

Astuces Java

Les librairies qui changent la vie...
;-)

*Évènement du Comité Technique - 13/06/2012
Préparé et animé par Vivian & Bruno*



Java API 1.5+

Copie de tableaux

(voir `ArrayUtils` d'Apache Commons Lang pour les méthodes utilitaires concernant les tableaux, ainsi que Guava, notamment pour la copie de collection "intelligente", voir les contraintes)

PROBLÉMATIQUE

SOLUTION

```
String[] monTableauOriginal = new String[5];
```

```
String[] copieTableau = Arrays.copyOf(monTableauOriginal,  
    monTableauOriginal.length);
```

```
String[] copiePartielleTableau = Arrays.copyOfRange(monTableauOriginal, 0,  
    2);
```

Java API 1.5+

Modification d'une collection pendant son parcours

PROBLÉMATIQUE

SOLUTION

```
List<String> liste = new CopyOnWriteArrayList<String>();  
for (String element : liste) {  
    if (element.startsWith("a")) {  
        liste.remove(element);  
    }  
}
```

Java API 1.5+

Compter le nombre d'occurrences d'un objet dans une collection

PROBLÉMATIQUE

`Collections.frequency(collection, object);`

SOLUTION

Java API 1.5+

Manipuler une collection

PROBLÉMATIQUE

SOLUTION

`Collections.emptyList();` // existe aussi pour Set, Map

`Collections.sort(liste);`

`Collections.sort(liste, unComparator);`

`Collections.sort(liste, Collections.reverseOrder());`

`Collections.unmodifiableList(liste);` // existe aussi pour Set, Map, SortedSet, SortedMap

`Collections.reverse(liste);`

Apache Commons Lang

Déterminer l'égalité ou le résultat de la comparaison de deux objets, sachant qu'ils peuvent tous les deux être NULL

PROBLÉMATIQUE

SOLUTION

```
ObjectUtils.equals(object1, object2);  
ObjectUtils.compare(object1, object2);
```

Apache Commons Lang

Renvoyer un objet différent si l'objet courant/testé est NULL

PROBLÉMATIQUE

```
ObjectUtils.defaultIfNull(objet, objetParDefautSiNull);
```

SOLUTION

Apache Commons Lang

Créer une liste/map à partir des valeurs d'une "enum"

PROBLÉMATIQUE

SOLUTION

```
EnumUtils.getEnumList(enumClass);  
EnumUtils.getEnumMap(enumClass);
```


Apache Commons Lang

Rendre une condition booléenne lisible, manipuler des données booléennes

PROBLÉMATIQUE

SOLUTION

```
BooleanUtils.and(condition1, condition2, condition3);
```

```
BooleanUtils.or(condition1, condition2, condition3);
```

```
BooleanUtils.toBoolean(nombreATester, nombrePourValeurVrai,  
nombrePourValeurFaux);
```

```
BooleanUtils.toBoolean(chaineATester, chainePourValeurVrai,  
chainePourValeurFaux);
```

```
BooleanUtils.toString(conditionObjetBoolean, chainePourValeurVrai,  
chainePourValeurFaux, chaineSiBooleanNull);
```

Apache Commons Lang

Manipulation de chaînes de caractères

PROBLÉMATIQUE

SOLUTION

StringUtils & WordUtils

```
StringUtils.abbreviate(maChaine, 10);
```

```
StringUtils.defaultIfEmpty(maChaine, "aucune valeur");
```

```
StringUtils.join(listeDeChaines, " - ");
```

```
StringUtils.difference("abcde", "abxyz"); // "xyz"
```

```
StringUtils.capitalize("des astuces sont présentées"); // Des astuces sont présentées
```

```
WordUtils.capitalize("des astuces sont présentées"); // Des Astuces Sont Présentées
```

```
WordUtils.initials("Chuck Norris"); // CN
```

Apache Commons Lang

Échapper les caractères d'une chaîne pour un format/langage particulier ou retirer les caractères d'échappement

PROBLÉMATIQUE

SOLUTION

```
StringEscapeUtils.escapeCsv(input);
```

```
StringEscapeUtils.escapeXml(input);
```

```
StringEscapeUtils.escapeJava(input);
```

```
StringEscapeUtils.escapeHtml4(input);
```

```
StringEscapeUtils.unescapeXXX(input);
```

Apache Commons Lang

Générer des chaînes de caractères

PROBLÉMATIQUE

SOLUTION

```
RandomStringUtils.randomAlphabetic(5); // azepi
```

```
RandomStringUtils.randomAlphanumeric(5); // az12e
```

```
RandomStringUtils.randomNumeric(5); // 45872
```

```
RandomStringUtils.randomAscii(5); // 4e[/A
```

Apache Commons Lang

Accéder à des propriétés relatives au système ou à l'utilisateur

PROBLÉMATIQUE

SOLUTION

```
SystemUtils.FILE_ENCODING;  
SystemUtils.IS_JAVA_1_6;  
SystemUtils.JAVA_IO_TMPDIR;  
SystemUtils.USER_HOME;  
...
```

Apache Commons IO

Manipulation de fichiers : lecture, écriture, copie, comparaison

(voir également IOUtils, qui n'est pas une classe IOcean ! ;-))

PROBLÉMATIQUE

SOLUTION

```
FileUtils.readFileToByteArray(file); FileUtils.readFileToString(file, encoding);  
FileUtils.writeByteArrayToFile(file, data);  
FileUtils.writeStringToFile(file, data, encoding);  
FileUtils.listFiles(dir, fileFilter, dirFilter);  
FileUtils.copyDirectory(srcDir, destDir);  
FileUtils.moveDirectory(srcDir, destDir); FileUtils.moveFile(srcFile, destFile);  
FileUtils.contentEquals(file1, file2);  
FileUtils.sizeOf(file);
```

Apache Commons IO

Manipulation des chemins et noms de fichiers

PROBLÉMATIQUE

SOLUTION

```
FilenameUtils.normalize(filename); /* /foo// = /foo */
```

```
FilenameUtils.normalizeNoEndSeparator(filename); /* /foo// = /foo */
```

```
FilenameUtils.separatorsToUnix | separatorsToWindows(path); /* conversion \ vers / */
```

```
FilenameUtils.getFullPathNoEndSeparator(path); /* C:\a\b\c.txt = C:\a\b */
```

```
FilenameUtils.getName(path); /* a/b/c.txt = c.txt */
```

```
FilenameUtils.getExtension(path); /* a/b/c.jpg = ".jpg" */
```

```
FilenameUtils.isExtension("monFichier.png", new String[] {"png", "jpg"});
```

```
FilenameUtils.wildcardMatch("a.jpg", "*.jpg");
```

Joda Time

Afficher quand a eu lieu la dernière modification de la fiche. Le client ne veut pas que ce soit la date mais depuis quand cette modification a eu lieu

PROBLÉMATIQUE

SOLUTION

```
Period period = new Period(committed, new DateTime());
PeriodFormatterBuilder builder = new PeriodFormatterBuilder().builder
    .appendMinutes().
    .appendSuffix( "minute ago", " minutes ago");
period.toString(builder.toFormatter());
```


Joda Time

Calculer l'âge d'une personne (sans faire un gros paté de code ;))

PROBLÉMATIQUE

SOLUTION

```
DateMidnight birthdate = new DateMidnight(1970, 1, 20);  
DateTime now = new DateTime();  
Years.yearsBetween(birthdate, now).getYears()
```

Joda Time

Addition et soustraction de durée(s)

PROBLÉMATIQUE

SOLUTION

```
DateTime dt = new DateTime(2005, 3, 26, 12, 0, 0, 0);
DateTime plusPeriod = dt.plus(Period.days(1));
DateTime plusDuration = dt.plus(new Duration(24L*60L*60L*1000L));

dt.minus(Period.months(2));

// enlève l'écart de durée qu'il y a entre dt et aujourd'hui
DateTime minusDuration = dt.minus(new Duration(dt, new DateTime()));
```

Guava

PROBLÉMATIQUE

Pré conditions ou s'assurer que le contrat de notre api est respecté

SOLUTION

```
import static com.google.common.base.Preconditions.*;

public List<Invoice> getInvoices(Account compte) {
    checkNotNull(compte); // déclenche un null pointer exception
}

public void updateGasEmission(float emission) {
    checkArgument(emission > 10, "emission (%s) rate must be superior",
emission);
}
```

Guava

Eviter de retourner null. Indiquer clairement que le retour d'une méthode peut ne pas avoir de résultat.
Eviter les NullPointerException. Améliorer la lisibilité

PROBLÉMATIQUE

Exemple, je veux afficher la ville d'une personne

//API

```
public Optional<Address> getAddress() {  
    return Optional.fromNullable(address)  
}
```

//Client de l'API

```
personne.getAdress().or(defaultAddress).getVille();
```

//Vérifier si la personne a une adresse :

```
personne.getAdress().isPresent()
```

SOLUTION

Guava

Ecrire rapidement une méthode permettant de consulter les objets de manière lisible dans le debugger
=> redéfinition du toString() avec un helper

PROBLÉMATIQUE

SOLUTION

```
@Override
public String toString() {
    return Objects.toStringHelper(this)
        .add("firstName", firstName)
        .add("lastName", lastName)
        .omitNullValues()
        .toString();
}
```

Guava

Scinder une chaîne (avec gestion correcte des espaces ou valeurs vides) ou réunir plusieurs chaînes (avec gestion des null)

PROBLÉMATIQUE

```
Splitter.on(",")  
  .omitEmptyStrings()  
  .trimResults()  
  .split("test1, , test2,      test3");  
// on obtient ["test1", "test2", "test3"]
```

SOLUTION

```
Joiner.on(",")  
  .skipNulls()  
  .join("test1", null, "test2");  
// "test1, test2"  
// on peut aussi utiliser .useForNull("default")
```

Guava

Créer et manipuler une collection qui permet de gérer plusieurs valeurs pour une même clé ; utilisée pour représenter un graphe par exemple

PROBLÉMATIQUE

SOLUTION

```
Multimap<Integer, String> map= ImmutableSetMultimap.of(1, "a", 2, "b", 3, "c",  
1, "a2");
```

```
Multimap<Integer, String> map = new ImmutableSetMultimap.Builder<Integer,  
String>()  
    .put(1, "a")  
    .put(2, "b")  
    .put(3, "c")  
    .put(1, "a2")  
    .build();
```

Guava... à découvrir et approfondir

Immutable collections...

Multiset...

BiMap...

Table...

ClassToInstanceMap...

Contraintes...

Classes utilitaires pour manipulation et opérations sur les collections...

I/O...

Messages...

Programmation concurrente...

Programmation fonctionnelle...

Questions / Echanges

Vous connaissiez ?

Qu'est-ce que vous utilisez déjà ?

Qu'est-ce que vous utiliserez ?

Qu'est-ce que vous utilisez d'autre ?

Quelles sont les autres problématiques récurrentes que vous rencontrez ?

(ROTI) Est-ce que cette présentation en valait la peine ?

1 : Inutile, 2 : Tout juste utile, 3 : Moyenne, 4 : Bonne, 5: Excellente

Ressources

Apache Commons

- <http://commons.apache.org/>

Guava

- <http://code.google.com/p/guava-libraries/>

JodaTime

- <http://joda-time.sourceforge.net/>