

Знакомство с проектом Apache

JSP & Servlets
20 уровень, 0 лекция

ОТКРЫТА

Знакомство с Apache Commons

Конечно, начнем с истории!

Все началось в 1999 году с регистрации “Apache Group” от имени Apache Software Foundation (далее ASF). Проектом, который поддерживался фондом, был Apache HTTPD Web Server, созданный между 1995 и 1999 годами.

Таким же стал и проект Jakarta Project (проект Jakarta), который появился в результате сотрудничества Sun Microsystems, IBM, Oracle и ребят из Apache. И в 2001 в ходе работы команда разработчиков заметила, что они часто пишут одинаковую функциональность, иногда просто напросто копируют ее друг у друга. Такой код называется шаблонным. У них получилось собрать большое количество кода, который помогал разработчикам, но не было библиотеки для его хранения.

Так появился проект Jakarta Commons, в который добавляли компоненты на Java (в основном на основе уже существующего кода). Позже проект был переименован в Apache Commons.

Если говорить более широко, то Apache Commons — это “большая коллекция маленьких Java-утилит”. Она используется во многих проектах с открытым исходным кодом.

Утилиты Apache Commons лежат в основе таких проектов, как Apache Tomcat, Struts, Hibernate и др.

Конечно, все это можно подключить ручками, без системы сборки (Maven, Gradle), но мы не будем этого делать и просто добавим их в наш проект.

Для работы с Maven вначале добавляем соответствующую зависимость:

```
<dependency>
  <groupId>org.apache.commons</groupId>
  <artifactId>commons-lang3</artifactId>
  <version>${apache.common.version}</version>
</dependency>
```

Где `${apache.common.version}` — это версия этой библиотеки.

Для Gradle (Groovy):

```
implementation 'org.apache.commons:commons-lang3:3.12.0'
```

Популярные библиотеки Apache Commons

Вот список наиболее используемых классов и методов:

Apache Commons: Lang

В данной библиотеке есть такие пакеты:

1	Packages
2	org.apache.commons.lang
3	org.apache.commons.lang.builder
4	org.apache.commons.lang.enum

5	org.apache.commons.lang.enums
6	org.apache.commons.lang.exception
7	org.apache.commons.lang.math
8	org.apache.commons.lang.mutable
9	org.apache.commons.lang.reflect
10	org.apache.commons.lang.text
11	org.apache.commons.lang.time

Здесь можно удобно и быстро работать со строками, reflection, сериализацией, объектами и системой. Отметим наиболее используемые методы:

StringUtils

Огромное количество методов для манипуляций со строками.

- `is(Not)Blank/Empty(String)` — о проверке такого типа: `if (s!=null && s.trim().length()>0)` пора забыть, и тут есть хорошая замена

StringEscapeUtils

- `(un)escapeSql(String)` — замена `PreparedStatement`
- `(un)escapeHtml(String)` — для обработки значений из HTML

ToStringBuilder

- `reflectionToString(Object)` — реализация `toString()`, основанная на рефлексии. При удалении какого-то поля с помощью рефлексии результат метода изменится.

EqualsBuilder & hashCodeBuilder

- `reflectionEquals/HashCode(Object)` — хорошая замена автоматической генерации со своим плюсом: эти два метода учитывают структурные изменения объекта в процессе работы, например, добавление полей

ExceptionUtils

- `getFullStrackTrace(Throwable)` — вывод всего StackTrace в виде строки

Apache Commons: Collections

1	Packages
2	org.apache.commons.collections4
3	org.apache.commons.collections4.bag
4	org.apache.commons.collections4.bidimap
5	org.apache.commons.collections4.collection
6	org.apache.commons.collections4.comparators
7	org.apache.commons.collections4.functors
8	org.apache.commons.collections4.iterators
9	org.apache.commons.collections4.keyvalue
10	org.apache.commons.collections4.list
11	org.apache.commons.collections4.map
12	org.apache.commons.collections4.multimap
13	org.apache.commons.collections4.multiset
14	org.apache.commons.collections4.properties
15	org.apache.commons.collections4.queue
16	org.apache.commons.collections4.sequence
17	org.apache.commons.collections4.set
18	org.apache.commons.collections4.splitmap
19	org.apache.commons.collections4.trie
20	org.apache.commons.collections4.trie.analyzer

`CollectionUtils` — класс для удобной работы с коллекциями:

`filter/find(Collection, Predicate)` — фильтрация и поиск по предикату `forAllDo(Collection, Closure)` — выполняет Closure для каждого элемента, но этот метод *deprecated*, используйте `Iterator.forEach()` `is(Not)Empty(Collection)` — позволяет не проверять на null перед вызовом `isEqualCollection(Collection, Collection)` — помогает сравнить две коллекции

Есть также много других классов разного уровня полезности. Здесь и дальше я перечисляю наиболее часто используемые чисто в моем случае.

Apache Commons: IO

1	<code>Packages</code>
2	<code>org.apache.commons.io</code>
3	<code>org.apache.commons.io.comparator</code>
4	<code>org.apache.commons.io.file</code>
5	<code>org.apache.commons.io.file.spi</code>
6	<code>org.apache.commons.io.filefilter</code>
7	<code>org.apache.commons.io.function</code>
8	<code>org.apache.commons.io.input</code>
9	<code>org.apache.commons.io.input.buffer</code>
10	<code>org.apache.commons.io.monitor</code>
11	<code>org.apache.commons.io.output</code>
12	<code>org.apache.commons.io.serialization</code>

Дополнительно помогает работать с файлами в Java:

FileUtils

- `copyDirectory(File, File)` — копирование директорий
- `copyFile(File, File)` — копирование файлов
- `listFiles(File, String[], boolean)` — список файлов по расширению и рекурсивно
- `readFileToString(File, String)`
- `writeStringToFile(File, String)`

IOUtils

- `closeQuietly(Reader/Writer/InputStream/OutputStream)` — закрывает поток данных
- `copy(InputStream, OutputStream)` — копирование из одного потока в другой

[← Предыдущая лекция](#)

[Следующая лекция →](#)

+11

Комментарии (2)

популярные

новые

старые

JavaCoder

Введите текст комментария

Шикарная библиотечка..

Ответить

+2

Alexey Svorkin

Уровень 37

31 мая 2022, 15:11

я первый не первый

Ответить

+2

ОБУЧЕНИЕ

- Курсы программирования
- Курс Java
- Помощь по задачам
- Подписки
- Задачи-игры

СООБЩЕСТВО

- Пользователи
- Статьи
- Форум
- Чат
- Истории успеха
- Активности

КОМПАНИЯ

- О нас
- Контакты
- Отзывы
- FAQ
- Поддержка



JavaRush — это интерактивный онлайн-курс по изучению Java-программирования с нуля. Он содержит 1200 практических задач с проверкой решения в один клик, необходимый минимум теории по основам Java и мотивирующие фишки, которые помогут пройти курс до конца: игры, опросы, интересные проекты и статьи об эффективном обучении и карьере Java-девелопера.

ПОДПИСЫВАЙТЕСЬ

ЯЗЫК ИНТЕРФЕЙСА

Русский

СКАЧИВАЙТЕ НАШИ ПРИЛОЖЕНИЯ



