

Comment utiliser la Javadoc ?

par Janitrix

Date de publication : 15/10/2007

Dernière mise à jour :

La documentation officielle Java, couramment appelée Javadoc, est un outil très puissant mais souvent inconnu des développeurs. Une fois que vous aurez dompté la Javadoc, Java n'aura plus de secret pour vous !

- I - Introduction
- II - Présentation générale
- III - Méthode de recherche
- IV - Localisation des informations importantes
- V - La recherche simplifiée par JavaSearch
- VI - Conclusion
- VII - Remerciements

I - Introduction

Le langage Java dispose d'une API (Application Program Interface) très riche, avec pas moins de 203 packages, 3777 classes, 28306 méthodes, dans la version Java SE 6. Beaucoup de ressources donc, et il est très facile de s'y perdre. Tout programmeur Java qui débute se demandera un jour comment tronquer une chaîne de caractères, changer la couleur d'un texte, agrandir une fenêtre, bref, apprendre à réaliser des opérations de base.

C'est là qu'intervient la JavaDoc, énorme documentation qui regroupe toutes les classes et méthodes de l'édition standard de Java (il en existe pour les autres versions de Java, mais cela fonctionne de la même façon). Une fois que vous aurez appris à faire une recherche dans cette documentation, la seule barrière qui demeurera sera votre imagination.

Enfin, sachez que toute cette documentation est en anglais.

II - Présentation générale

La Javadoc de la dernière version de Java SE est disponible à cette adresse : <http://java.sun.com/javase/6/docs/api/>. La fenêtre en haut à gauche, marquée A, liste tous les packages disponibles. La fenêtre de gauche marquée C liste toutes les classes du package que vous avez sélectionné. Par défaut, elle contient toutes les classes (c'est donc une liste très longue).



La page d'accueil de la Javadoc

Enfin, la fenêtre centrale marquée B liste par défaut tous les packages avec une description pour chacun (encore une très longue liste). Dès que vous sélectionnez une classe ou une interface, cette fenêtre contient tout ce qui concerne cette classe : description, méthodes, variables.

III - Méthode de recherche

Avant de faire quoi que ce soit, précisons tout de suite quelle sera la démarche pour rechercher une méthode ou une classe. Imaginons que vous souhaitez couper une chaîne de caractères. Vous n'avez à ce moment aucune idée de la démarche à suivre. Plutôt que de vous précipiter sur un forum afin de demander une réponse toute prête, prenez une minute afin de réfléchir.

Premièrement, demandez-vous si ce que vous voulez réaliser est basique ou non. Tronquer une chaîne de caractères est très courant, et nous pouvons donc penser que cette méthode est dans l'API de base.

Ensuite, localisez les classes qui ont le plus de chances de contenir la méthode recherchée. Notre méthode concerne les chaînes de caractères donc elle a de fortes chances de se trouver dans la classe String, classe de base pour utiliser les chaînes de caractères en Java.

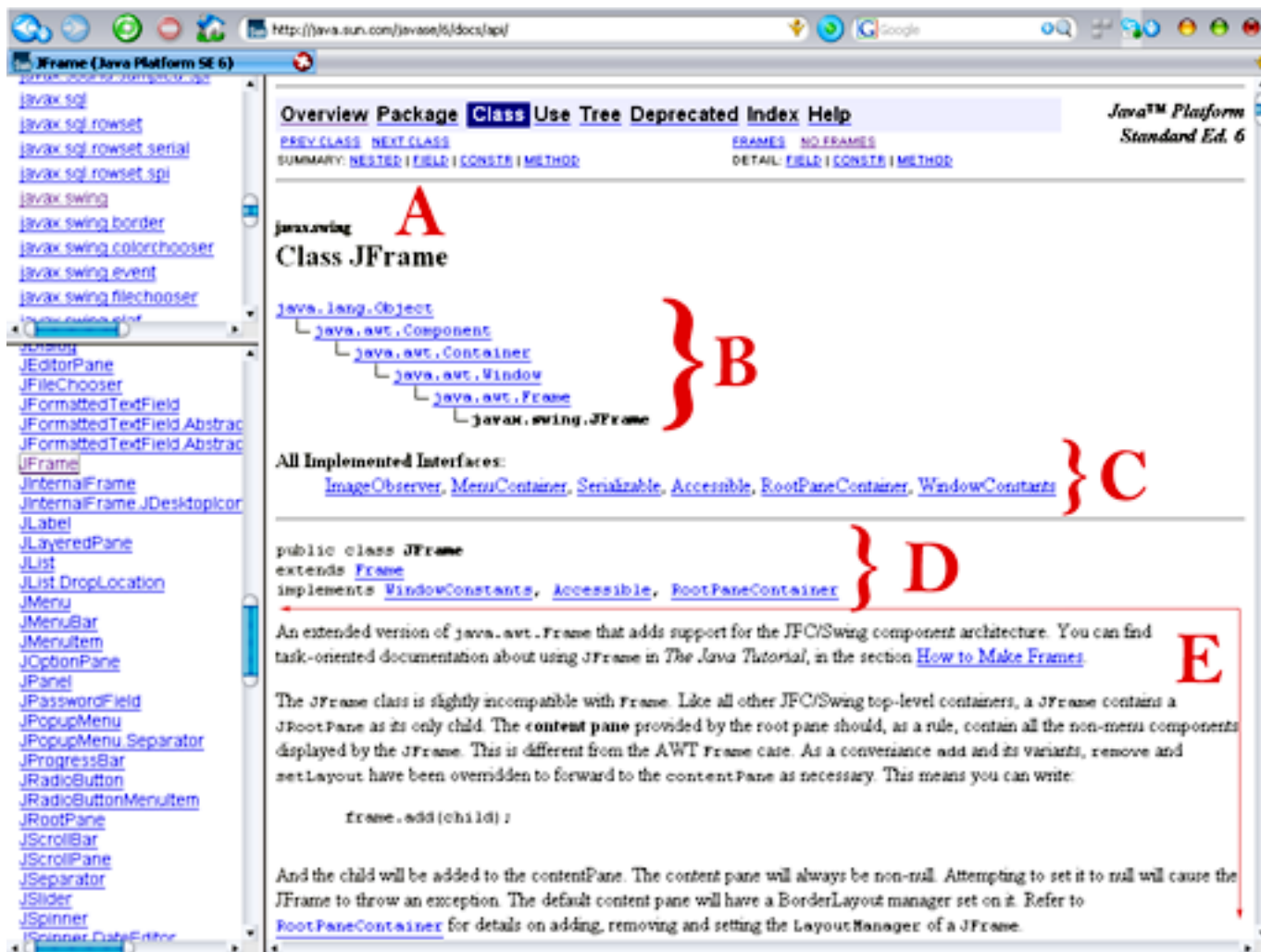
Si par contre vous n'avez aucune idée de la classe qui pourrait contenir cette méthode, vous pouvez sûrement déterminer le package. Par exemple: vous souhaitez réaliser une connexion réseau, le package doit être java.net, le nom laisse à penser que cela concerne la programmation réseau.

Cherchez alors cette classe dans la liste des classes dans la fenêtre de gauche. Rappelez-vous que cette liste peut être affinée si vous connaissez déjà le package de la classe.

Le mieux, si vous débutez dans un domaine de Java (graphismes, réseaux, fichiers, XML) est un tutoriel. Il vous expliquera quelles sont les classes à utiliser, et vous apprendra le nécessaire pour vous débrouiller.

IV - Localisation des informations importantes

Une fois que vous avez ouvert une page concernant une classe bien précise, une multitude d'informations s'offrent à vous, et il n'est pas toujours aisé de s'y retrouver.



Réprésentation d'une classe dans la Javadoc

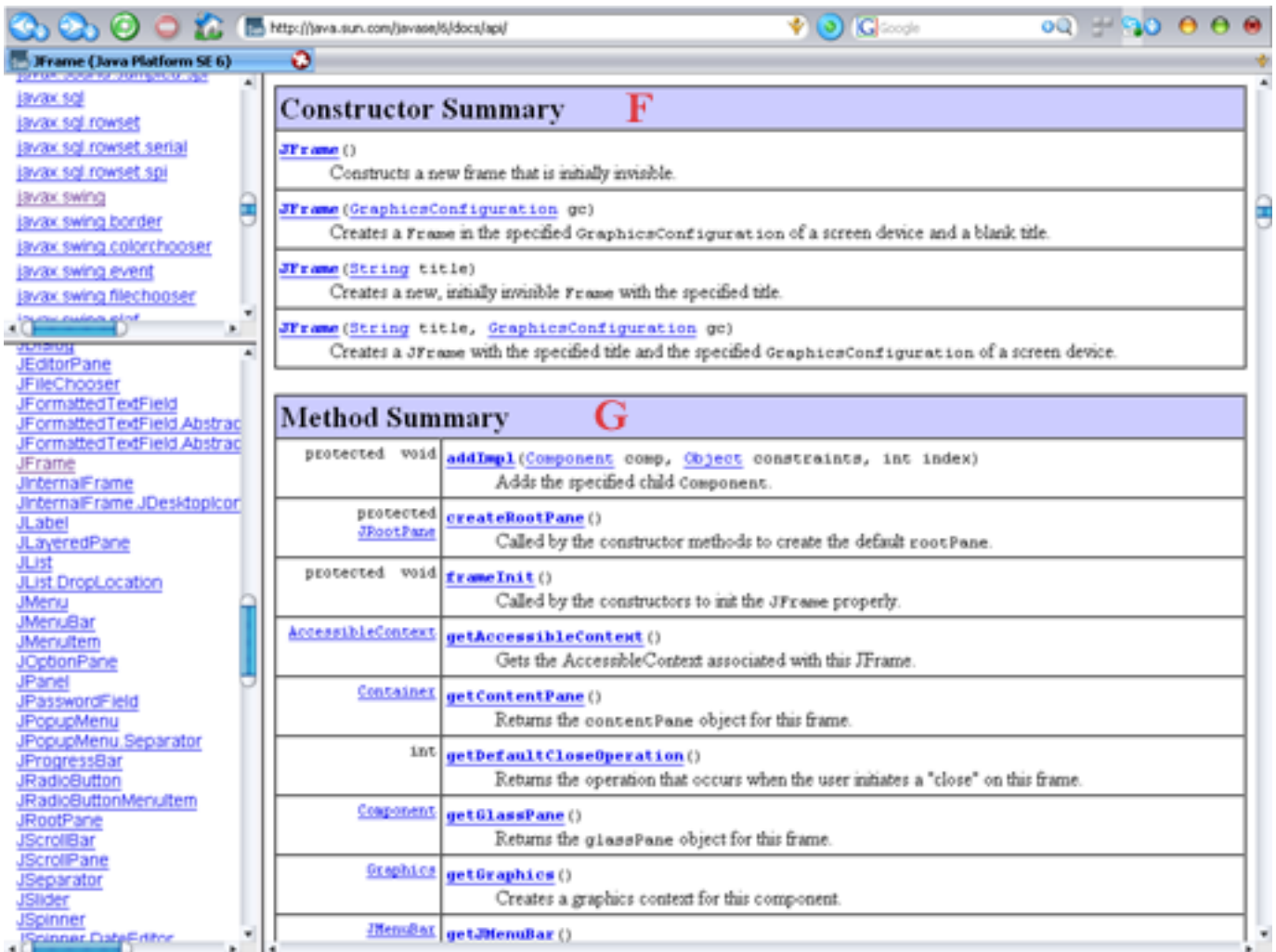
La zone A contient simplement le package et le nom de la classe sélectionnée. La zone B contient toute l'arborescence de la classe, c'est-à-dire sa classe parente, puis la classe parente de sa classe parente, etc.

La zone C contient toutes les interfaces qui sont implémentées par la classe. Il peut y avoir d'autres informations. Par exemple, si vous avez sélectionné une interface, cette zone contiendra toutes les classes qui implémentent cette interface.

La zone D contient la déclaration complète de la classe.

Enfin, la zone E contient une description de la classe, avec parfois un exemple d'utilisation.

Plus bas, d'autres informations, avec notamment :



Constructor Summary F

JFrame ()	Constructs a new frame that is initially invisible.
JFrame (GraphicsConfiguration gc)	Creates a <code>Frame</code> in the specified <code>GraphicsConfiguration</code> of a screen device and a blank title.
JFrame (String title)	Creates a new, initially invisible <code>Frame</code> with the specified title.
JFrame (String title, GraphicsConfiguration gc)	Creates a <code>JFrame</code> with the specified title and the specified <code>GraphicsConfiguration</code> of a screen device.

Method Summary G

protected void	addImpl (Component comp, Object constraints, int index) Adds the specified child <code>Component</code> .
protected JRootPane	createRootPane () Called by the constructor methods to create the default <code>rootPane</code> .
protected void	frameInit () Called by the constructors to init the <code>JFrame</code> properly.
AccessibleContext	getAccessibleContext () Gets the <code>AccessibleContext</code> associated with this <code>JFrame</code> .
Container	getContentPane () Returns the <code>contentPane</code> object for this frame.
int	getDefaultCloseOperation () Returns the operation that occurs when the user initiates a "close" on this frame.
Component	getGlassPane () Returns the <code>glassPane</code> object for this frame.
Graphics	getGraphics () Creates a <code>graphics</code> context for this component.
JMenuBar	getMenuBar ()

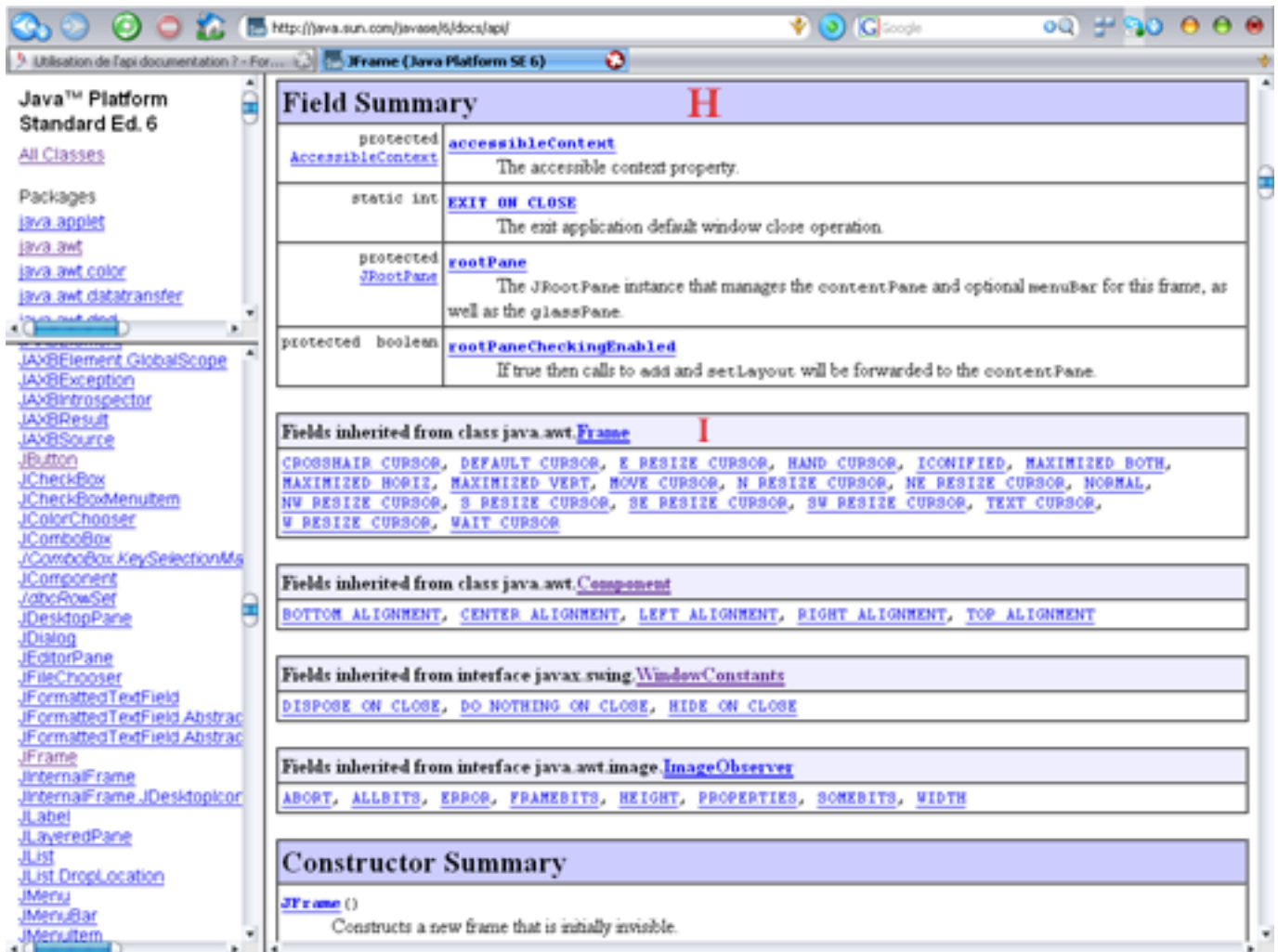
Liste des méthodes disponibles

- la zone F qui contient tous les constructeurs (car ici il s'agit d'une classe, si on avait choisi une interface, ce « block » n'existerait pas). Il y a les paramètres et une description. Si vous cliquez sur le constructeur, vous irez plus bas dans la page, où il y a une description beaucoup plus détaillée.

- la zone G qui contient ce que vous cherchez, la liste de toutes les méthodes de la classe avec là aussi une petite description, les paramètres, la valeur de retour et la visibilité de la méthode (public, private, protected, static ou non, etc.). De plus, si vous cliquez sur la méthode, vous irez plus bas dans la page pour avoir une description plus complète et des liens connexes.

Si vous cliquez sur un paramètre, vous irez sur la page consacrée à la classe (ou interface) de ce paramètre. Vous pouvez naviguer ainsi dans la JavaDoc.

Vous avez également accès aux variables de la classe :



Variables et champs de la classe

- la zone H qui contient toutes les variables déclarées dans la classe elle-même.
- la zone I qui contient les variables héritées de la classe parente directe. Généralement, les zones « Fields inherited from... » contiennent les variables des classes parentes en descendant la hiérarchie.

De plus, il peut arriver qu'un champ contienne la mention « Deprecated » comme le champ suivant :

```
int getCursorType ()
    Deprecated. As of JDK version 1.1, replaced by Component.getCursor().
```

Les champs dépréciés

Ce qui suit est valable pour les constructeurs et les variables également. Cette mention indique que la méthode/variable ou constructeur a été déprécié, c'est-à-dire que son utilisation est désormais déconseillée dans les nouvelles versions du JDK parce qu'une nouvelle méthode apporte de meilleurs résultats/performances par exemple. Sinon, vous pouvez également trouver une mention « Since x.x ». Cette mention indique la version du JDK dans laquelle la ressource est apparue pour la première fois.

V - La recherche simplifiée par JavaSearch

Comme vous avez pu le voir, la JavaDoc est très puissante et sans limite. Cependant, vous serez peut être fatigué de devoir naviguer longtemps entre les classes et méthodes pour trouver votre bonheur, aussi simple soit-il. **JavaSearch** a donc été créé.

Cet outil vous permet de rechercher dans toute la Javadoc une classe, un package, une méthode ou un constructeur. Vous pouvez spécifier plusieurs options de recherche. Son utilisation est simple et très intuitive. Toute la Javadoc est intégrée à JavaSearch, ainsi, dès que vous avez trouvé ce que vous cherchiez, vous pourrez consulter le contenu directement sur le site !

Vous pouvez restreindre votre recherche à une version de Java SE (1.4, 1.5, 1.6) ou Java EE 5. Bref, un lien à mettre tout de suite dans vos favoris :)

VI - Conclusion

La connaissance de la Javadoc est indispensable pour tout programmeur Java qui se respecte. Elle va vous ouvrir des horizons infinis. Utilisée avec un outil comme JavaSearch, la Javadoc devient vraiment puissante, et votre apprentissage de Java sera facilité.

Retenez cependant que la Javadoc n'est ni un tutoriel ni un cours. C'est une documentation, qui se veut complète et exhaustive, mais en aucun cas « ludique ». Son but n'est pas de vous apprendre à programmer en Java.

VII - Remerciements

Je remercie buchs et bmeurant pour leurs corrections orthographiques, Hikage, le y@m's, wichtounet et Ricky 81 pour leurs conseils, et plus généralement l'équipe de rédaction pour leur support et leur aide précieuse (ainsi que leur patience légendaire).

