

Murdoc Home

Powered by Open Sources Licences from:

Adaptavist Theme Builder, Atlassian Jira, Atlassian Confluence, GreenHopper Plugin, Yourkit Java Profiler

[Download \(stable\)](#) | [JIRA](#) | [Code \(browse | checkout | statsvn | svnstat\)](#) | [Bamboo](#) | [Fisheye](#)

Introdução

Bem vindo.

Murdoc é uma biblioteca extensível para preenchimento de *Data Transfer Objects* (DTOs) em aplicações Java. Se você conhece o [Apache Bean Utils](#), pense em algo realmente muito mais prático, flexível e rápido.

Não deixe de olhar

[Sample Mappings](#)

Uso

Para você utilizar o Murdoc, preenchendo seus DTOs em qualquer lugar, a partir de qualquer origem e com uma única linha de código cliente:

1. [Baixe](#) a ultima versao do JAR do projeto.
2. Importe para seu projeto o JAR do Murdoc.
3. Invoque o Murdoc:

```
//Boierplate code
SomeBackendObject = new mockSomeBackendObject();
AnotherBackendObject anotherBackendObject = new
mockAnotherBackendObject();
AndAnotherAndAnotherAndAnotherBackendObject
andAnotherAndAnotherAndAnotherBackendObject = new
mockAndAnotherAndAnotherAndAnotherBackendObjectBackendObject();
SomeDto someDto = new SomeDto();

//Real interesting code. TIP: You can use multiple backends at
once to fill the DTO
SomeDto = Murdoc.fill(someDto, someBackendObject,
anotherBackendObject, andAnotherAndAnotherAndAnotherBackendObject);
```

Desenvolvimento

Para você estender as funcionalidades e preencher DTOs realmente exóticos:

1. Baixe o projeto do [repositorio](#) (usuario:guest senha:guest)
2. Crie uma classe de conversao.
3. Invoque o Murdoc com sua classe de conversao:

```
//Note the conversion class used to translate backend objects to
DTO.
MyExoticDto myExoticDto = Murdoc.fill(new MyExoticDto(),
MyExoticConversionClass.class, myBackendObject,
andAnotherBackendObject);

//Here, the conversion class is attached to DTO using annotation.
So, it is unnecessary here.
MyExoticDto myExoticDto = Murdoc.fill(new MyExoticDto(),
myBackendObject, andAnotherBackendObject);
```

Exemplo de Uso

Veja página [Sample Mappings](#)