

# Programmation orientée objet en langage JAVA

## Présentation de la documentation JAVA

Claude Duvallet

Université du Havre  
UFR Sciences et Techniques  
25 rue Philippe Lebon - BP 540  
76058 LE HAVRE CEDEX  
Claude.Duvallet@gmail.com  
<http://litis.univ-lehavre.fr/~duvallet/>



# Présentation de la JavaDOC

- 1 Introduction
- 2 Présentation générale
- 3 Description des classes
- 4 Utilisation de la JavaDOC

## Introduction à la JavaDOC

- Documentation indispensable pour programmer en Java.
- Description complète de toute l'API Java :
  - description des packages ;
  - description des classes ;
  - description des méthodes ;
  - description des constantes et des attributs ;
- Disponible au format HTML.
- Disponible en Anglais, en Japonais et en Chinois.
- Possibilité de l'utiliser en ligne sur Internet ou encore de la télécharger sur votre poste de travail.
- Adresse : `http://java.sun.com/javase/6/docs/`.

# Écran général de la JavaDOC

File Edit View History Bookmarks Tools Help

http://localhost/docs/api/

Overview Package Class Use **Tree** Deprecated Index Help

PREV NEXT FRAMES NO FRAMES

## Java™ Platform, Standard Edition 6 API Specification

This document is the API specification for version 6 of the Java™ Platform, Standard Edition 6.

See: [Description](#)

Packages	
<a href="#">java.applet</a>	Provides the classes necessary to create an applet and the classes an applet uses to communicate with its applet context.
<a href="#">java.awt</a>	Contains all of the classes for creating user interfaces and for painting graphics and images.
<a href="#">java.awt.color</a>	Provides classes for color spaces.
	Provides interfaces and classes for

## Trois zones principales

- Une zone pour l'ensemble des packages disponibles.
- Une zone pour les interfaces et les classes disponibles au sein des packages.
  - cette zone comporte par défaut toutes les classes et interfaces disponibles ;
  - elle est dépendante du package sélectionné.
- Une zone de description :
  - cette zone comporte par défaut la description de tous les packages ;
  - elle est dépendante de la classes qui a été sélectionnée ;
  - elle sert à présenté la description complète des classes.

## La zone de description d'une classe

java.net

### Class CacheResponse

[java.lang.Object](#)

└ [java.net.CacheResponse](#)

Direct Known Subclasses:

[SecureCacheResponse](#)

public abstract class **CacheResponse**  
extends [Object](#)

Represent channels for retrieving resources from the ResponseCache. Instances of such a class provide an `InputStream` also a `getHeaders()` method which returns the associated response headers.

Since:

1.5

### Constructor Summary

[CacheResponse\(\)](#)

### Method Summary

abstract <a href="#">InputStream</a>	<a href="#">getBody()</a> Returns the response body as an <code>InputStream</code> .
abstract <a href="#">Map&lt;String, List&lt;String&gt;&gt;</a>	<a href="#">getHeaders()</a> Returns the response headers as a <code>Map</code> .

## Description générale de la classe

- Texte descriptif de la classe.
- Arborescence de la classe (héritage).
  - référence au package d'appartenance ;
  - par défaut toutes les classes héritent de `java.lang.Object`.
- Liste des constructeurs, des méthodes et des attributs :
  - liste des paramètres pour les méthodes (ou les constructeur) avec leur type et le type de retour de la méthode ;
  - type de chaque attribut ou des constantes.

## Le détail d'une classe

### Constructor Detail

#### CacheResponse

```
public CacheResponse()
```

### Method Detail

#### getHeaders

```
public abstract Map<String,List<String>> getHeaders()  
                                     throws IOException
```

Returns the response headers as a Map.

**Returns:**

An immutable Map from response header field names to lists of field values. The status line has null as its

**Throws:**

[IOException](#) - if an I/O error occurs while getting the response headers

#### getBody

```
public abstract InputStream getBody()  
                             throws IOException
```

Returns the response body as an InputStream.

**Returns:**

an InputStream from which the response body can be accessed



## Description détaillée de la classe

- Description détaillée des constructeurs et des méthodes.
  - objectifs de la méthode ;
  - retour de la méthode ;
  - détails des paramètres avec leur type ;
  - exception soulevée, type et description ;
  - ancienneté de la méthode : depuis quelle version elle est présente ;
  - lien avec d'autres méthodes ou d'autres classe.

## Utilisation de la JavaDOC

- Pourquoi utiliser la JavaDOC ?
  - rechercher le nom exact d'une classe répondant à un besoin ;
  - recherche la syntaxe exacte d'une méthode.
- Comment utiliser la JavaDOC ?
  - évaluer ce que l'on connaît de notre besoin ;
  - trouver la classe correspondante à notre besoin ;
  - trouver la méthode correspondante ;
  - ne pas hésiter à consulter les méthodes obtenues par héritage et à remonter la recherche dans la hiérarchie de classes ;
  - ne pas hésiter à suivre les liens "See Also".

⇒ **JavaDOC = outil indispensable pour un développeur JAVA.**