



# exemplos

Pequenos exemplos como ferramentas didáticas

 Search projects[Project Home](#) [Wiki](#) [Issues](#) [Source](#)Search  Current pages ▼ for  Search

## ☆ equinox

Introdução ao Equinox (Fábio Nogueira de Lucena)

Updated Feb 4, 2010 by [projeto....@gmail.com](#)

## Introdução ao Equinox ¶

### Equinox é uma implementação de OSGi

Equinox é uma implementação da especificação OSGi. Detalhes de OSGi podem ser obtidos em <http://osgi.org>. A versão mais recente da implementação Equinox pode ser obtida em <http://www.eclipse.org/equinox/>.

OSGi é uma tecnologia que oferece para Java um padrão para montagem dinâmica de sistemas baseados em módulos. Dito de outra forma, permite a definição de módulos, gerencia a visibilidade de APIs oferecidas pelos módulos, gerencia dependências entre os módulos e as versões destes. Módulos são componentes menores e reutilizáveis. Tais módulos são denominados pela OSGi como bundles. Doravante empregaremos o termo bundle que é a unidade de modularização em OSGi.

Bundles podem ser instalados, iniciados, interrompidos e desinstalados e atualizados em tempo de execução. Bundles publicam serviços dinamicamente e podem descobrir e requisitar serviços disponibilizados por outros bundles. Em consequência, serviços podem aparecer e desaparecer em tempo de execução. O programador Java poderá consultar a documentação em <http://www2.osgi.org/javadoc/r4/> para detalhes da versão 4 da especificação.

OSGi é uma especificação, ou seja, documenta e atribui funções precisas a tais serviços, mas não os implementa. Equinox, por outro lado, é uma implementação desta especificação.

Existem várias outras implementações de OSGi além de Equinox. Algumas delas incluem Apache Felix (<http://cwiki.apache.org/FELIX/index.html>) e Knoplerfish (<http://www.knoplerfish.org/>) e Modulefusion (um exemplo de como fazer uso) (<http://code.google.com/p/modulefusion/>).

Um aspecto relevante é que *qualquer bundle pode ser executado em qualquer implementação de OSGi*. Ou seja, um bundle desenvolvido e testado na implementação Felix pode ser executado na implementação Equinox e vice-versa.

### Instalação

A versão mais recente do Equinox pode ser obtida em <http://download.eclipse.org/eclipse/equinox/>. A implementação do Equinox é fornecida em um único arquivo .jar, algo como **org.eclipse.osgi\_3.2.2.R32x\_v20070118.jar**. Naturalmente, o nome deste arquivo irá variar conforme a versão obtida. Por simplicidade, você pode simplesmente renomear o arquivo obtido para **osgi.jar**. Não há nenhum prejuízo de funcionalidade e foi exatamente o que foi feito para efeito dos exemplos abaixo.

Toda instalação do Eclipse também contém uma implementação do OSGi. Este arquivo está no diretório plugins onde for instalado o Eclipse.

### Executando o Equinox

Para iniciar o Equinox, simplesmente execute o comando abaixo:

```
java -jar osgi.jar -console
```

A execução deste comando irá oferecer ao usuário a seguinte resposta:

```
osgi>
```

Isto significa que a implementação de OSGi empregada está em execução. Ou seja, está pronta para oferecer serviços. Para obter uma lista das opções disponíveis digite o ponto de interrogação ? pressione a Enter. Como resposta terá uma lista completa dos serviços oferecidos. No restante do texto teremos oportunidade para experimentar parte deles. Alguns, contudo, são amplamente empregados e são reproduzidos abaixo:

- `install <url>` instala um bundle obtido do url fornecido
- `start <bundle id ou nome>` inicia o bundle indicado
- `stop <bundle id ou nome>` interrompe a execução do bundle indicado
- `ss` lista o status dos bundles instalados
- `dig <bundle id ou nome>` relata qualquer problema com o bundle indicado

Observe que as opções, boa parte delas, atuam sobre um bundle específico. De fato, todo o trabalho oferecido por OSGi é para uso dos bundles. Ou seja, primeiro é preciso construí-los antes de usufruir das opções acima.

### Primeiro bundle em três passos

#### Primeiro passo: criar um bundle

Bundle é o elemento de composição de OSGi para o qual são oferecidos serviços. O acesso a eles é obtido por meio de uma instância de BundleContext. Uma simples classe pode obter esta referência ao implementar a interface BundleActivator. Esta interface oferece dois métodos: `start(BundleContext contexto)` e `stop(BundleContext contexto)`.

Abaixo segue um bundle que não faz muito, exceto indicar que foi iniciado (quando iniciado) e que foi interrompido (quando interrompido).

```
package introducao;

import org.osgi.framework.BundleActivator;
import org.osgi.framework.BundleContext;

public class MeuPrimeiroBundle implements BundleActivator {

    public void start(BundleContext context) throws Exception {
        System.out.println("Iniciado");
    }

    public void stop(BundleContext context) throws Exception {
        System.out.println("Interrompido");
    }

}
```

## Segundo passo: criar arquivo com meta-informações (mf)

Deverá ser produzido um arquivo mf (manifest file) contendo informações que serão empregadas pelo OSGi. Para este exemplo segue o arquivo abaixo.

```
Manifest-Version: 1.0
Bundle-ManifestVersion: 2
Bundle-Name: Introdução ao Equinox
Bundle-SymbolicName: introducao
Bundle-Version: 1.0.0
Bundle-Activator: introducao.MeuPrimeiroBundle
Bundle-Localization: plugin
Import-Package: org.osgi.framework;version="1.3.0"
```

Duas das linhas acima são particularmente importantes. Aquela que descreve qual a classe empregada para a ativação do bundle (Bundle-Activator), ou seja, a classe !introducao.MeuPrimeiroBundle (fornecida acima) e as dependências do bundle para pacotes externos (Import-Package).

## Terceiro passo: compilar e empacotar o bundle

A compilação da classe MeuPrimeiroBundle exige o arquivo jar contendo a implementação de OSGi.

```
javac -d \tmp -cp osgi.jar MeuPrimeiroBundle.java
```

O empacotamento de um bundle mínimo exige apenas a classe compilada e o arquivo mf.

```
jar cfm mpb.jar manifes.mf -C \tmp introducao
```

## Executando o bundle criado

Já vimos como executar o Equinox. O comando abaixo instala o bundle contido no arquivo **mpb.jar**. É assumido que a execução do Equinox é iniciada do mesmo diretório onde este arquivo jar está disponível.

```
osgi> install file:mpb.jar
Bundle id is 3
```

Observe o retorno indicando o id do bundle recém instalado. Este id é uma forma de identificar o bundle recém-instalado. Outro comando útil é ilustrado abaixo.

```
osgi> ss
id      State      Bundle
0       ACTIVE     org.eclipse.osgi_3.2.2.R32x_v20070118
3       INSTALLED  introducao_1.0.0
```

Observe que ss (show status) exibe os bundles em execução e o estado correspondente a cada um. Neste momento observa-se que, embora instalado, o bundle de id 3, aquele de nome introducao\_1.0.0, não está ativo. Para ativá-lo execute o comando abaixo.

```
osgi> start 3
Iniciado
```

O retorno fornecido é a mensagem exibida como consequência da execução do método start (veja código da classe MeuPrimeiroBundle). Observe que agora o status foi alterado.

```
osgi> ss
id      State      Bundle
0       ACTIVE     org.eclipse.osgi_3.2.2.R32x_v20070118
3       ACTIVE     introducao_1.0.0
```

Agora, o bundle introducao\_1.0.0 também está ativo. Podemos agora interromper a execução e observar o resultado.

```
osgi> stop 6
Interrompido
```

Nenhuma surpresa, ao interromper a execução o método stop foi executado. Por fim, podemos desinstalar o bundle produzido com o comando abaixo.

```
osgi> uninstall 6
```

Nenhum retorno é fornecido. Contudo, podemos consultar o status, conforme visto acima e verificar que, de fato, a implementação de OSGi em execução agora não mais gerencia o bundle recém-criado.

```
osgi> ss
0    ACTIVE    org.eclipse.osgi_3.2.2.R32x_v20070118
```

## Considerações finais

OSGi oferece uma extensa lista de serviços. Alguns deles foram ilustrados de forma bem simplificada acima. Neste texto empregou-se a implementação conhecida por Equinox, a mesma empregada pelo Eclipse (<http://eclipse.org>). Siga os links no início do texto para detalhes.

---

Comment by [tclj...@gmail.com](mailto:tclj...@gmail.com), Nov 8, 2008

O texto que se refere ao numero de id 6 está errado tem que conter o valor 3!

---

Enter a comment:

Hint: You can use [Wiki Syntax](#).

Submit

[Terms](#) - [Privacy](#) - [Project Hosting Help](#)

Powered by [Google Project Hosting](#)