @param request {@link SpuBindingCategoriesRequest}
     * @throws ProductException <br/>
     *                              002500000002 : 参数异常<br>
     *                              002500009999 : 其他异常<br>
     *                              002500000000 : 系统异常<br>
     */
    CentreResponse<Boolean> bindingCategories(SpuBindingCategoriesRequest request) throws ProductException;

    /**
     * <p>
     * 渠道商品绑定展示类目
     * </p>
     * 场景：业务线设置商品类目
     *
     * @param request {@link SpuBindingCategoriesRequest}
     * @throws ProductException <br/>
     *                              002500000002 : 参数异常<br>
     *                              002500009999 : 其他异常<br>
     *                              002500000000 : 系统异常<br>
     */
    CentreResponse<Boolean> bindingCategories(BatchSpuBindingCategoriesRequest request) throws ProductException;

    /**
     * 保存SPU属性参数
     *
     * @param request {@link SpuAttributeSaveRequest}
     * @return true:成功
     */
    CentreResponse<Boolean> saveAttribute(SpuAttributeSaveRequest request);

    /**
     * 保存商品描述信息
     *
     * @param request {@link SpuDescriptionSaveRequest}
     * @return true:成功
     */
    CentreResponse<Boolean> saveDescription(SpuDescriptionSaveRequest request);

    /**
     * 保存手机端详情
     *
     * @param request {@link SpuDescriptionSaveRequest}
```

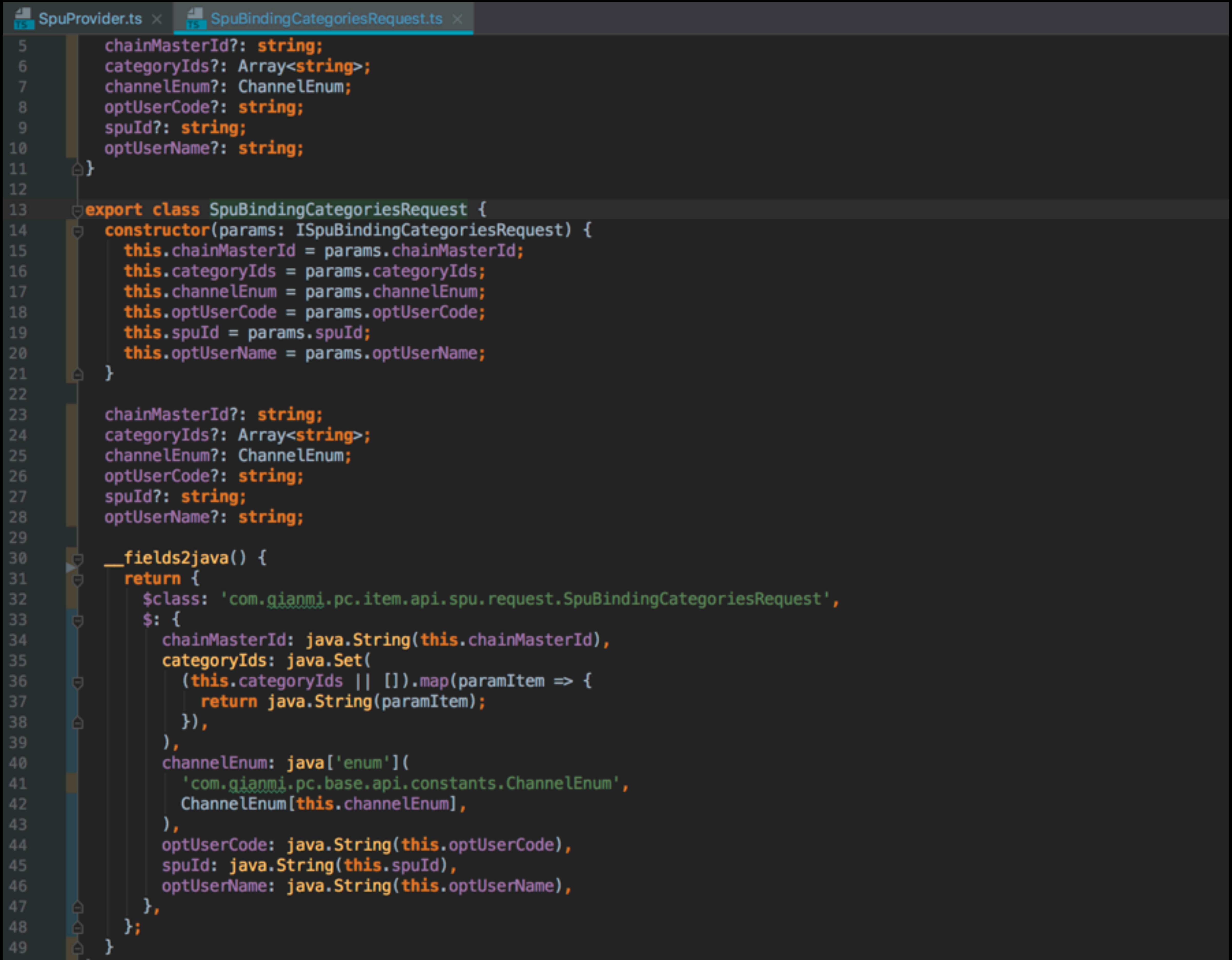
Error >override



```
import {ResponseResult} from './../../../base/api/respons
import {SpuShieldAreaIdsRequest} from './request/SpuShiel
import {BatchSpuBindingCategoriesRequest} from './request
import {argumentMap} from 'interpret-util';
import {TDubboCallResult, Dubbo} from 'dubbo2.js';

export interface ISpuProvider {
  add(request: SpuAddRequest): TDubboCallResult<CentreRes
  setShieldUsers(
    request: SpuShieldUserIdsRequest,
  ): TDubboCallResult<CentreResponse<boolean>>;
  bindingCategories(
    request: SpuBindingCategoriesRequest,
  ): TDubboCallResult<CentreResponse<boolean>>;
  bindingCategories(
    request: BatchSpuBindingCategoriesRequest,
  ): TDubboCallResult<CentreResponse<boolean>>;
  addStick(
    request: SpuAddStickRequest,
  ): TDubboCallResult<CentreResponse<boolean>>;
  update(
    productModifyRequest: SpuUpdateRequest,
  ): TDubboCallResult<CentreResponse<boolean>>;
  recycling(
    request: SpuRecyclingRequest,
  ): TDubboCallResult<CentreResponse<Array<string>>>;
  delisting(request: SpuDelistingRequest): TDubboCallResu
  recovery(
    request: SpuRecoveryRequest,
  ): TDubboCallResult<CentreResponse<Array<string>>>;
```

```
import {SpuAttributeSaveRequest} from './request/SpuAttributeSaveReque
import {BindingSupplierRequest} from './request/BindingSupplierReques
import {SpuListingRequest} from './request/SpuListingRequest';
import {ResponseResult} from './../../../base/api/response/ResponseRe
import {SpuShieldAreaIdsRequest} from './request/SpuShieldAreaIdsReque
import {argumentMap} from 'interpret-util';
import {TDubboCallResult, Dubbo} from 'dubbo2.js';

export interface ISpuProvider ...

export const SpuProviderWrapper = {
  add: argumentMap,
  setShieldUsers: argumentMap,
  addStick: argumentMap,
  update: argumentMap,
  recycling: argumentMap,
  delisting: argumentMap,
  recovery: argumentMap,
  saveDescription: argumentMap,
  delete: argumentMap,
  cancelSticky: argumentMap,
  setShieldLevels: argumentMap,
  setOverSold: argumentMap,
  bindingCategories: argumentMap,
  saveMobileDescription: argumentMap,
  dragStick: argumentMap,
  saveAttribute: argumentMap,
  bindingOfflineSupplier: argumentMap,
  listing: argumentMap,
  setShieldAreas: argumentMap,
  unBundlingOfflineSupplier: argumentMap,
};

export function SpuProvider(dubbo: Dubbo): ISpuProvider {
  return dubbo.proxyService<ISpuProvider>({
    dubboInterface: 'com.qianmi.pc.item.api.spu.SpuProvider',
    methods: SpuProviderWrapper,
  });
}

//generate by interpret-cli dubbo2.js
```

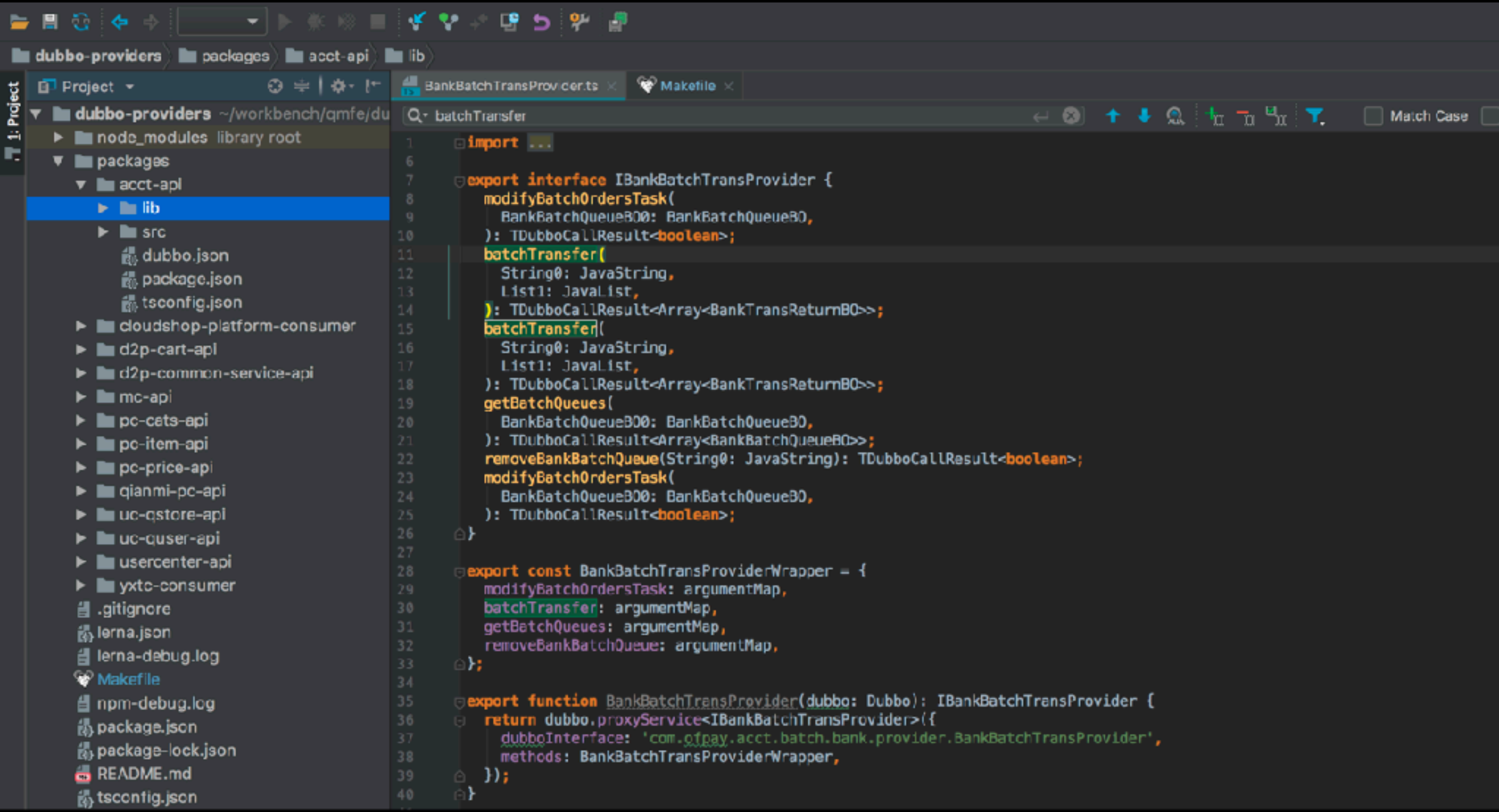
Error >override



```
  chainMasterId?: string;
  categoryIds?: Array<string>;
  channelEnum?: ChannelEnum;
  optUserCode?: string;
  spuId?: string;
  optUserName?: string;
}

export class SpuBindingCategoriesRequest {
  constructor(params: ISpuBindingCategoriesRequest) {
    this.chainMasterId = params.chainMasterId;
    this.categoryIds = params.categoryIds;
    this.channelEnum = params.channelEnum;
    this.optUserCode = params.optUserCode;
    this.spuId = params.spuId;
    this.optUserName = params.optUserName;
  }

  chainMasterId?: string;
  categoryIds?: Array<string>;
  channelEnum?: ChannelEnum;
  optUserCode?: string;
  spuId?: string;
  optUserName?: string;

  __fields2java() {
    return {
      $class: 'com.qianmi.pc.item.api.spu.request.SpuBindingCategoriesRequest',
      $: {
        chainMasterId: java.String(this.chainMasterId),
        categoryIds: java.Set(
          (this.categoryIds || []).map(paramItem => {
            return java.String(paramItem);
          }),
        ),
        channelEnum: java['enum'](
          'com.qianmi.pc.base.api.constants.ChannelEnum',
          ChannelEnum[this.channelEnum],
        ),
        optUserCode: java.String(this.optUserCode),
        spuId: java.String(this.spuId),
        optUserName: java.String(this.optUserName),
      },
    };
  }
```

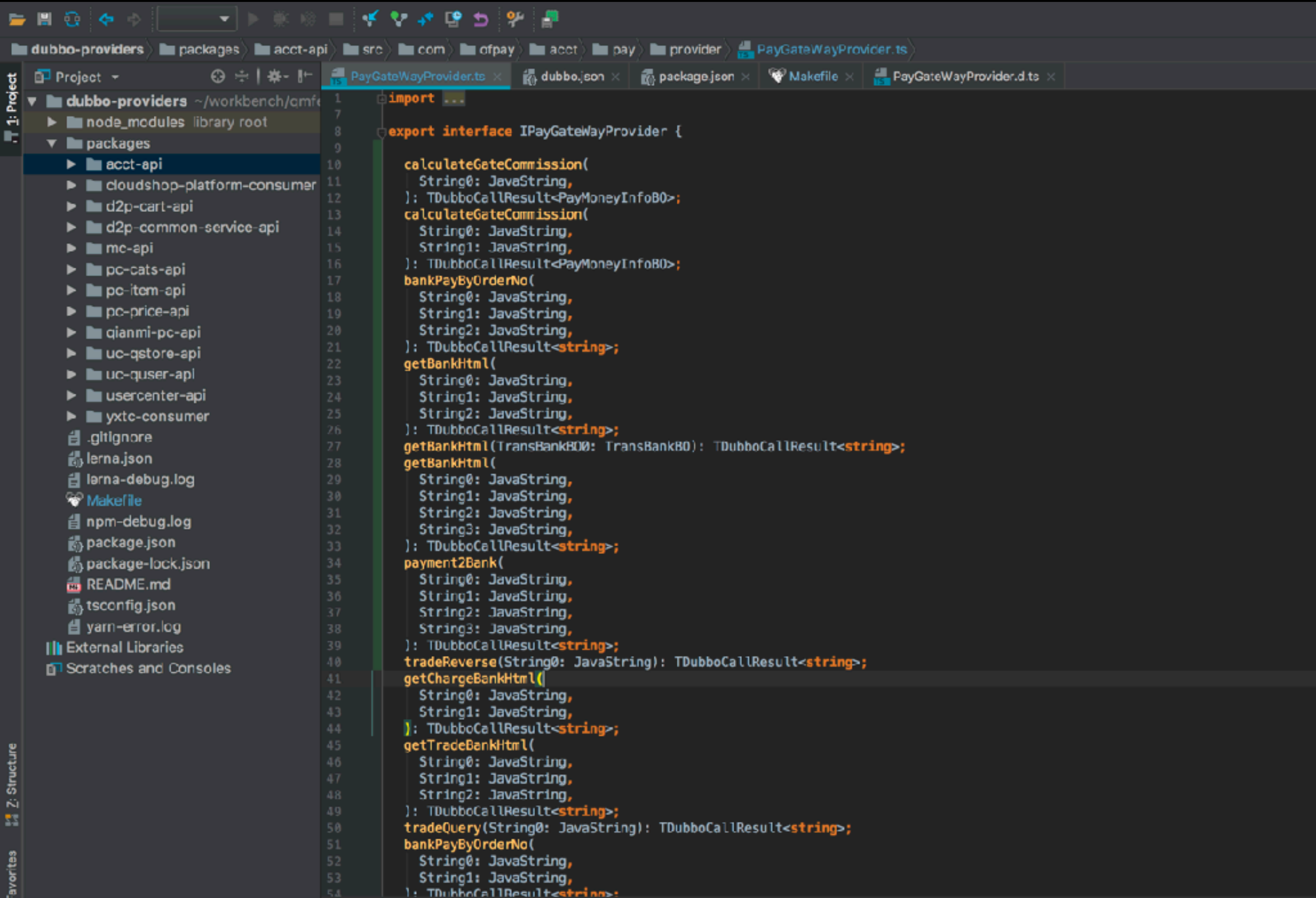
当基本类型遇上overRide

Error >override



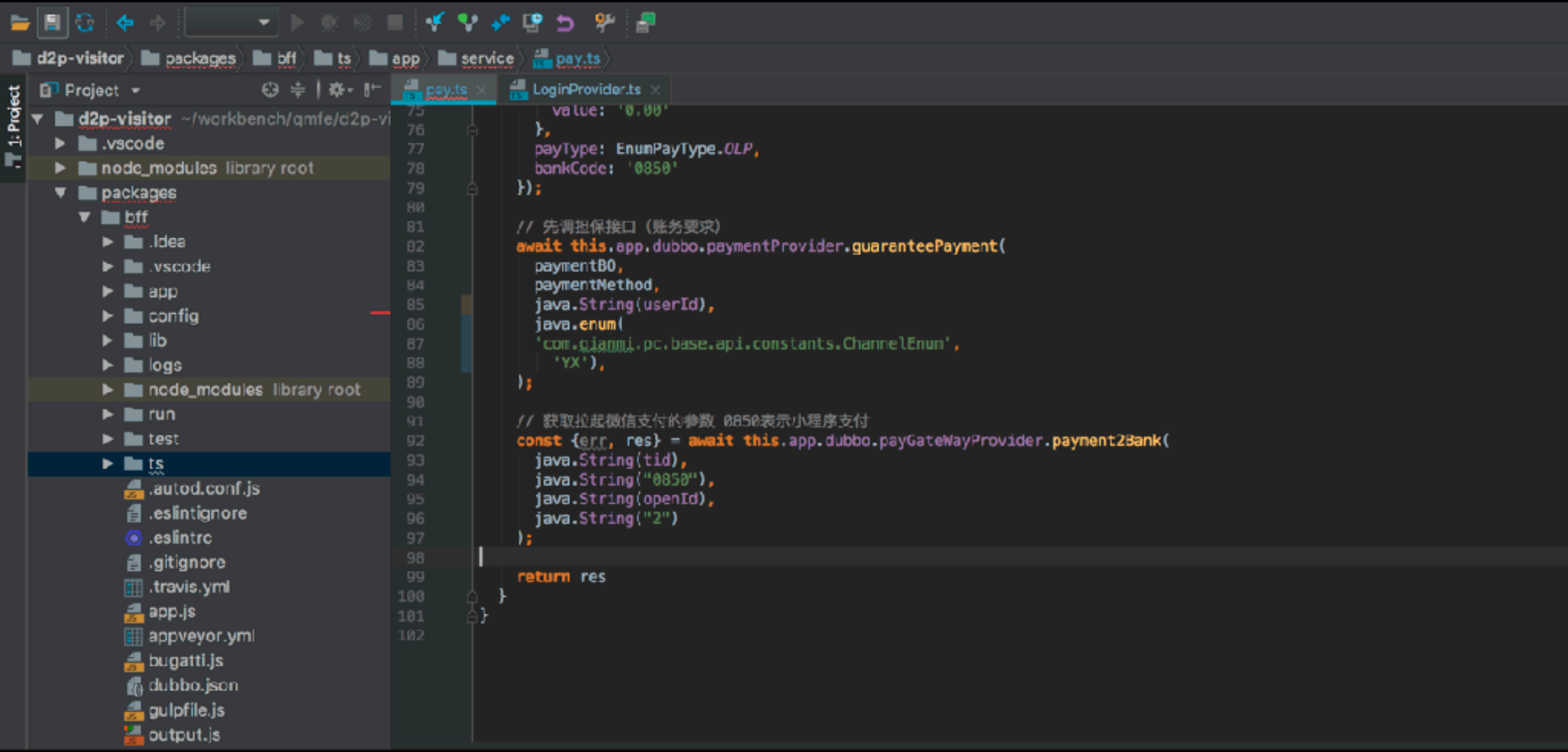
```typescript
import ...

export interface IBankBatchTransProvider {
  modifyBatchOrdersTask(
    BankBatchQueueBO0: BankBatchQueueBO,
  ): TDubboCallResult<boolean>;
  batchTransfer(
    String0: JavaString,
    List1: JavaList,
  ): TDubboCallResult<Array<BankTransReturnBO>>;
  batchTransfer(
    String0: JavaString,
    List1: JavaList,
  ): TDubboCallResult<Array<BankTransReturnBO>>;
  getBatchQueues(
    BankBatchQueueBO0: BankBatchQueueBO,
  ): TDubboCallResult<Array<BankBatchQueueBO>>;
  removeBankBatchQueue(String0: JavaString): TDubboCallResult<boolean>;
  modifyBatchOrdersTask(
    BankBatchQueueBO0: BankBatchQueueBO,
  ): TDubboCallResult<boolean>;
}

export const BankBatchTransProviderWrapper = {
  modifyBatchOrdersTask: argumentMap,
  batchTransfer: argumentMap,
  getBatchQueues: argumentMap,
  removeBankBatchQueue: argumentMap,
};

export function BankBatchTransProvider(dubbo: Dubbo): IBankBatchTransProvider {
  return dubbo.proxyService<IBankBatchTransProvider>({
    dubboInterface: 'com.ofpay.acct.batch.bank.provider.BankBatchTransProvider',
    methods: BankBatchTransProviderWrapper,
  });
}
```

Error >override



```
import ...

export interface IPayGateWayProvider {

  calculateGateCommission(
    String0: JavaString,
  ): TDubboCallResult<PayMoneyInfoBO>;
  calculateGateCommission(
    String0: JavaString,
    String1: JavaString,
  ): TDubboCallResult<PayMoneyInfoBO>;
  bankPayByOrderNo(
    String0: JavaString,
    String1: JavaString,
    String2: JavaString,
  ): TDubboCallResult<string>;
  getBankHtml(
    String0: JavaString,
    String1: JavaString,
    String2: JavaString,
  ): TDubboCallResult<string>;
  getBankHtml(TransBankBO0: TransBankBO): TDubboCallResult<string>;
  getBankHtml(
    String0: JavaString,
    String1: JavaString,
    String2: JavaString,
    String3: JavaString,
  ): TDubboCallResult<string>;
  payment2Bank(
    String0: JavaString,
    String1: JavaString,
    String2: JavaString,
    String3: JavaString,
  ): TDubboCallResult<string>;
  tradeReverse(String0: JavaString): TDubboCallResult<string>;
  getChargeBankHtml(
    String0: JavaString,
    String1: JavaString,
  ): TDubboCallResult<string>;
  getTradeBankHtml(
    String0: JavaString,
    String1: JavaString,
    String2: JavaString,
  ): TDubboCallResult<string>;
  tradeQuery(String0: JavaString): TDubboCallResult<string>;
  bankPayByOrderNo(
    String0: JavaString,
    String1: JavaString,
  ): TDubboCallResult<string>;
```

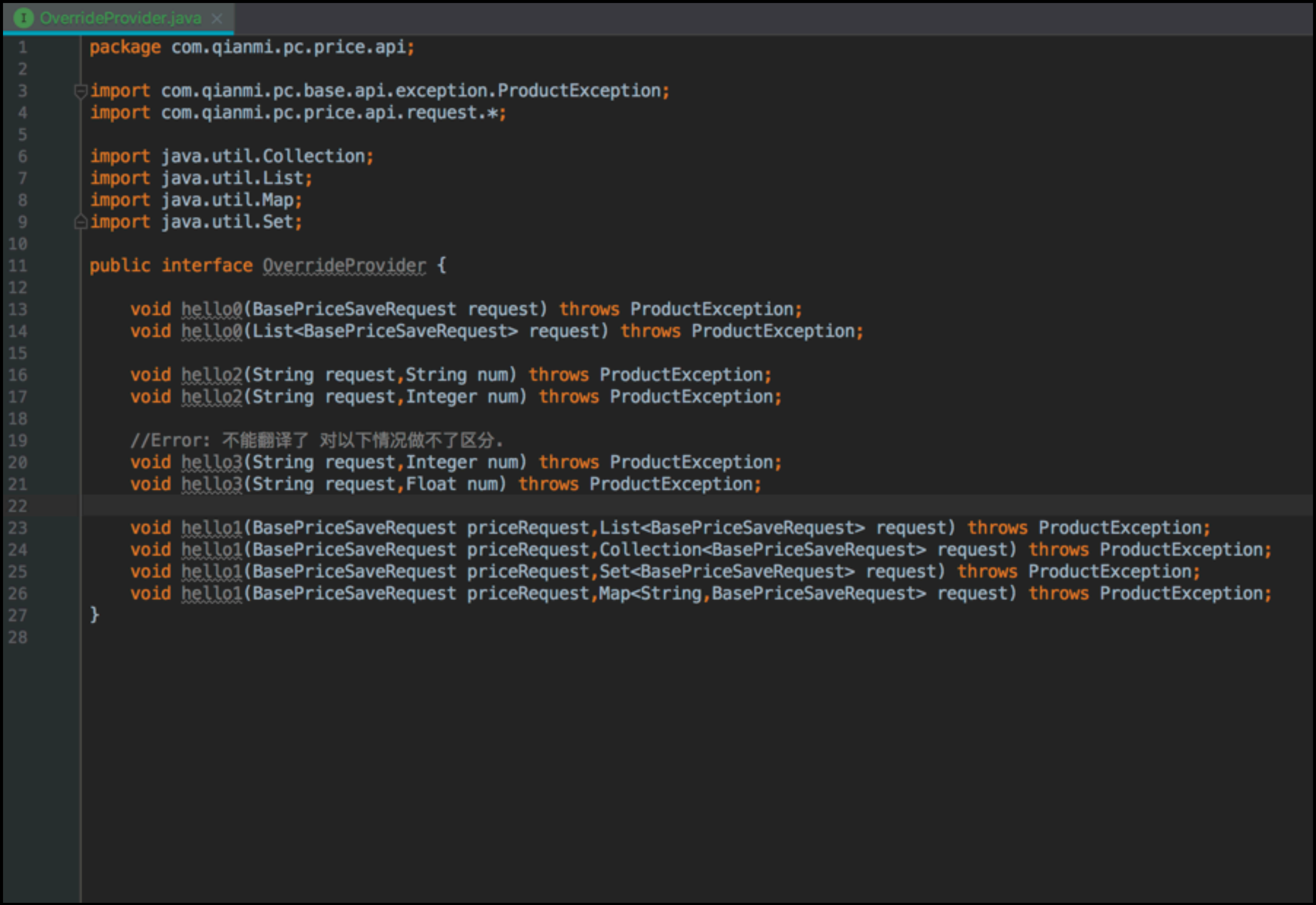
Error >override



```
        value: '0.00'
      },
      payType: EnumPayType.OLP,
      bankCode: '0850'
    });

    // 先调担保接口 (账务要求)
    await this.app.dubbo.paymentProvider.guaranteePayment(
      paymentBO,
      paymentMethod,
      java.String(userId),
      java.enum(
      'com.qianmi.pc.base.api.constants.ChannelEnum',
        'YX'),
    );

    // 获取拉起微信支付的参数 0850表示小程序支付
    const {err, res} = await this.app.dubbo.payGateWayProvider.payment2Bank(
      java.String(tid),
      java.String("0850"),
      java.String(openId),
      java.String("2")
    );

    return res
  }
}
```

为什么不翻译

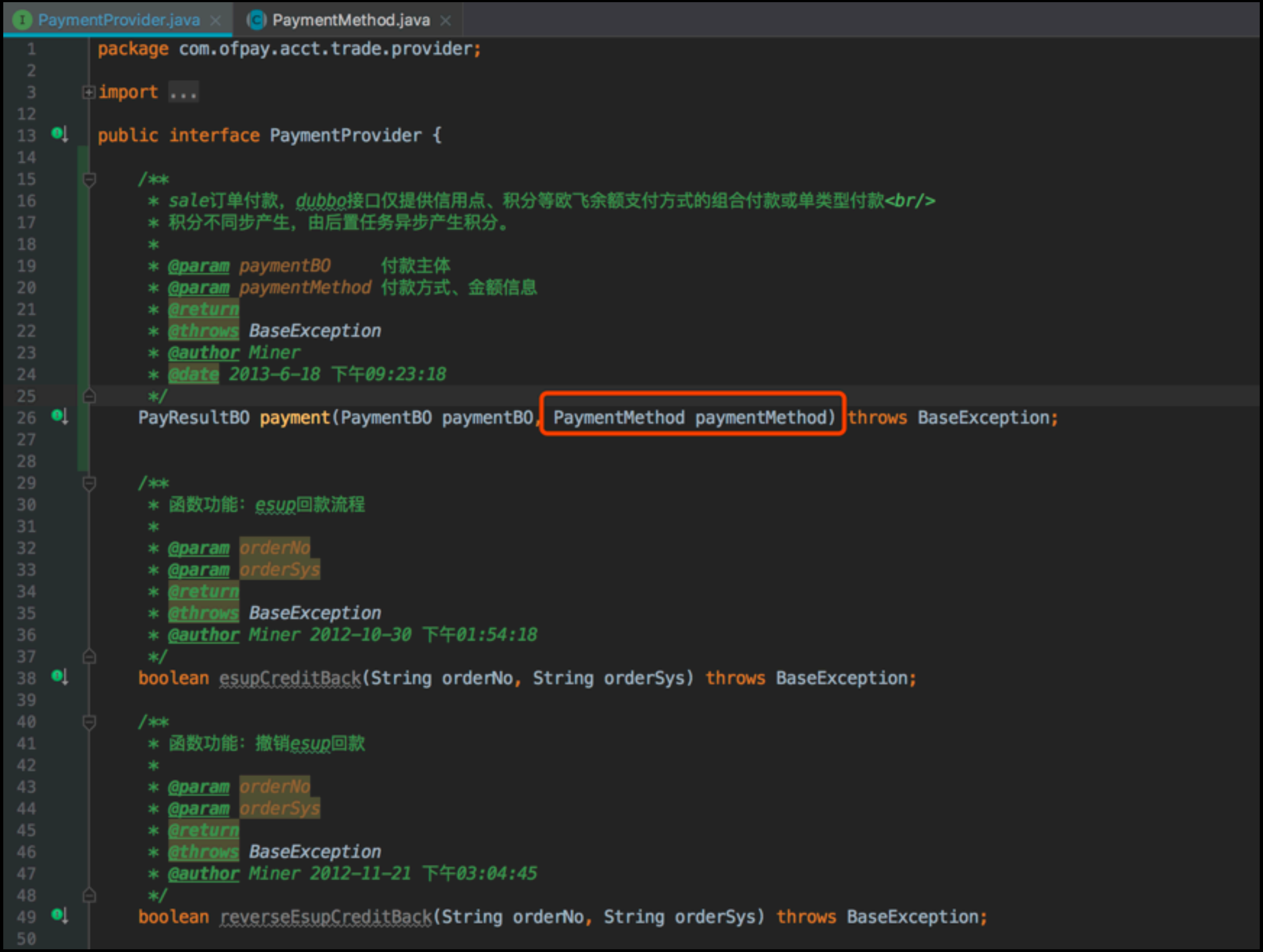
Error >override



```java
package com.qianmi.pc.price.api;

import com.qianmi.pc.base.api.exception.ProductException;
import com.qianmi.pc.price.api.request.*;

import java.util.Collection;
import java.util.List;
import java.util.Map;
import java.util.Set;

public interface OverrideProvider {

    void hello0(BasePriceSaveRequest request) throws ProductException;
    void hello0(List<BasePriceSaveRequest> request) throws ProductException;

    void hello2(String request,String num) throws ProductException;
    void hello2(String request,Integer num) throws ProductException;

    //Error: 不能翻译了 对以下情况做不了区分.
    void hello3(String request,Integer num) throws ProductException;
    void hello3(String request,Float num) throws ProductException;

    void hello1(BasePriceSaveRequest priceRequest,List<BasePriceSaveRequest> request) throws ProductException;
    void hello1(BasePriceSaveRequest priceRequest,Collection<BasePriceSaveRequest> request) throws ProductException;
    void hello1(BasePriceSaveRequest priceRequest,Set<BasePriceSaveRequest> request) throws ProductException;
    void hello1(BasePriceSaveRequest priceRequest,Map<String,BasePriceSaveRequest> request) throws ProductException;
}
```

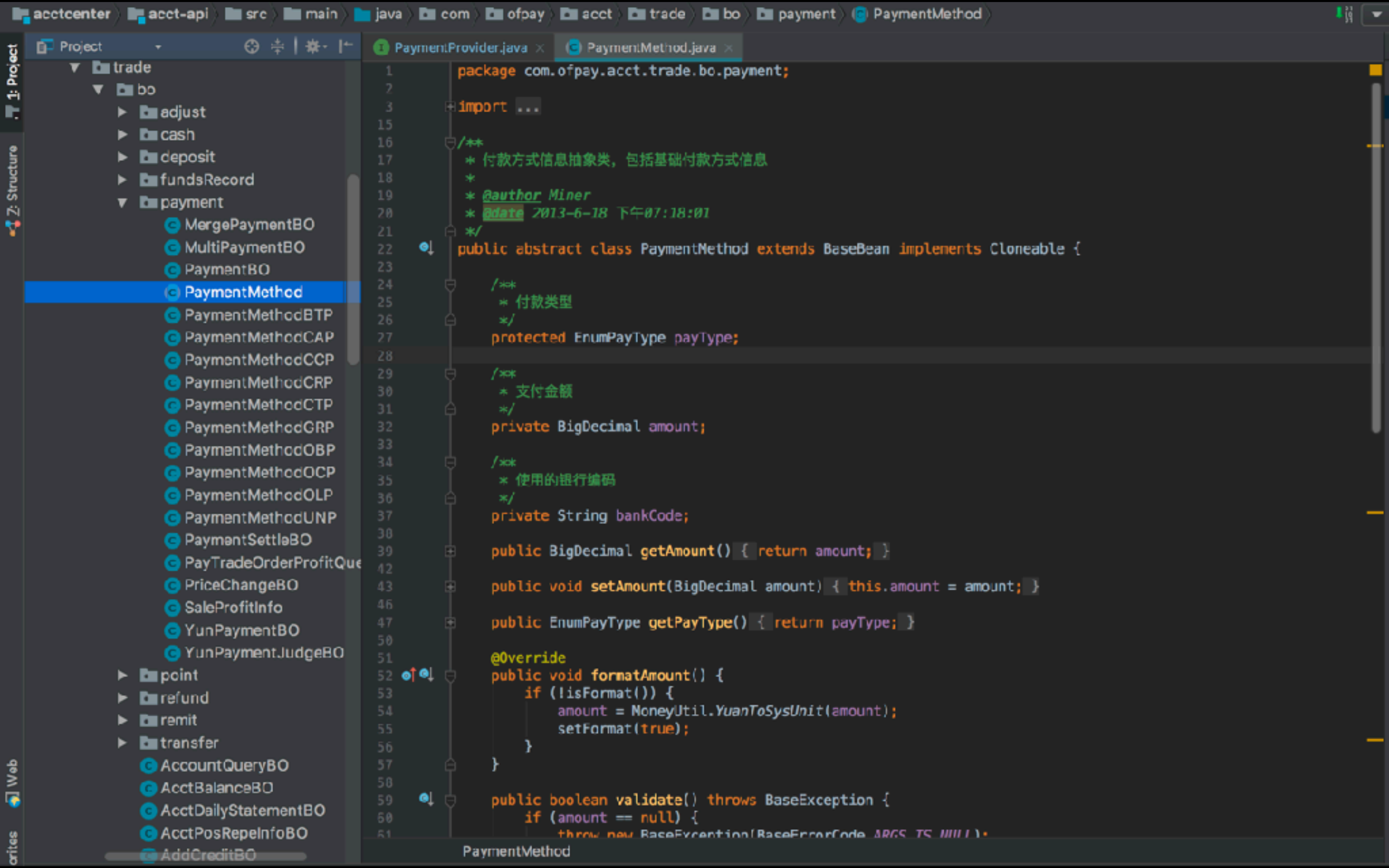
抽象类参数

Error >抽象类参数



```java
package com.ofpay.acct.trade.provider;

import ...

public interface PaymentProvider {

    /**
     * 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/>
     * 积分不同步产生，由后置任务异步产生积分。
     *
     * @param paymentBO     付款主体
     * @param paymentMethod 付款方式、金额信息
     * @return
     * @throws BaseException
     * @author Miner
     * @date 2013-6-18 下午09:23:18
     */
    PayResultBO payment(PaymentBO paymentBO, PaymentMethod paymentMethod) throws BaseException;


    /**
     * 函数功能：esup回款流程
     *
     * @param orderNo
     * @param orderSys
     * @return
     * @throws BaseException
     * @author Miner 2012-10-30 下午01:54:18
     */
    boolean esupCreditBack(String orderNo, String orderSys) throws BaseException;

    /**
     * 函数功能：撤销esup回款
     *
     * @param orderNo
     * @param orderSys
     * @return
     * @throws BaseException
     * @author Miner 2012-11-21 下午03:04:45
     */
    boolean reverseEsupCreditBack(String orderNo, String orderSys) throws BaseException;

```

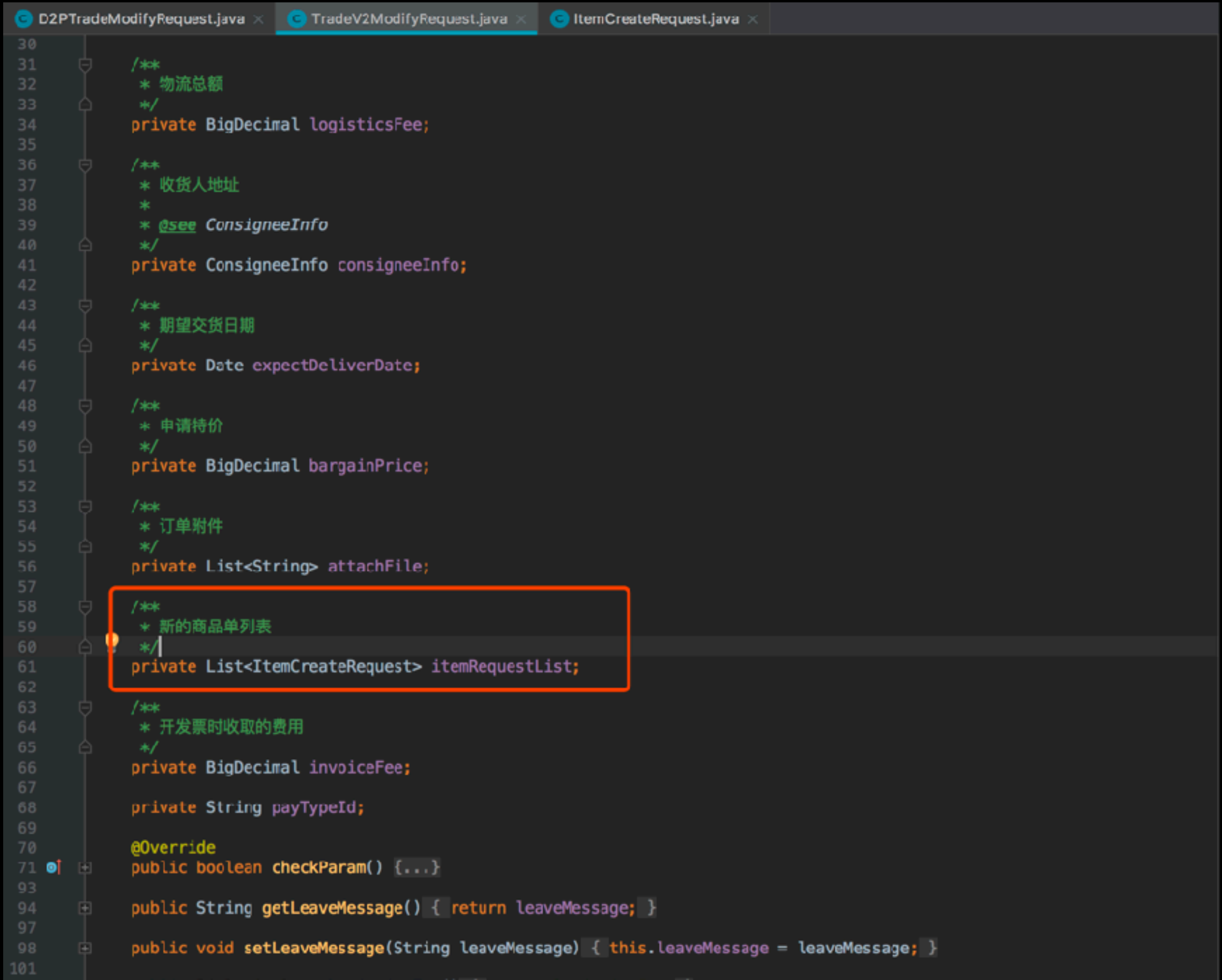
Error >抽象类参数



```java
package com.ofpay.acct.trade.bo.payment;

import ...

/**
 * 付款方式信息抽象类，包括基础付款方式信息
 *
 * @author Miner
 * @date 2013-6-18 下午07:18:01
 */
public abstract class PaymentMethod extends BaseBean implements Cloneable {

    /**
     * 付款类型
     */
    protected EnumPayType payType;

    /**
     * 支付金额
     */
    private BigDecimal amount;

    /**
     * 使用的银行编码
     */
    private String bankCode;

    public BigDecimal getAmount() { return amount; }

    public void setAmount(BigDecimal amount) { this.amount = amount; }

    public EnumPayType getPayType() { return payType; }

    @Override
    public void formatAmount() {
        if (!isFormat()) {
            amount = MoneyUtil.YuanToSysUnit(amount);
            setFormat(true);
        }
    }

    public boolean validate() throws BaseException {
        if (amount == null) {
            throw new BaseException(BaseErrorCode.ARGS_IS_NULL);
```

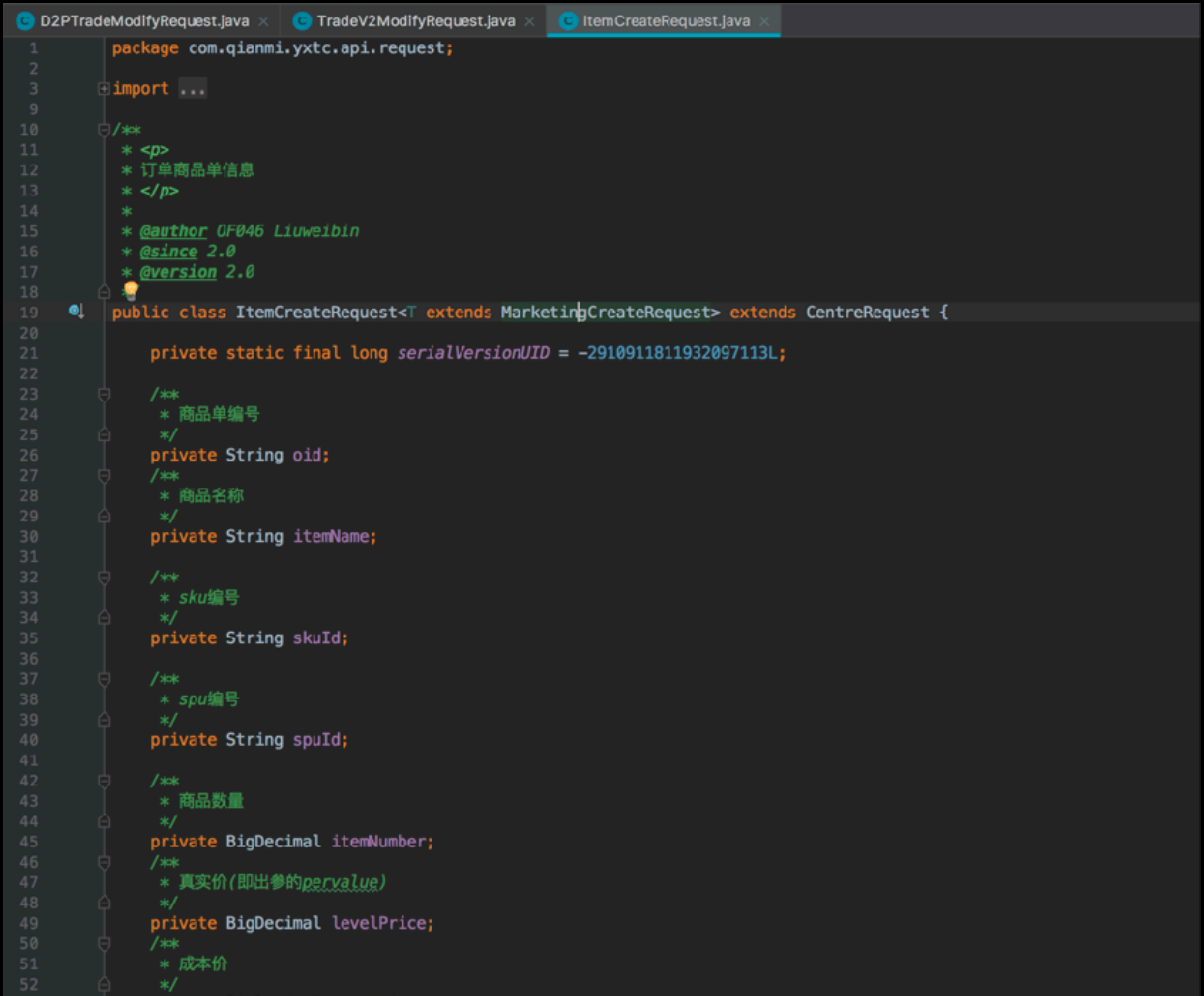
泛型?

Error > 泛型



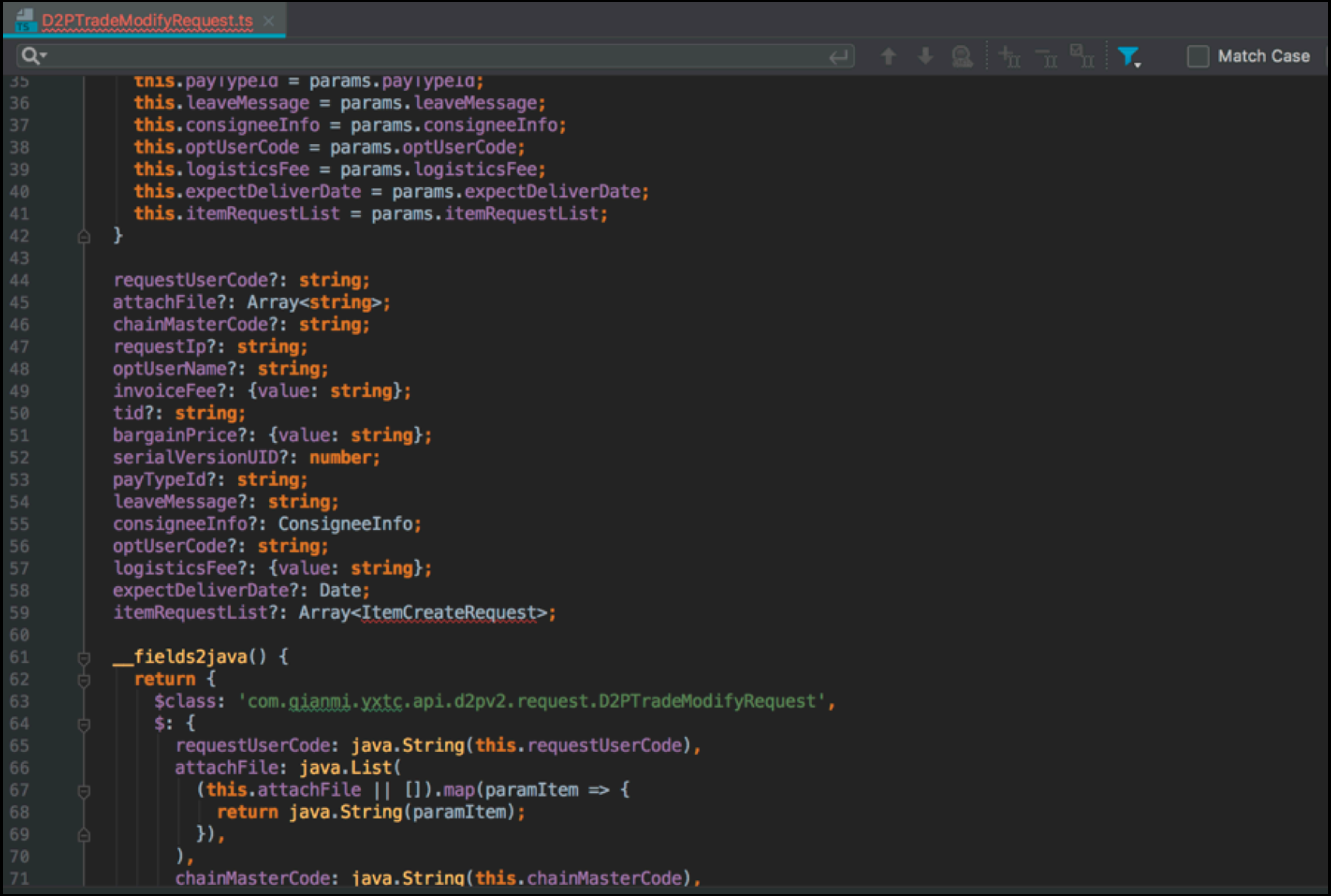
```java

    /**
     * 物流总额
     */
    private BigDecimal logisticsFee;

    /**
     * 收货人地址
     *
     * @see ConsigneeInfo
     */
    private ConsigneeInfo consigneeInfo;

    /**
     * 期望交货日期
     */
    private Date expectDeliverDate;

    /**
     * 申请特价
     */
    private BigDecimal bargainPrice;

    /**
     * 订单附件
     */
    private List<String> attachFile;

    /**
     * 新的商品单列表
     */
    private List<ItemCreateRequest> itemRequestList;

    /**
     * 开发票时收取的费用
     */
    private BigDecimal invoiceFee;

    private String payTypeId;

    @Override
    public boolean checkParam() {...}

    public String getLeaveMessage() { return leaveMessage; }

    public void setLeaveMessage(String leaveMessage) { this.leaveMessage = leaveMessage; }

```

Error > 泛型



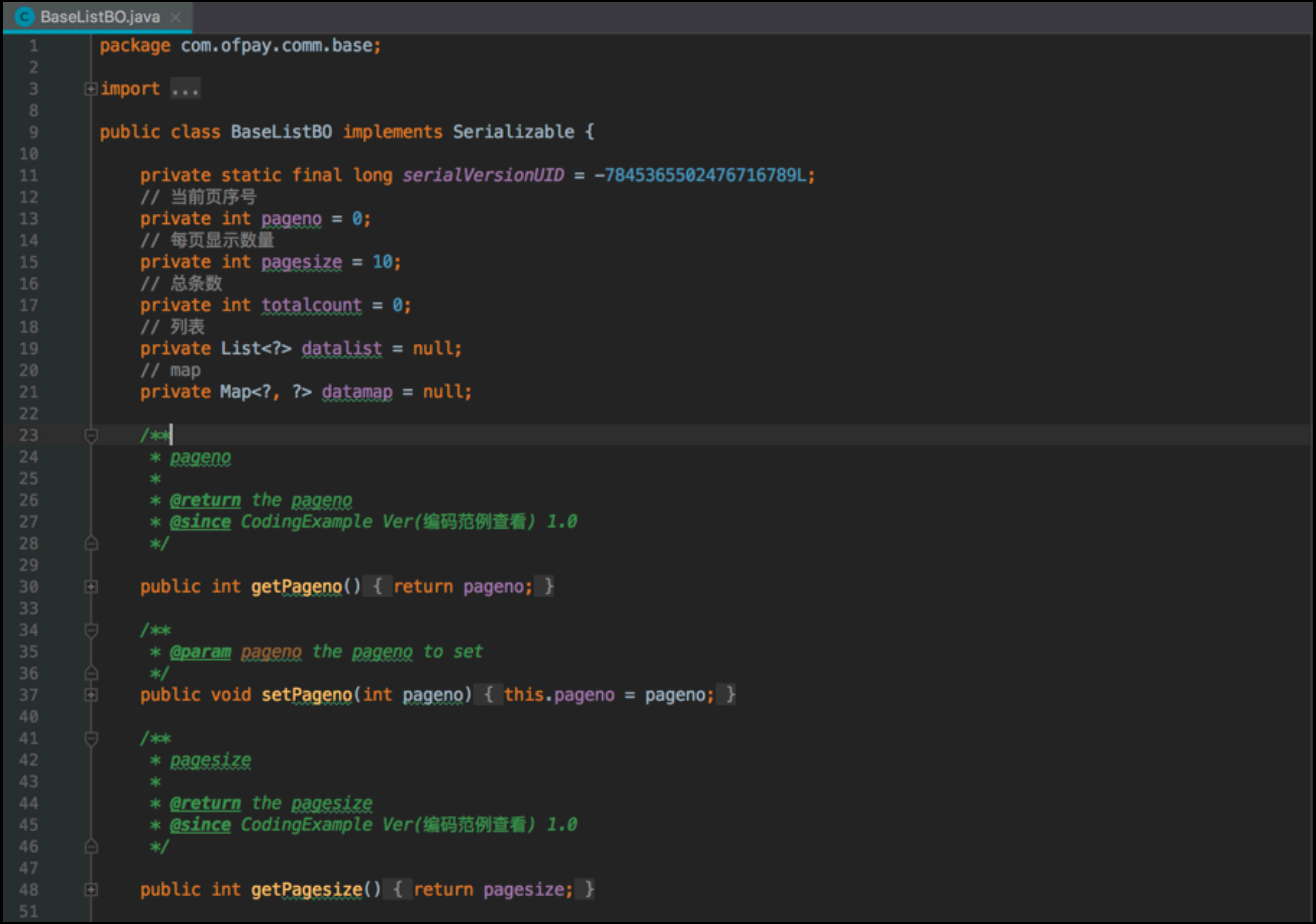
```java
package com.qianmi.yxtc.api.request;

import ...

/**
 * <p>
 * 订单商品单信息
 * </p>
 *
 * @author OF046 Liuweibin
 * @since 2.0
 * @version 2.0
 */
public class ItemCreateRequest<T extends MarketingCreateRequest> extends CentreRequest {

    private static final long serialVersionUID = -2910911811932097113L;

    /**
     * 商品单编号
     */
    private String oid;
    /**
     * 商品名称
     */
    private String itemName;

    /**
     * sku编号
     */
    private String skuId;

    /**
     * spu编号
     */
    private String spuId;

    /**
     * 商品数量
     */
    private BigDecimal itemNumber;
    /**
     * 真实价(即出参的pervalue)
     */
    private BigDecimal levelPrice;
    /**
     * 成本价
     */
```

Error > 泛型



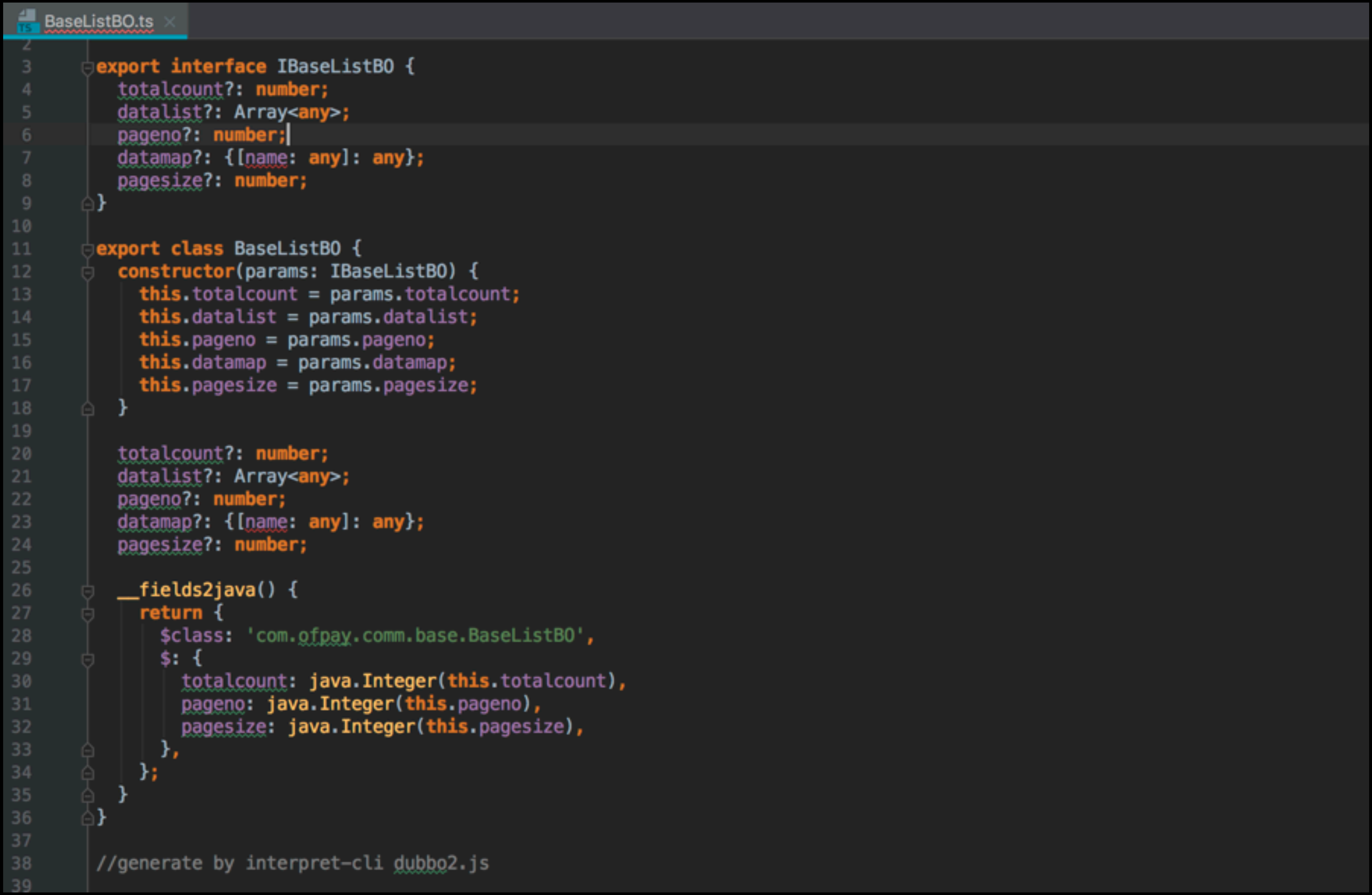
```
    this.payTypeId = params.payTypeId;
    this.leaveMessage = params.leaveMessage;
    this.consigneeInfo = params.consigneeInfo;
    this.optUserCode = params.optUserCode;
    this.logisticsFee = params.logisticsFee;
    this.expectDeliverDate = params.expectDeliverDate;
    this.itemRequestList = params.itemRequestList;
  }

  requestUserCode?: string;
  attachFile?: Array<string>;
  chainMasterCode?: string;
  requestIp?: string;
  optUserName?: string;
  invoiceFee?: {value: string};
  tid?: string;
  bargainPrice?: {value: string};
  serialVersionUID?: number;
  payTypeId?: string;
  leaveMessage?: string;
  consigneeInfo?: ConsigneeInfo;
  optUserCode?: string;
  logisticsFee?: {value: string};
  expectDeliverDate?: Date;
  itemRequestList?: Array<ItemCreateRequest>;

  __fields2java() {
    return {
      $class: 'com.qianmi.yxtc.api.d2pv2.request.D2PTradeModifyRequest',
      $: {
        requestUserCode: java.String(this.requestUserCode),
        attachFile: java.List(
          (this.attachFile || []).map(paramItem => {
            return java.String(paramItem);
          }),
        ),
        chainMasterCode: java.String(this.chainMasterCode),
```

Error > 泛型



```java
package com.ofpay.comm.base;

import ...

public class BaseListBO implements Serializable {

    private static final long serialVersionUID = -7845365502476716789L;
    // 当前页序号
    private int pageno = 0;
    // 每页显示数量
    private int pagesize = 10;
    // 总条数
    private int totalcount = 0;
    // 列表
    private List<?> datalist = null;
    // map
    private Map<?, ?> datamap = null;

    /**
     * pageno
     *
     * @return the pageno
     * @since CodingExample Ver(编码范例查看) 1.0
     */

    public int getPageno() { return pageno; }

    /**
     * @param pageno the pageno to set
     */
    public void setPageno(int pageno) { this.pageno = pageno; }

    /**
     * pagesize
     *
     * @return the pagesize
     * @since CodingExample Ver(编码范例查看) 1.0
     */

    public int getPagesize() { return pagesize; }

```

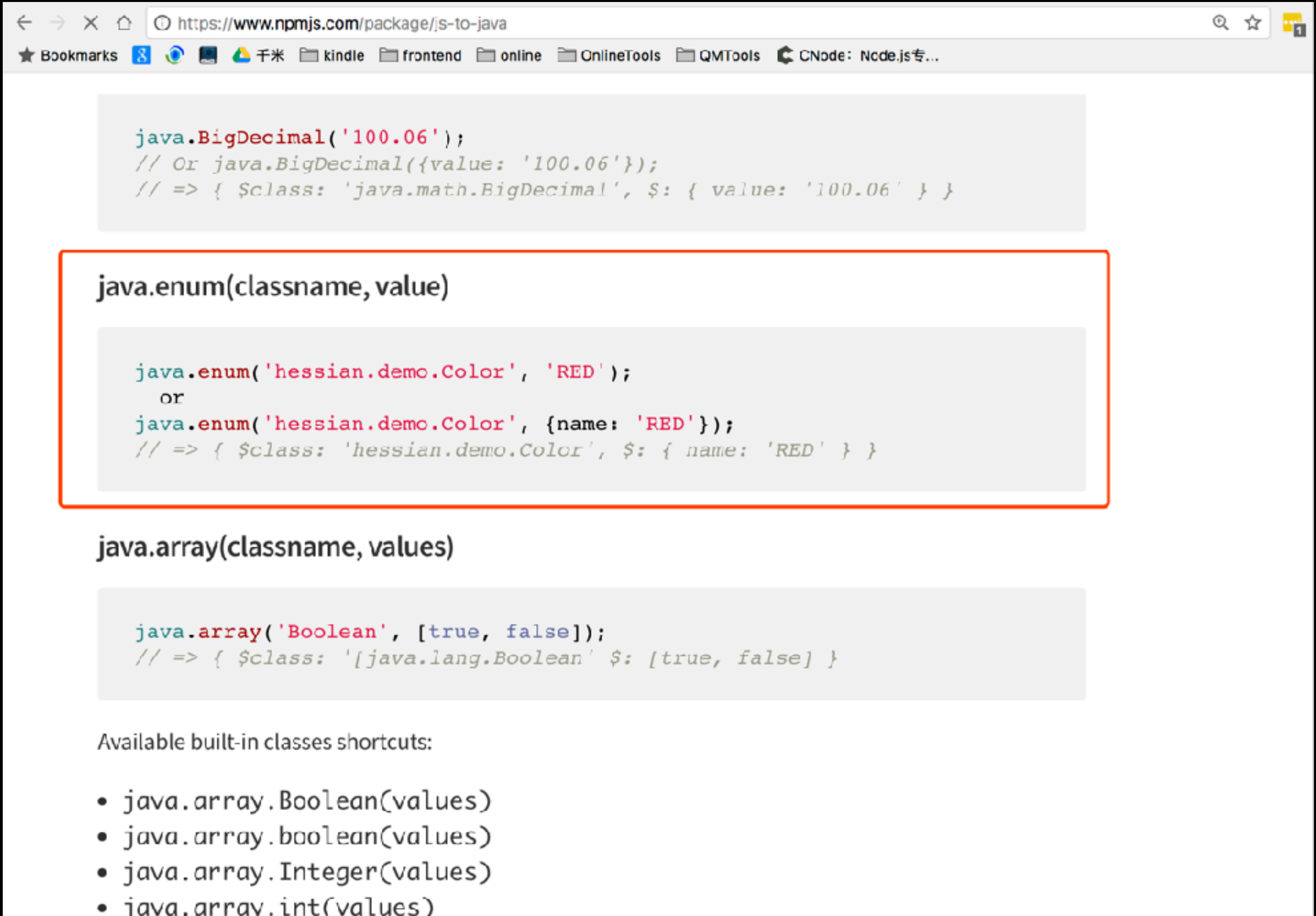
Error > 泛型



```typescript

export interface IBaseListBO {
  totalcount?: number;
  datalist?: Array<any>;
  pageno?: number;
  datamap?: {[name: any]: any};
  pagesize?: number;
}

export class BaseListBO {
  constructor(params: IBaseListBO) {
    this.totalcount = params.totalcount;
    this.datalist = params.datalist;
    this.pageno = params.pageno;
    this.datamap = params.datamap;
    this.pagesize = params.pagesize;
  }

  totalcount?: number;
  datalist?: Array<any>;
  pageno?: number;
  datamap?: {[name: any]: any};
  pagesize?: number;

  __fields2java() {
    return {
      $class: 'com.ofpay.comm.base.BaseListBO',
      $: {
        totalcount: java.Integer(this.totalcount),
        pageno: java.Integer(this.pageno),
        pagesize: java.Integer(this.pagesize),
      },
    };
  }
}

//generate by interpret-cli dubbo2.js
```

Enum类型

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



```
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' } }
```

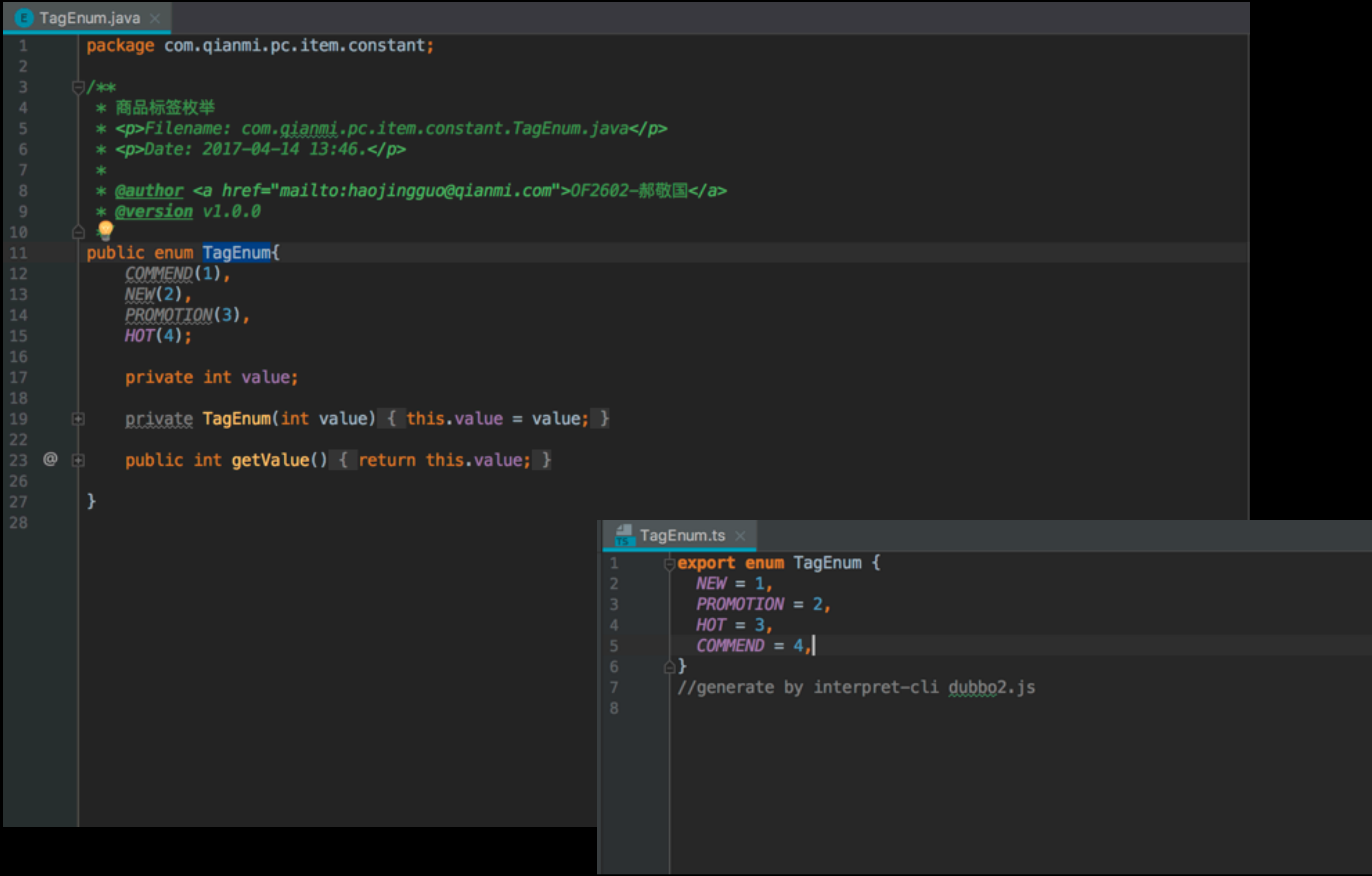
java.array(classname, values)

```
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)`

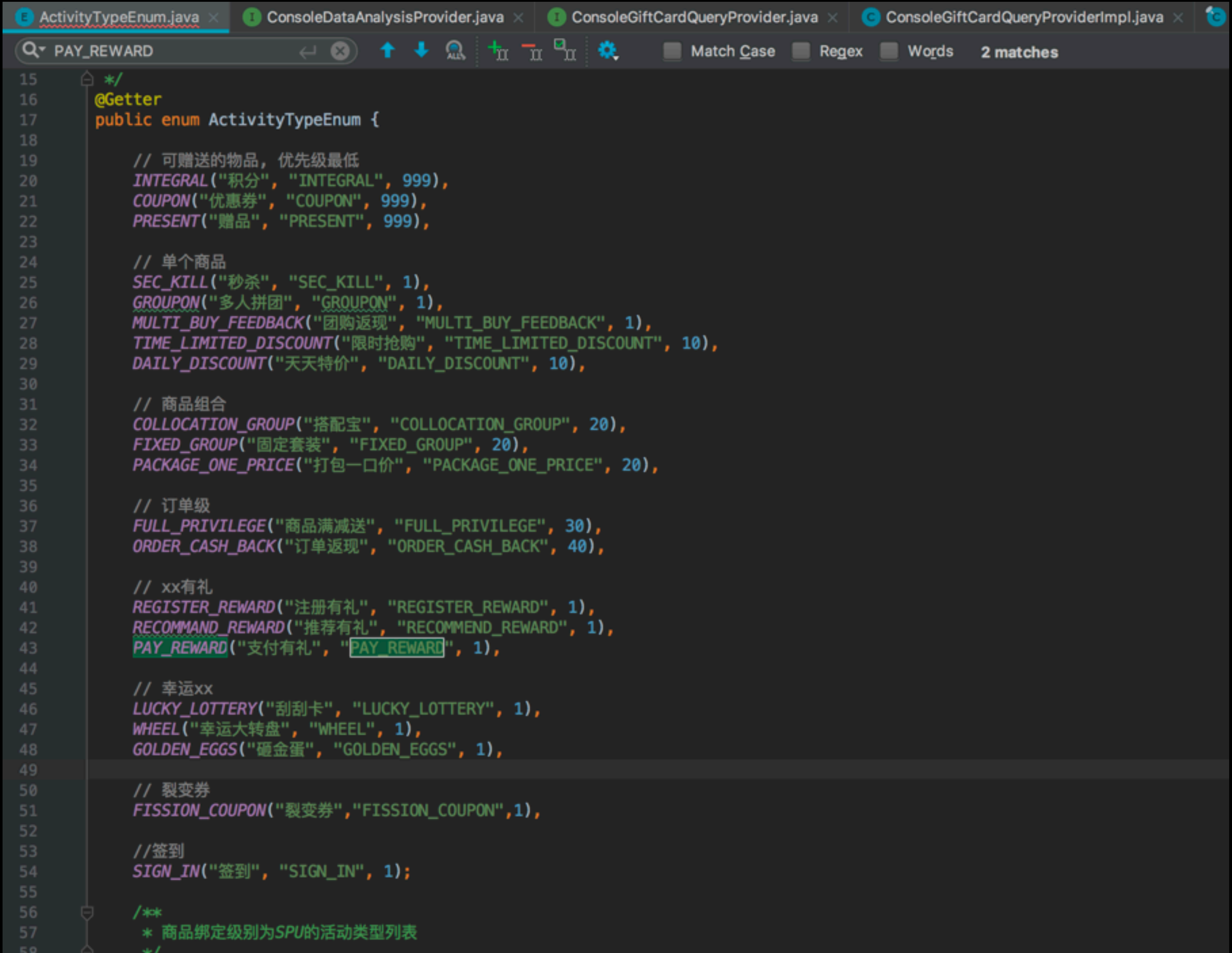
how > 翻译流程 > 转换 > Java to Ts > Enum转换实例



```java
package com.qianmi.pc.item.constant;

/**
 * 商品标签枚举
 * <p>Filename: com.qianmi.pc.item.constant.TagEnum.java</p>
 * <p>Date: 2017-04-14 13:46.</p>
 *
 * @author <a href="mailto:haojingguo@qianmi.com">OF2602-郝敬国</a>
 * @version v1.0.0
 */
public enum TagEnum{
    COMMEND(1),
    NEW(2),
    PROMOTION(3),
    HOT(4);

    private int value;

    private TagEnum(int value) { this.value = value; }

    public int getValue() { return this.value; }

}
```

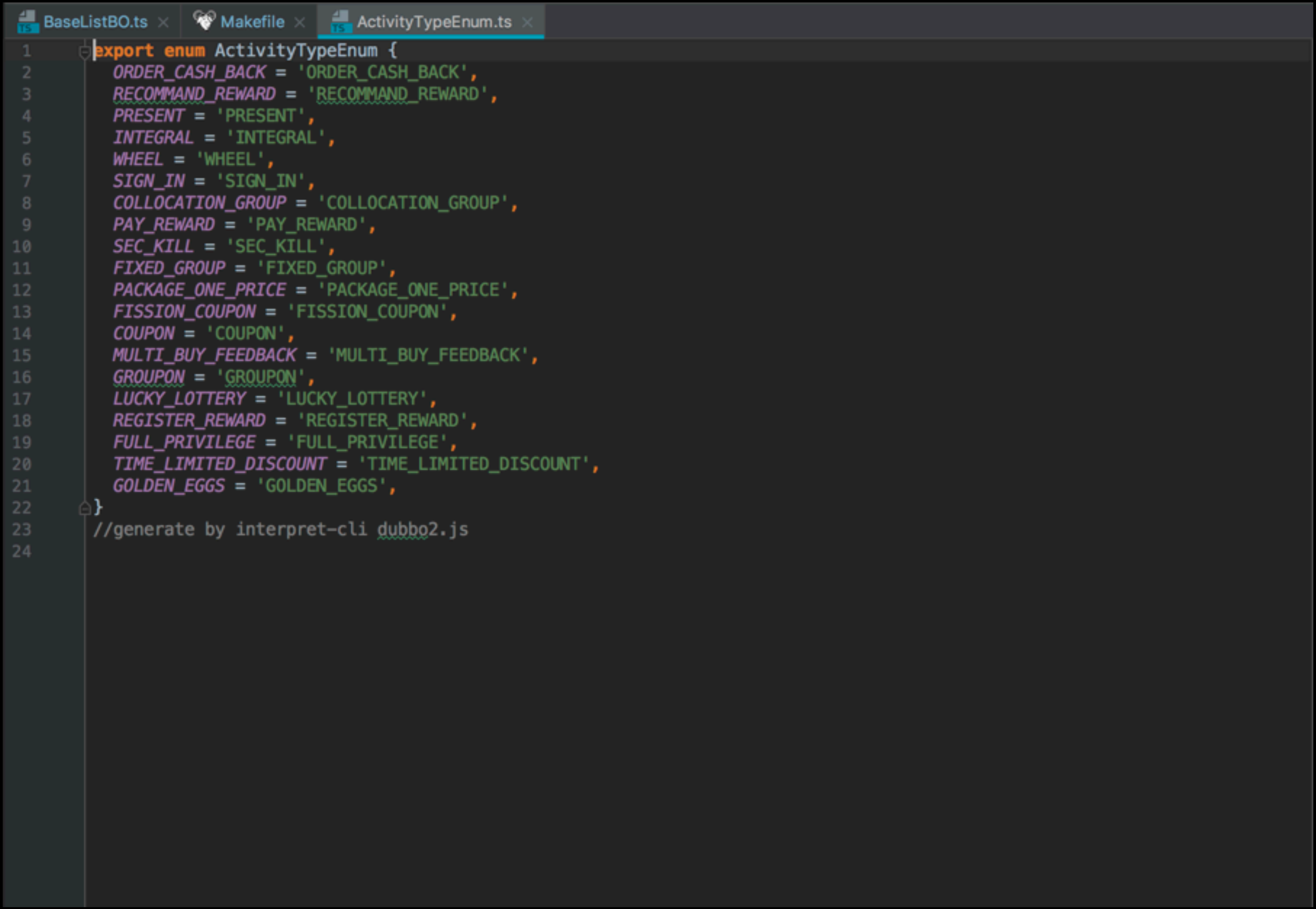
```typescript
export enum TagEnum {
  NEW = 1,
  PROMOTION = 2,
  HOT = 3,
  COMMEND = 4,
}
//generate by interpret-cli dubbo2.js
```

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


ActivityTypeEnum.java
ConsoleDataAnalysisProvider.java
ConsoleGiftCardQueryProvider.java
ConsoleGiftCardQueryProviderImpl.java
PAY_REWARD
Match Case
Regex
Words
2 matches

```java
 */
@Getter
public enum ActivityTypeEnum {

    // 可赠送的物品, 优先级最低
    INTEGRAL("积分", "INTEGRAL", 999),
    COUPON("优惠券", "COUPON", 999),
    PRESENT("赠品", "PRESENT", 999),

    // 单个商品
    SEC_KILL("秒杀", "SEC_KILL", 1),
    GROUPON("多人拼团", "GROUPON", 1),
    MULTI_BUY_FEEDBACK("团购返现", "MULTI_BUY_FEEDBACK", 1),
    TIME_LIMITED_DISCOUNT("限时抢购", "TIME_LIMITED_DISCOUNT", 10),
    DAILY_DISCOUNT("天天特价", "DAILY_DISCOUNT", 10),

    // 商品组合
    COLLOCATION_GROUP("搭配宝", "COLLOCATION_GROUP", 20),
    FIXED_GROUP("固定套装", "FIXED_GROUP", 20),
    PACKAGE_ONE_PRICE("打包一口价", "PACKAGE_ONE_PRICE", 20),

    // 订单级
    FULL_PRIVILEGE("商品满减送", "FULL_PRIVILEGE", 30),
    ORDER_CASH_BACK("订单返现", "ORDER_CASH_BACK", 40),

    // xx有礼
    REGISTER_REWARD("注册有礼", "REGISTER_REWARD", 1),
    RECOMMAND_REWARD("推荐有礼", "RECOMMEND_REWARD", 1),
    PAY_REWARD("支付有礼", "PAY_REWARD", 1),

    // 幸运xx
    LUCKY_LOTTERY("刮刮卡", "LUCKY_LOTTERY", 1),
    WHEEL("幸运大转盘", "WHEEL", 1),
    GOLDEN_EGGS("砸金蛋", "GOLDEN_EGGS", 1),

    // 裂变券
    FISSION_COUPON("裂变券","FISSION_COUPON",1),

    //签到
    SIGN_IN("签到", "SIGN_IN", 1);

    /**
     * 商品绑定级别为SPU的活动类型列表
     */
```

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



```
export enum ActivityTypeEnum {
  ORDER_CASH_BACK = 'ORDER_CASH_BACK',
  RECOMMAND_REWARD = 'RECOMMAND_REWARD',
  PRESENT = 'PRESENT',
  INTEGRAL = 'INTEGRAL',
  WHEEL = 'WHEEL',
  SIGN_IN = 'SIGN_IN',
  COLLOCATION_GROUP = 'COLLOCATION_GROUP',
  PAY_REWARD = 'PAY_REWARD',
  SEC_KILL = 'SEC_KILL',
  FIXED_GROUP = 'FIXED_GROUP',
  PACKAGE_ONE_PRICE = 'PACKAGE_ONE_PRICE',
  FISSION_COUPON = 'FISSION_COUPON',
  COUPON = 'COUPON',
  MULTI_BUY_FEEDBACK = 'MULTI_BUY_FEEDBACK',
  GROUPON = 'GROUPON',
  LUCKY_LOTTERY = 'LUCKY_LOTTERY',
  REGISTER_REWARD = 'REGISTER_REWARD',
  FULL_PRIVILEGE = 'FULL_PRIVILEGE',
  TIME_LIMITED_DISCOUNT = 'TIME_LIMITED_DISCOUNT',
  GOLDEN_EGGS = 'GOLDEN_EGGS',
}
//generate by interpret-cli dubbo2.js
```

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