

久谦 JavaDoc编写方法 软件

软件研究院

概述

编写目的

注释规范

高级应用

导出转换

概述



SINCE 1997



JavaDoc

- 自动生成文档的技术
- 从程序源代码中抽取注释
- Html (CHM) 形式的帮助文档



JDK的JAVADOC是什样子呢?



overview

Java™ Platform Standard Ed. 6
[所有类](#)

软件包
[java.applet](#)
[java.awt](#)
[java.awt.color](#)
[java.awt.datatransfer](#)
[java.awt.dnd](#)

所有类

- [AbstractAction](#)
- [AbstractAnnotationValueVisitor6](#)
- [AbstractBorder](#)
- [AbstractButton](#)
- [AbstractCellEditor](#)
- [AbstractCollection](#)
- [AbstractColorChooserPanel](#)
- [AbstractDocument](#)
- [AbstractDocument.AttributeContext](#)
- [AbstractDocument.Content](#)
- [AbstractDocument.ElementEdit](#)
- [AbstractElementVisitor6](#)
- [AbstractExecutorService](#)
- [AbstractInterruptibleChannel](#)
- [AbstractLayoutCache](#)
- [AbstractLayoutCache.NodeDimensions](#)
- [AbstractList](#)
- [AbstractListModel](#)
- [AbstractMap](#)
- [AbstractMap.SimpleEntry](#)
- [AbstractMap.SimpleImmutableEntry](#)
- [AbstractMarshallerImp](#)
- [AbstractMethodError](#)
- [AbstractOwnableSynchronizer](#)
- [AbstractPreferences](#)
- [AbstractProcessor](#)
- [AbstractQueue](#)
- [AbstractQueueImpl.Synchronizer](#)

概述 软件包 类 使用 树 已过时 索引 帮助

上一个 下一个 [框架](#) [无框架](#)

Java™ Platform Standard Edition 6 API 规范

本文档是 Java™ Platform Standard Edition 6 的 API 规范。

请参见: [描述](#)

| Java SE Platform 软件包 | |
|---|---|
| java.applet | 提供创建 applet 所必需的类和 applet 用来与其 applet 上下文通信的类。 |
| java.awt | 包含用于创建用户界面和绘制图形图像的所有类。 |
| java.awt.color | 提供用于颜色空间的类。 |
| java.awt.datatransfer | 提供在应用程序之间和在应用程序内部传输数据的接口和类。 |
| java.awt.dnd | Drag 和 Drop 是一种直接操作动作，在许多图形用户界面系统中都会遇到它，它提供了一种机制，能够在两个与 GUI 中显示元素逻辑相关的实体之间传输信息。 |
| java.awt.event | 提供处理由 AWT 组件所激发的各类事件的接口和类。 |
| java.awt.font | 提供与字体相关的类和接口。 |
| java.awt.geom | 提供用于在与二维几何形状相关的对象上定义和执行操作的 Java 2D 类。 |
| java.awt.im | 提供输入方法框架所需的类和接口。 |
| java.awt.im.spi | 提供启用可以与 Java 运行时环境一起使用的输入方法开发的接口。 |
| java.awt.image | 提供创建和修改图像的各种类。 |
| java.awt.image.renderable | 提供用于生成与呈现无关的图像的类和接口。 |
| java.awt.print | 为通用的打印 API 提供类和接口。 |
| java.beans | 包含与开发 beans 有关的类，即基于 JavaBeans™ 架构的组件。 |
| java.beans.beancontext | 提供与 bean 上下文有关的类和接口。 |

package

Java™ Platform
Standard Ed. 6

[All Classes](#)

Packages

- [java.applet](#)
- [java.awt](#)
- [java.awt.color](#)
- [java.awt.datatransfer](#)
- [java.awt.dnd](#)
- [java.awt.event](#)

[java.awt](#)

Interfaces

- [ActiveEvent](#)
- [Adjustable](#)
- [Composite](#)
- [CompositeContext](#)
- [ItemSelectable](#)
- [KeyEventDispatcher](#)
- [KeyEventPostProcessor](#)
- [LayoutManager](#)
- [LayoutManager2](#)
- [MenuContainer](#)
- [Paint](#)
- [PaintContext](#)
- [PrintGraphics](#)
- [Shape](#)
- [Stroke](#)
- [Transparency](#)

Classes

- [AlphaComposite](#)
- [AWTEvent](#)
- [AWTEventMulticaster](#)
- [AWTKeyStroke](#)
- [AWTPermission](#)
- [BasicStroke](#)
- [BorderLayout](#)
- [BufferCapabilities](#)
- [BufferCapabilities.FlipContents](#)

Overview **Package** Class Use Tree Deprecated Index Help
Java™ Platform
Standard Ed. 6

[PREV PACKAGE](#) [NEXT PACKAGE](#)
[FRAMES](#) [NO FRAMES](#)

Package java.awt

Contains all of the classes for creating user interfaces and for painting graphics and images.

See:

[Description](#)

Interface Summary

| | |
|---------------------------------------|--|
| ActiveEvent | An interface for events that know how to dispatch themselves. |
| Adjustable | The interface for objects which have an adjustable numeric value contained within a bounded range of values. |
| Composite | The Composite interface, along with CompositeContext , defines the methods to compose a draw primitive with the underlying graphics area. |
| CompositeContext | The CompositeContext interface defines the encapsulated and optimized environment for a compositing operation. |
| ItemSelectable | The interface for objects which contain a set of items for which zero or more can be selected. |
| KeyEventDispatcher | A KeyEventDispatcher cooperates with the current KeyboardFocusManager in the targeting and dispatching of all KeyEvents. |
| KeyEventPostProcessor | A KeyEventPostProcessor cooperates with the current KeyboardFocusManager in the final resolution of all unconsumed KeyEvents. |
| LayoutManager | Defines the interface for classes that know how to lay out Containers. |
| LayoutManager2 | Defines an interface for classes that know how to layout Containers based on a layout constraints object. |
| MenuContainer | The super class of all menu related containers. |
| Paint | This Paint interface defines how color patterns can be generated for Graphics2D operations. |
| PaintContext | The PaintContext interface defines the encapsulated and optimized environment to generate color patterns in device space for fill or stroke operations on a Graphics2D . |
| PrintGraphics | An abstract class which provides a print graphics context for a page. |
| Shape | The Shape interface provides definitions for objects that represent some form of geometric shape. |

完成
切换浏览模式 IE打开
100%



class

概述 软件包 **类** 使用 树 已过时 索引 帮助

Java™ Platform
Standard Ed. 6

[上一个类](#) [下一个类](#)

[框架](#) [无框架](#)

摘要: [类](#) | [字段](#) | [构造方法](#) | [方法](#)

详细信息: [字段](#) | [构造方法](#) | [方法](#)

java.lang

类 Boolean

[java.lang.Object](#)

└ [java.lang.Boolean](#)

所有已实现的接口:

[Serializable](#), [Comparable<Boolean>](#)

```
public final class Boolean
extends Object
implements Serializable, Comparable<Boolean>
```

Boolean 类将基本类型为 boolean 的值包装在一个对象中。一个 Boolean 类型的对象只包含一个类型为 boolean 的字段。

此外，此类还为 boolean 和 String 的相互转换提供了许多方法，并提供了处理 boolean 时非常有用的其他一些常量和方法。

从以下版本开始:

JDK1.0

另请参见:

[序列化表格](#)

字段摘要

| | | |
|--|-----------------------|----------------------------|
| <code>static Boolean</code> | FALSE | 对应基值 false 的 Boolean 对象。 |
| <code>static Boolean</code> | TRUE | 对应基值 true 的 Boolean 对象。 |
| <code>static Class<Boolean></code> | TYPE | 表示基本类型 boolean 的 Class 对象。 |



JAVADOC能给我们带来...

编写目的



SINCE 1997

目的

浏览查找

版本控制

变更管理



我们
一定要写好
JAVADOC

!!!

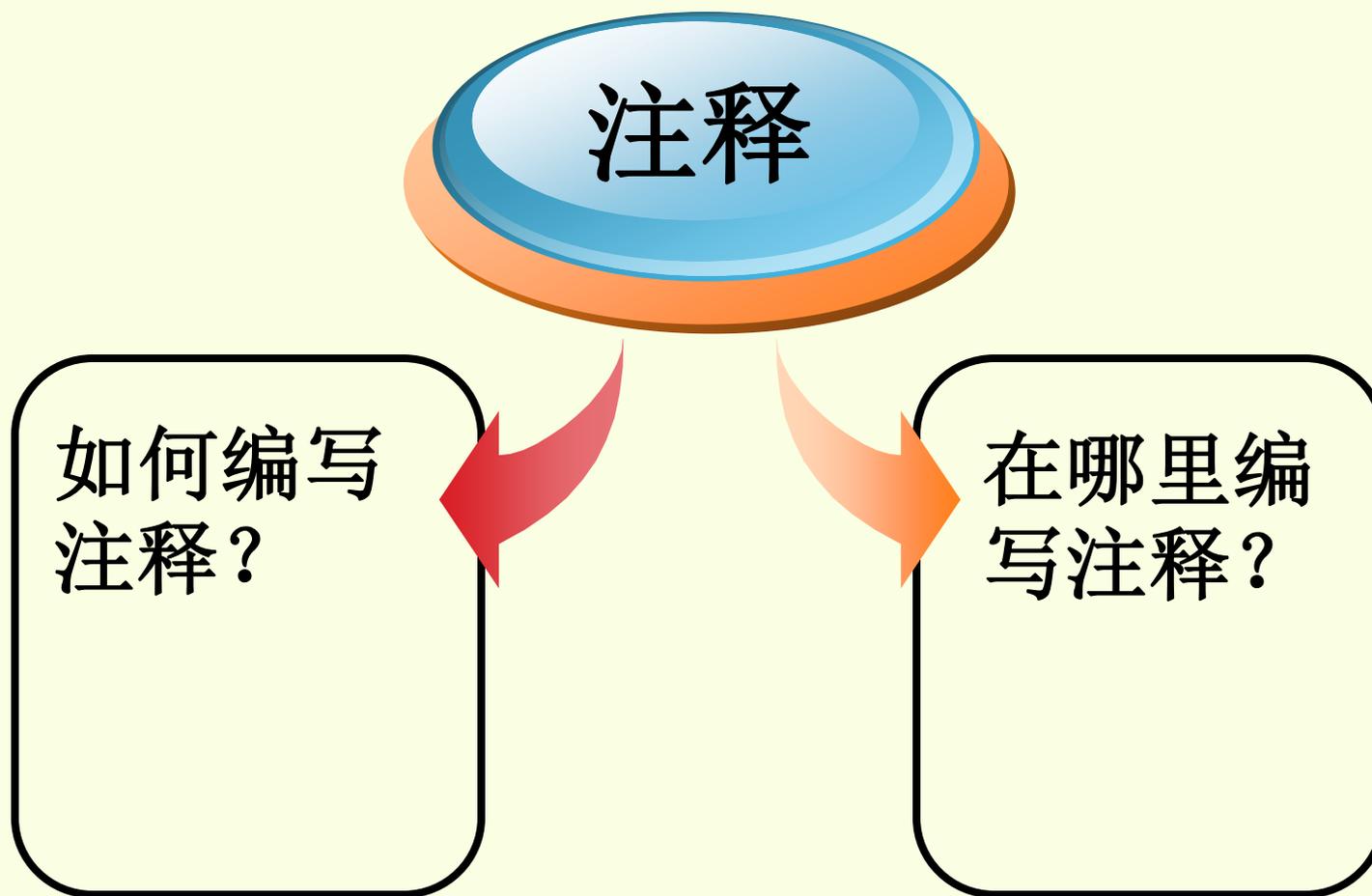


我不会写
JAVADOC
.....

注释规范



SINCE 1997



注释格式

□行注释

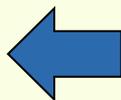
```
// //////////////////////////////////////用户及登陆相关////////////////////////////////////
```

□块注释

```
/*
 * 块注释
 */
```

□文档注释

Java Doc

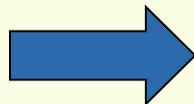


```
/**
 * 检查是否对某资源拥有某类权限<br>
 * 请不要使用该方法用作为过滤有权限的资源项, <br>
 * 请使用getXXX(Operation<? super TFacade> operation,...)<br>
 * 或findXXX(Operation<? super TFacade> operation,...)<br>
 *
 * @param operation
 *         操作
 * @param resource
 *         资源
 * @return 返回是权限信息
 */
public <TFacade> Authority getAuthority(
    Operation<? super TFacade> operation, ResourceStub<TFacade> resource);
```

注释细则——特殊字符

“/”、“*” 特殊字符的使用

```
/*  
 * 获得服务的标题  
 */  
@Override  
public final String getTitle() {  
    return super.getTitle();  
}
```



不会出现*****字符

```
/** *****  
 * 获得服务的标题  
 */  
@Override  
public final String getTitle() {  
    return super.getTitle();  
}
```



会出现*****字符

注释细则——摘要句

每个文档注释的第一句，简明而准确地描述**API**项，即摘要句；摘要句以最近的“.”空格”或其它注释标签结束。

```
/**
 * 获取序列化后的数据，继承类也可以重写该方法，实现自己的序列化
 * @return String 返回序列化后的字符串
 */
public String getData();

/**
 * 从字符串中反序列化属性.<br>
 * 继承类也可以重写该方法，实现自己的反序列化
 * @param data String 序列化的字符串
 */
public void setData(String data);
```



| 方法摘要 | |
|------------------|---|
| java.lang.String | getData() 获取序列化后的数据，继承类也可以重写该方法，实现自己的序列化 |
| void | setData (java.lang.String data) 从字符串中反序列化属性. |

标签（一）

| 名称 | 描述 |
|------------|---|
| @author | @author name-text 指定生成文档中的作者项，用于修饰类或接口，声明类型的作者，name-text可以指定多个名字（使用,隔开）。 |
| @version | @version version-text 声明类型版本信息，用于修饰类或接口。 |
| @param | @param parameter-name description 描述方法的参数，parameter-name为参数名称，不是参数类型，description为参数描述文本，一个@param标签描述一个参数，多个参数时，以多个标签分多行描述。 |
| @return | @return description 描述方法返回值，除void类型和构造方法外，所有的方法都应描述返回值。对于可能返回空值的方法，也应说明什么场景下返回空值。 |
| @exception | @exception class-name description 描述方法抛出的异常，class-name抛出异常的类型，description抛出异常的原因、场景等。 |
| @throws | @throws class-name description 与@exception同义。 |

 标签（二）

| 名称 | 描述 |
|-------------|---|
| @see | <p>@see reference 添加参见标题，其中有指向reference的链接或者文本项。</p> <ul style="list-style-type: none">(1) @see "string" 为"string"添加文本项，不产生链接；(2) @see Label 使用HTML标记产生链接；(3) @see package.class#member member方法名或属性名，与@link类似。 |
| {@link} | <p>{@link package.class#member label} 插入指向package.class#member的内嵌链接，package.class是引用的包名和类名，member是方法或属性名称，label是链接的显示标题。</p> |
| @since | <p>@since since-text 标明该方法/类/接口可以运行的JDK最低版本。</p> |
| @deprecated | <p>@deprecated deprecated-text 添加注释，表明不推荐使用该API，可以修饰类、接口、属性和方法。通常应说明废弃原因，是否有替代的使用方法。</p> |

注释结构——文件头注释

用于文件的头部，java源码中包声明之前，描述该文件的修订历史、版本、版权信息等。

```
/**=====
 * 版权：久其软件 版权所有 (c) 2010 - 2020
 * 包：com.jiuqi.dna.bap.query.define.intf.facade
 * 修改记录：
 * 日期      作者      内容
 * =====
 * 2010-7-19      wangg
 * =====*/

package com.jiuqi.dna.bap.query.define.intf.facade;

import com.jiuqi.dna.core.type.GUID;
```

注释结构——类、接口注释

在类或接口的声明之前，
用于概述类的功能、实现等描述信息、以及作者、版本等。

```
package com.jiuqi.dna.bap.query.define.intf.facade;

import com.jiuqi.dna.core.type.GUID;

/**
 * <p>查询结果工具栏按钮动作外观.</p>
 *
 * <p>Copyright: 版权所有 (c) 2010 - 2020<br>
 * Company: 久其</p>
 *
 * @author nancy.wang
 * @version 2010-7-19
 */

public interface FAction{
```

注释结构——变量、属性注释

描述变量的意义、取值含义和默认值等。

```
/**  
 * 基础数据选择控件系统选项默认值：显示常用选项的个数，默认为6.<br>  
 */  
public static final int SYSOPTION_SELECT_SHOWUSUALCOUNT_VALUE = 6;
```

注释结构——方法注释

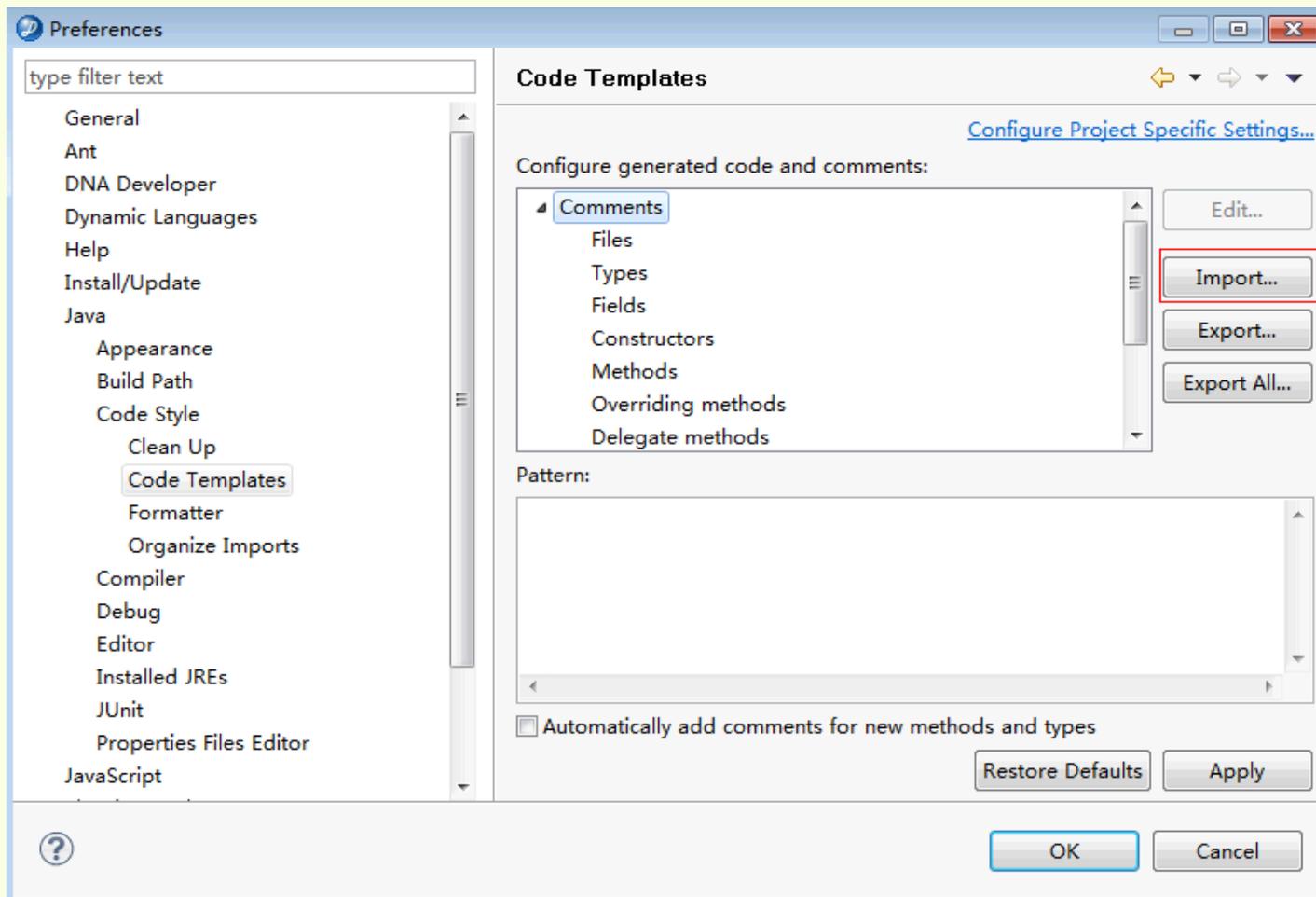
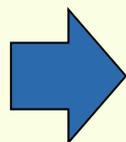
标明每个方法的输入、输出参数、异常信息和返回值类型，说明特殊变量取值的含义。

```
/**
 * 更新基础数据定义.<br>
 *
 * @param context 上下文
 * @param baseData 基础数据定义
 * @throws BaseDataException 根据id查找基础数据定义用于更新.<br>
 */
public static BaseData updateBaseData(BaseData baseData, Context context) throws BaseDataException {
}

/**
 * 获得查询执行界面情景接口.<br>
 *
 * @return Situation
 * @see #getContext()
 * @deprecated 在非界面线程中构建查询执行环境时返回值为空，使用getContext替代
 */
public Situation getSituation ();
```




注释模版



可以在
JAVADOC中
加入漂亮的图片么？



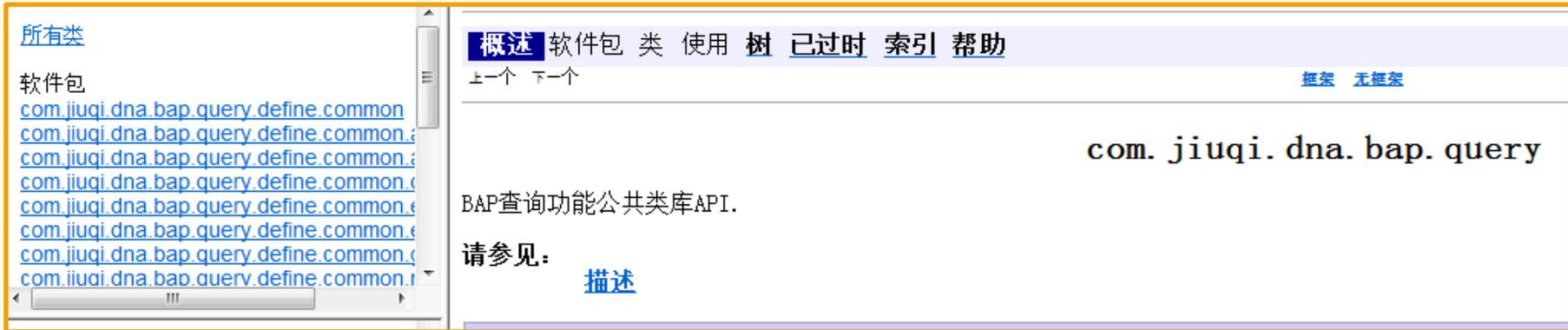
高级应用



SINCE 1997

描述文件_整体的描述Overview

- ❑ 必须为HTML文件；
- ❑ 与文档注释的写法一样，且支持标签；
- ❑ 无位置要求；
- ❑ 导出时需要选择该文件，才能引入；



描述文件_包的描述

- ❑ 必须为HTML文件；
- ❑ 与文档注释的写法一样，且支持标签；
- ❑ 有位置要求，在对应的包下；
- ❑ 文件名必须为package.html；

[com.jiuqi.dna.bap.query.define.intf.entity](#)
[com.jiuqi.dna.bap.query.define.intf.facade](#)
[com.jiuqi.dna.bap.query.define.intf.impl](#)
[com.jiuqi.dna.bap.query.define.intf.task](#)

[com.jiuqi.dna.bap.query.define.intf.facade](#)

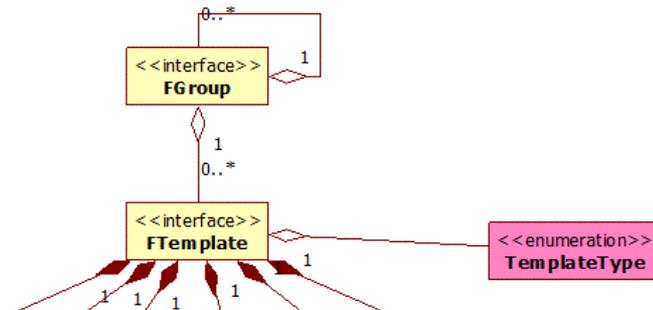
接口

[FAction](#)
[FChartSetting](#)
[FDataSourceSetting](#)
[FDynamicScriptRow](#)
[FField](#)
[FFieldSetting](#)
[FGroup](#)
[FLabel](#)
[FLabelSetting](#)
[FOrderByItem](#)
[FParameter](#)
[FParaSetting](#)
[FQueryProperty](#)
[FRelateQuery](#)
[FRelateQuerySetting](#)
[FResultSetting](#)

软件包 com.jiuqi.dna.bap.query.define.intf.facade 的描述

查询模板定义元数据外观类包。

包括查询模板定义元数据的各层次外观结构，模板定义元数据结构参见类图：



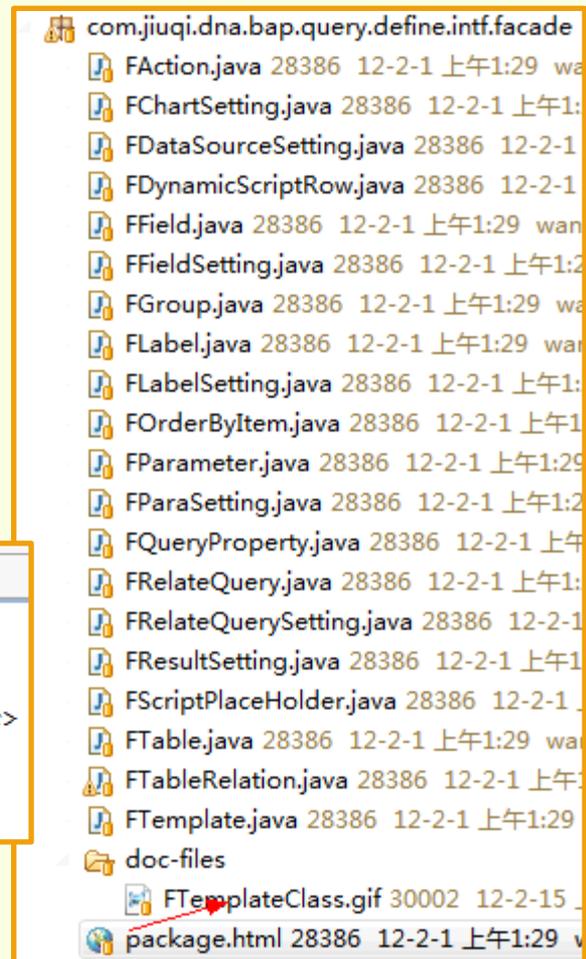


文件引用

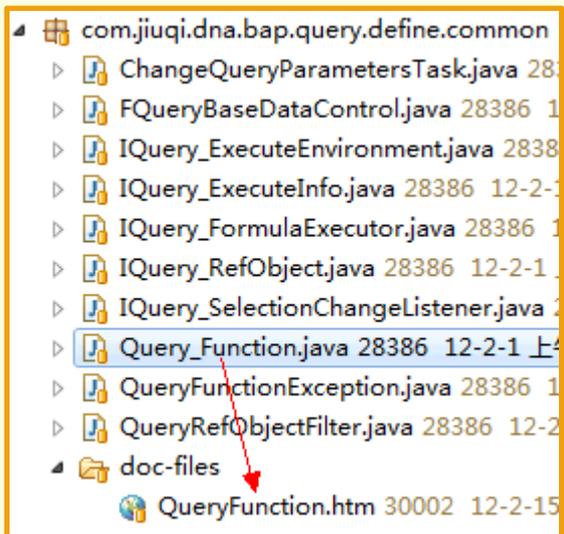
- 注释可以使用标准HTML的所有语法；
- 如果注释中需要引用其它文件，如图片、源码、静态页面等，文件必须放在引用源对象所在包下的doc-files文件夹中；

引用图片

`` ;



引用HTML静态页



```
/**
 * <p>查询扩展函数接口.</p>
 *
 *
 * 扩展动作继承本实现类，实现抽象方法，并在dna.xml中注册，注册格式如下：
 * <pre>
 *   <query-functions>
 *     <function class="....." />
 *   </query-functions>
 * </pre>
 *
 * 函数中校验方法“judgeResultType”内如需提示校验语法错误信息，可以抛出扩展
 *
 * <p>Copyright: 版权所有 (c) 2010 - 2020<br>
 * Company: 久其</p>
 *
 * 参考 <a href = doc-files/QueryFunction.htm> 示例代码</a>
 */
```

导出转换



SINCE 1997



JavaDoc导出

- 选
- 选

Generate Javadoc

Javadoc Generation
Configure Javadoc arguments.

Overview:

VM options (prefixed with '-J', e.g. -J-Xmx180m for larger heap space):

Extra Javadoc options (path names with white spaces must be enclosed in quotes):

JRE source compatibility: 1.5

Save the settings of this Javadoc export as an Ant script:
Ant Script:

Open generated index file in browser



目录(C) | 索引(I) | 搜索(S) | 收藏夹(I)

About Jd2chm...
[Overview](#)
 Hierarchy For All Packages
 com.jiuqi.dna.bap.query.define.common

[概述](#) [软件包](#) [类](#) [使用](#) [树](#) [已过时](#) [索引](#) [帮助](#)

[上一个](#) [下一个](#)

[框架](#) [无框架](#) [所有类](#)

目录(C) | 索引(I) | 搜索(S) | 收藏夹(I)

键入关键字进行查找(F):

F

- FAction**
- FChartSetting
- FChartSetting.Deriction
- FDataSourceSetting
- FDynamicScriptRow
- FField
- FFieldSetting
- FGroup
- FIELD_PATH_SEPERATOR
- FieldDataType
- FieldDataTypeDetector
- FieldDataTypeDetector()
- FieldImpl
- FieldImpl()
- FieldImpl(FField)
- FieldSettingImpl
- FieldSettingImpl()
- FieldSettingImpl(FFieldSet
- fileNameRegExp
- FLabel

目录(C) | 索引(I) | 搜索(S) | 收藏夹(I)

键入要搜索的单词(W):

FAction

列出主题(T)

显示(O)

选择主题(T):

找到: 17

| 标题 | 位置 | 名次 |
|-----------------------------------|---------|----|
| ActionImpl | TestCHM | 1 |
| ½Ó¿Ú com.jiuqi.dna.bap.query... | TestCHM | 2 |
| G - È=Òý | TestCHM | 3 |
| FResultSetting (Interface) | TestCHM | 4 |
| QueryTemplateHelper | TestCHM | 5 |
| FAction (Interface) | TestCHM | 6 |
| Uses of Package com.jiuqi.dn... | TestCHM | 7 |
| Hierarchy For All Packages | TestCHM | 8 |
| L - È=Òý | TestCHM | 9 |
| ÈúÓÐÀà | TestCHM | 10 |
| ÈúÓÐÀà | TestCHM | 11 |
| com.jiuqi.dna.bap.query.define... | TestCHM | 12 |
| com.jiuqi.dna.bap.query.define... | TestCHM | 13 |
| Hierarchy For Package com.jiu... | TestCHM | 14 |
| Hierarchy For Package com.jiu... | TestCHM | 15 |
| 1 - È=Òý | TestCHM | 16 |

ina.bap.query

[概述](#) [软件包](#) [类](#) [使用](#) [树](#) [已过时](#) [索引](#) [帮助](#)

[上一个类](#) [下一个类](#)

摘要: [嵌套](#) | [字段](#) | [构造方法](#) | [方法](#)

com.jiuqi.dna.bap.query.define.intf.facade

接口 **FAction**

所有已知实现类:

[ActionImpl](#)

```
public interface FAction
```

查询结果工具栏按钮动作外观.

Copyright: 版权所有 (c) 2010 - 2020

Company: 久其

构建平台

□ build.properties文件控制

```
build.properties  FQueryBaseDataContro  Query_Functi  qi.dna.bap
```

```
source.. = src/
output.. = classes/
bin.includes = META-INF/, \
              .., \
              dna.xml, \
              src/

javadoc.fileset.1.dir = src
javadoc.fileset.1.includes = com/jiuqi/dna/bap/query/define/intf/task/GroupDeleteTask.java, \
                             com/jiuqi/dna/bap/query/define/intf/task/TemplateDeleteTask.java, \
                             com/jiuqi/dna/bap/query/define/intf/task/TemplateModifyTask.java

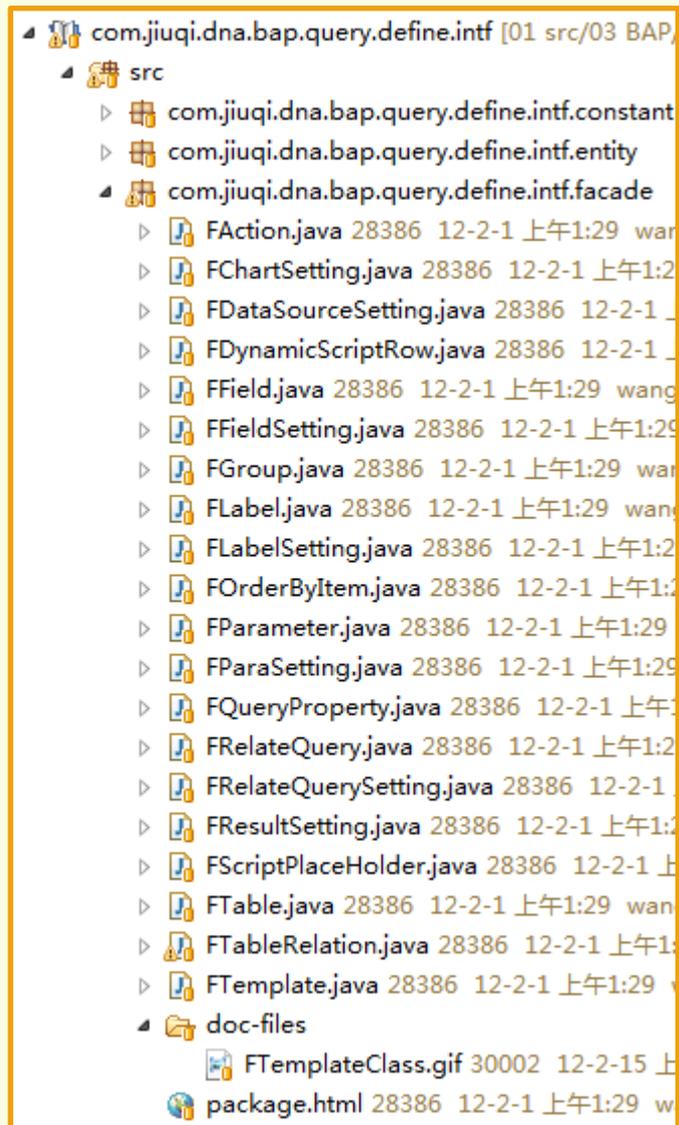
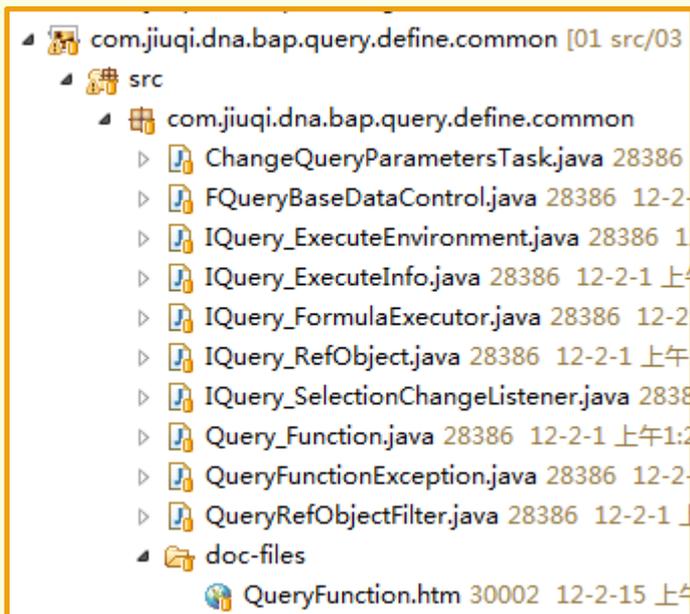
javadoc.fileset.2.dir = src
javadoc.fileset.2.includes = com/jiuqi/dna/bap/query/define/intf/util/JoinTypeHelper.java, \
                             com/jiuqi/dna/bap/query/define/intf/util/QueryTemplateHelper.java

javadoc.packageSet.3.dir = src
javadoc.packageSet.3.includes = com/jiuqi/dna/bap/query/define/intf/facade/**
```



参考示例

- ❑ com.jiuqi.dna.bap.query.define.common;
- ❑ com.jiuqi.dna.bap.query.define.intf;



专题交流回顾



JavaDoc与文档注释

注释的编写格式

注释的标签应用

注释的高级应用

HTML、CHM形式API

资料：[\\file01\培训资料\13 软件研究院\21 技术交流\01 javadoc](#)



完善JAVADOC去喽！！

久其软件
钻石品质

诚信协作
超越
奉献

谢谢

