

Studio

Studio IV

Título do documento

Sumário

[1. Introdução 3](#_Toc372551678)

[1.1. Objetivos do Treinamento 3](#_Toc372551679)

[2. Utilizar *Logs* 4](#_Toc372551680)

[3. Eventos Globais 6](#_Toc372551681)

[3.1. Eventos Disponíveis 6](#_Toc372551682)

[3.2. Customização de Eventos 7](#_Toc372551683)

[4. *Dataset* 13](#_Toc372551684)

[4.1. Tipos de Dataset 13](#_Toc372551685)

[4.2. Consulta de Dataset 14](#_Toc372551686)

[4.3. Customização de Dataset 16](#_Toc372551687)

[4.3.1. Passo a Passo 17](#_Toc372551688)

[4.4. Utilizar Dataset 24](#_Toc372551689)

[4.4.1. Via Ponto de Customização 24](#_Toc372551690)

[4.4.2. Via JavaScript Formulário 26](#_Toc372551691)

[4.4.3. Via Tag Formulário 27](#_Toc372551692)

[4.4.4. Via WebService 28](#_Toc372551693)

[4.4.5. Guia de Referência 28](#_Toc372551694)

[5. Criar e Utilizar Serviços 29](#_Toc372551695)

[5.1. Passo a Passo 29](#_Toc372551696)

[5.2 Implementação 34](#_Toc372551697)

[6. Conclusão 36](#_Toc372551698)

# Introdução

Olá!

Seja bem-vindo ao treinamento ***Studio* IV** do curso ***Studio*** do **Fluig.**

# Objetivos do Treinamento

Ao término deste treinamento você terá conhecido mais sobre:

* Utilizar *Logs*.
* Eventos Globais.
* Customizar Evento.
* *Dataset.*
* Tipos de *Dataset*.
* Consultar *Dataset*.
* *Dataset* Customizado.
* Utilizar *Dataset*.
* Criar e Utilizar Serviços.
* Adicionar e Visualizar um serviço no *Studio*.

# Utilizar *Logs*

Ao desenvolver *scripts* para eventos globais de processos, de formulários ou *datasets é* possível utilizar a variável ***global log*** para obter *feedback* da execução dos *scripts*.

**Níveis**

Existem quatro níveis de *log*:

**1 *- Debug,*** é utilizado para depurar a execução. Seu código é escrito da seguinte forma: ***log.debug***.

**2 *- Error,*** é utilizado para apresentar mensagens de erros. Seu código é escrito da seguinte forma: ***log.error***.

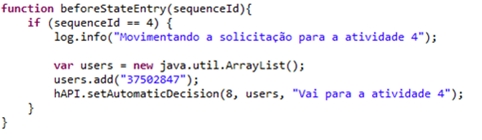
**3 - *Warn,*** é utilizado para informar possíveis problemas de execução. Seu código é escrito da seguinte forma: ***log.warn***.

**4 - *Info,*** é utilizado para apresentar mensagens no *log*. Seu código é escrito da seguinte forma: ***log.info***.



**Exemplo**

A utilização da variável ***global log*** pode ser conferida conforme exemplo.



Neste exemplo apresentado de uma customização de evento de processo, ao movimentar um processo para a atividade 4 será impresso uma mensagem informativa no *log* do Fluig, através da utilização do comando ***log.info***.

# Eventos Globais

Os eventos globais são eventos disparados durante a utilização do **Fluig**, em determinados momentos, como acessar a central de tarefas, publicar ou visualizar um documento. Estes eventos globais podem ser customizados.



# Eventos Disponíveis

A seguir visualize os eventos globais disponíveis no **Fluig.**

***addDocumentConversionExt***

Este evento permite customizar a conversão do documento realizada pelo **Fluig.**

***afterDocumentPublisher***

Este evento permite realizar alguma manipulação customizada nas propriedades da tela de publicação ou edição de documentos depois que ele foi salvo no banco de dados.

***beforeDocumentPublisher***

Este evento permite realizar validações customizadas nas propriedades da tela de publicação ou edição de documentos antes de salvar o documento no banco de dados.

***beforeDocumentViewer***

Este evento permite realizar validações customizadas antes da visualização de um documento.

***displayCentralTasks***

Este evento é disparado sempre antes de ser exibida a central de tarefas, e permite inserir, alterar ou remover os *links* da central de tarefas.

***displayCustomMetadata***

Este evento é disparado sempre antes de ser exibidos os campos customizados de um documento, tanto na publicação como na edição, e permite sugerir valores iniciais para os campos customizados do documento.

***validateCustomMetadata***

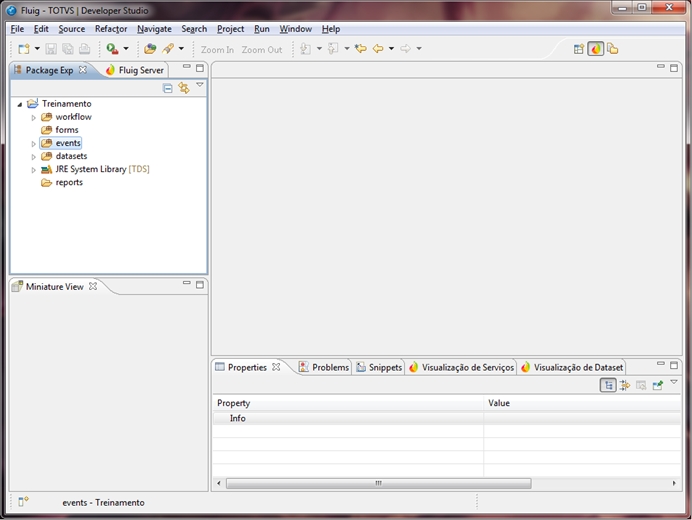
Este evento é disparado sempre antes de salvar os valores dos campos customizados de um documento, tanto na publicação como na edição, e permite alterar ou validar os valores dos campos customizados do documento.

# Customização de Eventos

A seguir visualize o passo a passo para **customizar evento global.**

**Passo 1**

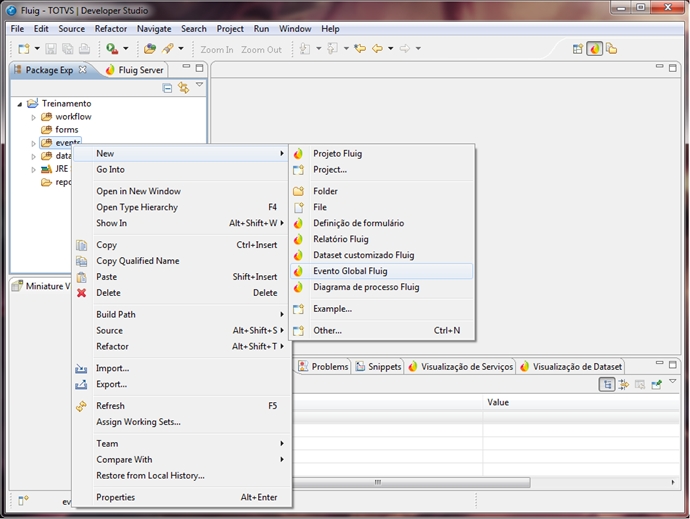
Para customizar um evento global, é necessário possuir um projeto **Fluig** no *Studio.*



No exemplo disposto, é utilizado o projeto **Treinamento.**

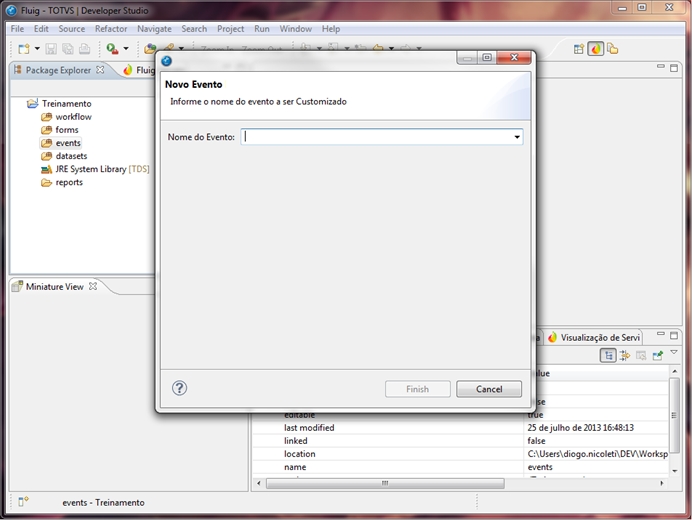
**Passo 2**

Para adicionar uma nova customização de evento global, clique com o botão direito do *mouse* na pasta ***Events*,** selecione a opção ***New*** e, posteriormente, **Evento Global Fluig.**



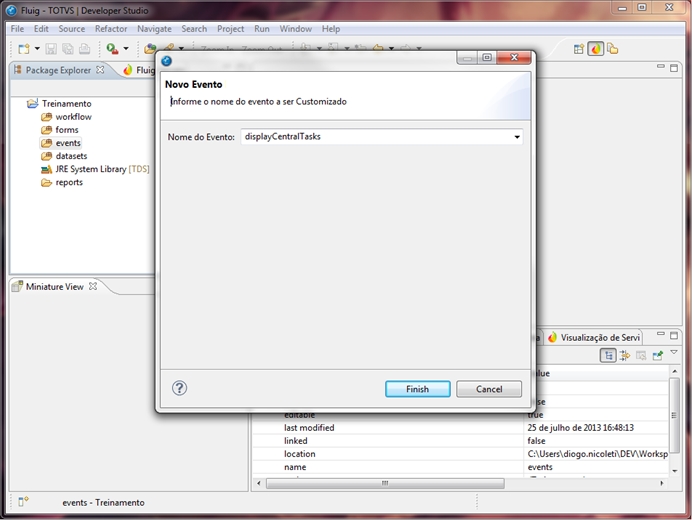
**Passo 3**

O assistente **Novo Evento Fluig** é aberto**.**



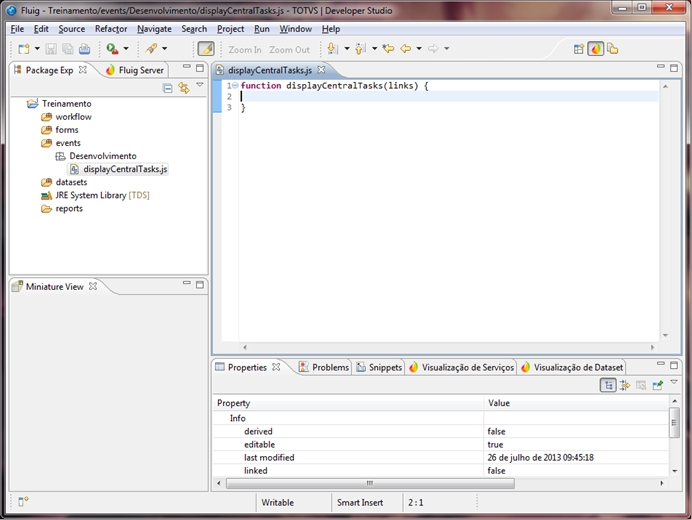
**Passo 4**

Selecione um dos eventos disponíveis no campo **Nome do Evento** e clique no botão ***Finish*.**



**Passo 5**

O evento é adicionado ao projeto dentro da pasta ***Events*** e o ***Script*** *é* aberto para edição.



**Passo 6**

Customização do evento implementada.

Observe que é possível criar outras funções que podem ser utilizadas na função do evento.



No exemplo disposto, foi customizado o evento ***displayCentralTasks*** para adicionar três novos *links* à central de tarefas: **Fluig, Fluig *Store* e *Google*.** Isto foi realizado por meio da função ***addLinks*,** a qual foi foi utilizada no evento ***displayCentralTasks.***

**Passo 7**

Após finalizar a customização do evento, é possível exportá-la para o servidor do **Fluig,** conforme verificado no treinamento ***Studio* I.**

Dessa forma, na **Central de Tarefas** existem três novos *links*, os quais foram adicionados via customização do evento ***displayCentralTasks.***

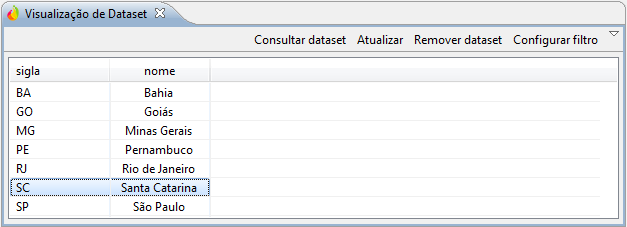


Para mais informações sobre a customização de eventos globais, acessar a página Customização de Eventos, no endereço ***dev.fluig.com.***

# *Dataset*

O *Dataset* é um conjunto de dados que permite padronizar o acesso às informações, independente da origem dos dados, ou seja, é possível acessar informações provenientes de vários locais, como aplicativos externos, banco de dados ou de funcionalidades próprias do **Fluig** por meio de um mesmo componente.

O *Dataset* disponibiliza recursos que permitem consultar seus dados e pode ser utilizado em qualquer ponto onde o **Fluig** permite customizações.



# Tipos de Dataset

Existem três tipos de *datasets*:

* ***Built-in datasets*:**Permite navegar em dados das entidades do próprio **Fluig**.
* ***CardIndex Datasets*:** Permite navegar nos dados existentes em registros de formulário do **Fluig.**
* ***Customized Dataset*:** Permite navegar nos dados retornados por uma customização. Os *Datasets* customizados podem ser utilizados em diversas situações como por exemplo a definição de uma lista de valores fixos, ou para extração de dados de um serviço externo via *WebService*.

Os *Datasets* customizados podem ser utilizados em diversas situações como a definição de lista de valores fixos, ou para extração de dados de um serviço externo via *WebService*.

# Consulta de Dataset

A seguir visualize o passo a passo para consultar ***datasets***.

**Studio**

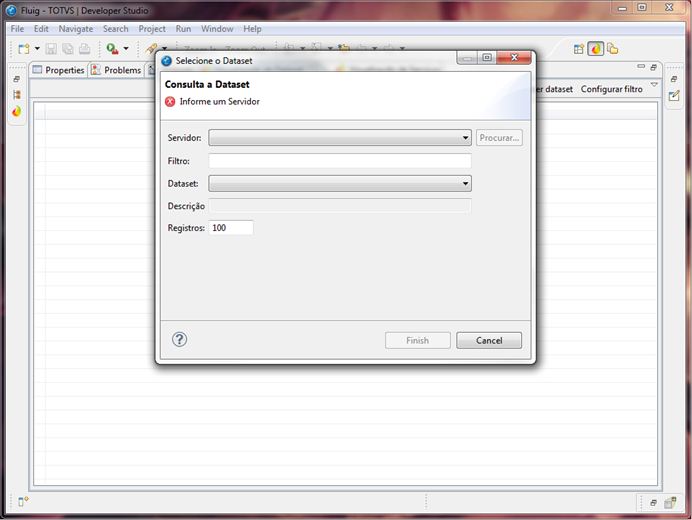
A consulta de *Dataset* é realizada no *Studio* através da visão**Visualização de *Dataset****.*



Para consultar um *Dataset,* clique em **Consultar *Dataset.***

**Assistente**

O assistente **Consulta a Dataset** é aberto.



**Servidor e Dataset**

Selecione o **Servidor** e o ***Dataset*** desejado e clique em ***Finish.***



Caso necessário, é possível limitar a quantidade de registros, informando o valor numérico no campo **Registros.**

**Dados apresentados**

Os dados do *Dataset* são apresentados.



Observe que na visão **Visualização de *Datasets*** existem outras opções:

• **Atualizar:** permite atualizar os dados do *Dataset* em exibição.

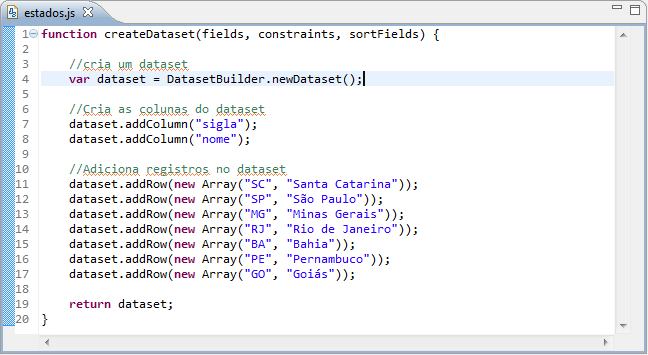
• **Configurar Filtro:** é aberto um assistente para aplicar filtros nos dados do *Dataset.*

• **Remover *Dataset*:** é aberto um assistente para remover *Datasets* do **Fluig.**

# Customização de Dataset

A customização de *dataset* é desenvolvida utilizando a linguagem *JavaScript*, sendo possível realizar chamadas a outros *Datasets*, à serviços externos ou criar o *Dataset* a partir de valores fixos.

Na imagem é apresentada a implementação de um *Dataset* customizado, no qual os campos estão fixos no código.

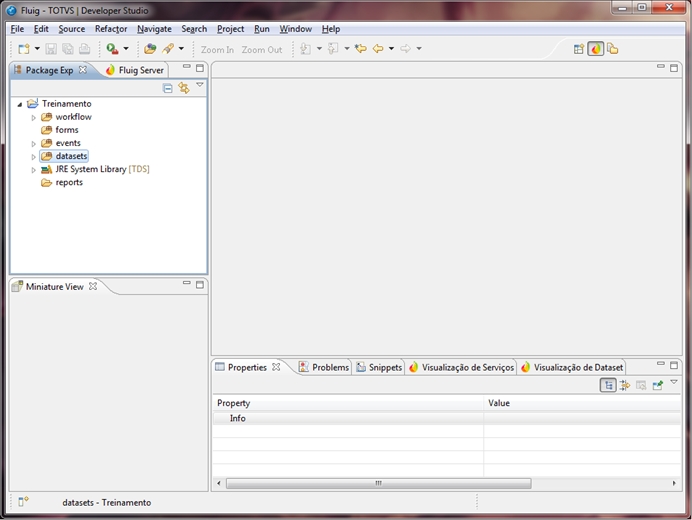


# Passo a Passo

A seguir visualize o passo a passo para desenvolver um **Dataset Customizado** e verificar sua execução no **Studio.**

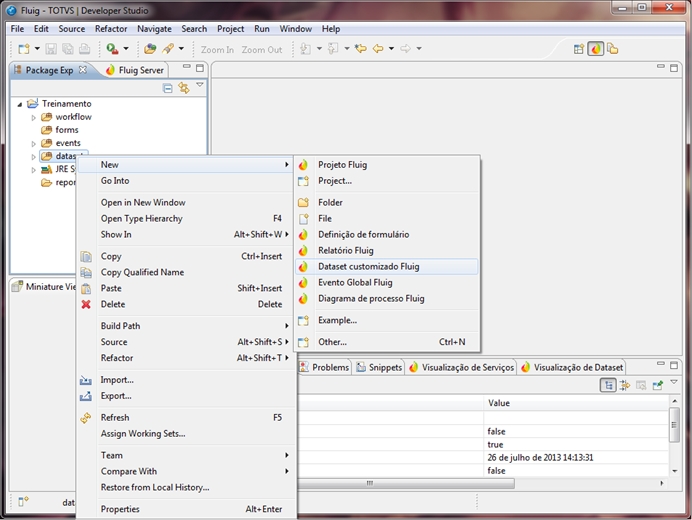
**Passo 1**

Para desenvolver um *Dataset*, é necessário um projeto **Fluig** no *Studio.* No exemplo disposto é utilizado o projeto **Treinamento.**



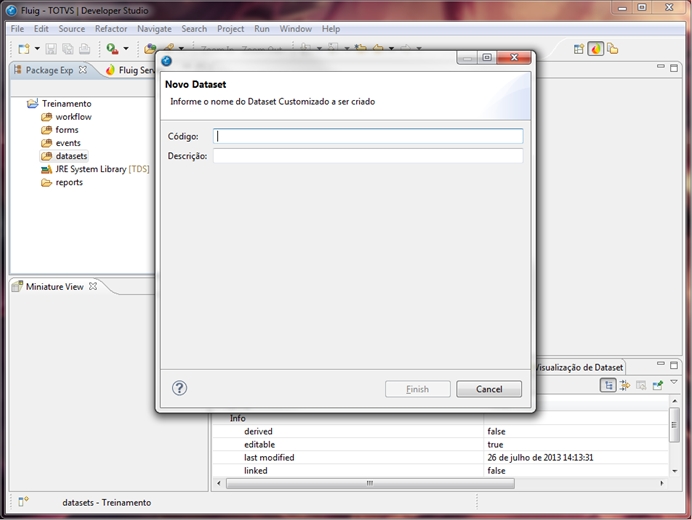
**Passo 2**

Para adicionar um novo ***Dataset* customizado,** clique com o botão direito do *mouse* na pasta ***Datasets,*** selecione a opção ***New*** e posteriormente clique em ***Dataset* Customizado Fluig.**



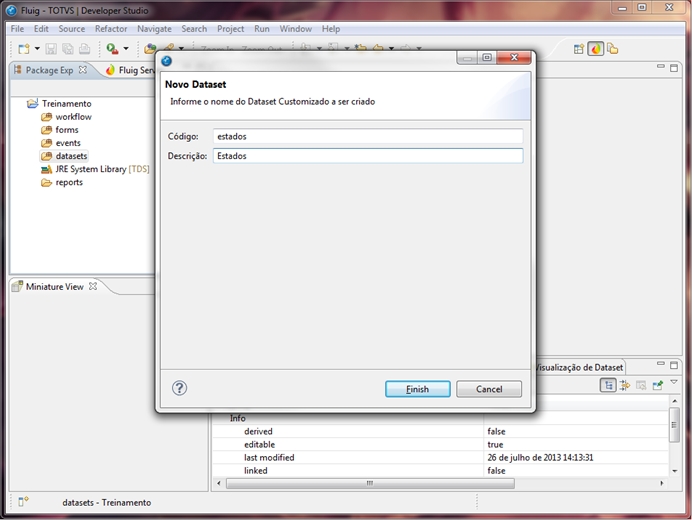
**Passo 3**

O assistente **Novo *Dataset* Fluig** é aberto.



**Passo 4**

Informe o código e a descrição do *Dataset* e clique em ***Finish.***

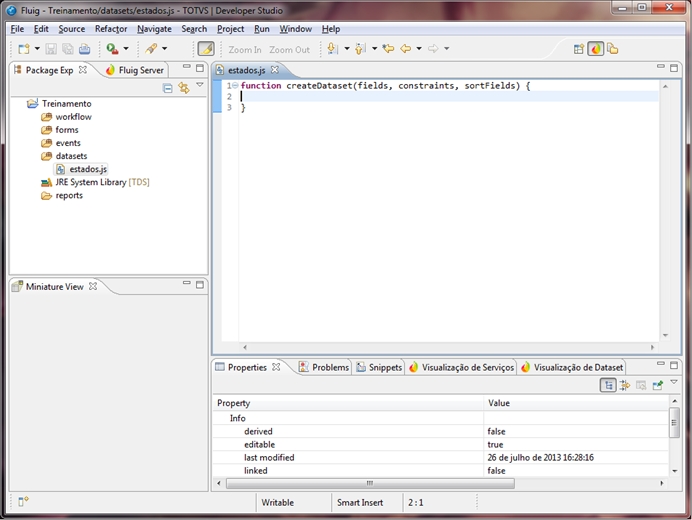


O código informado não pode ser igual ao de um *Dataset* existente no **Fluig.**

**Passo 5**

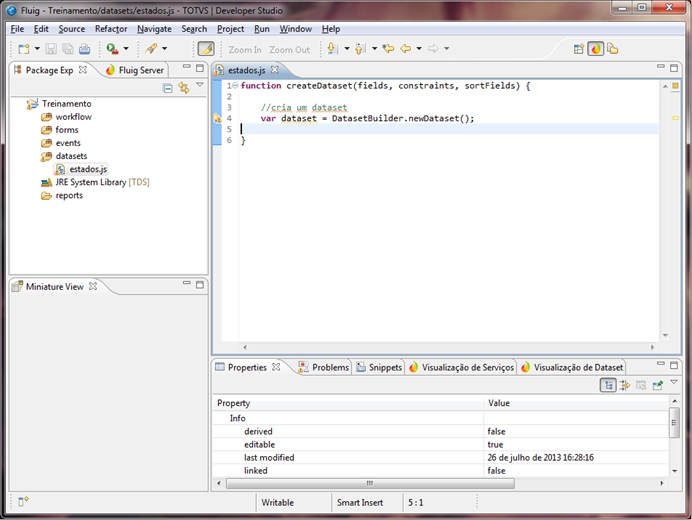
O *Dataset* é adicionado ao projeto dentro da pasta *datasets* e o *script* é aberto para edição.

No exemplo disposto, é codificado um *Dataset* que retornará os estados brasileiros.



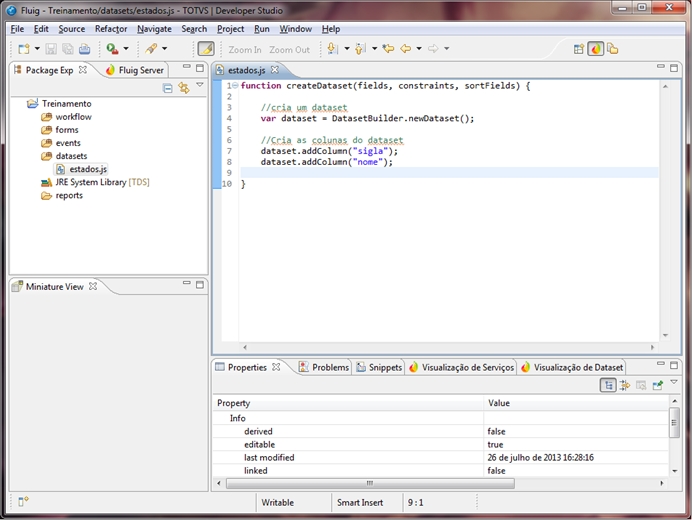
**Passo 6**

Primeiramente crie um novo *Dataset* através do método ***DatasetBuilder.newDataset*.**



**Passo 7**

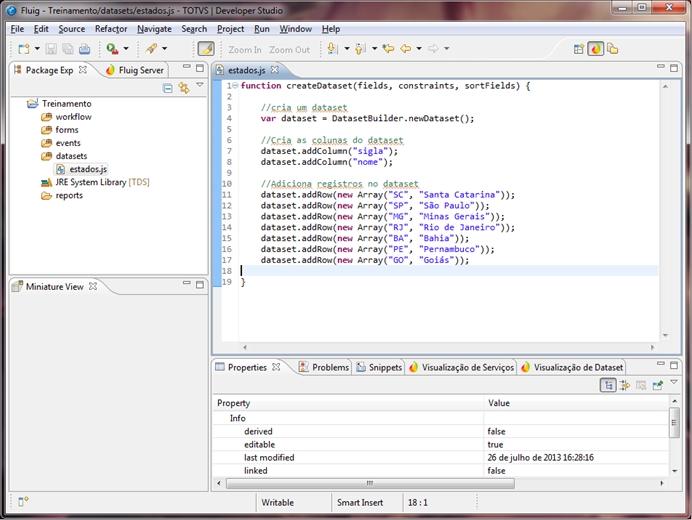
A partir do *Dataset* criado, utilize o método ***addColumn*** para definir as colunas do *Dataset*.



O parâmetro informado no método ***addColumn*** define o nome da coluna adicionada.

**Passo 8**

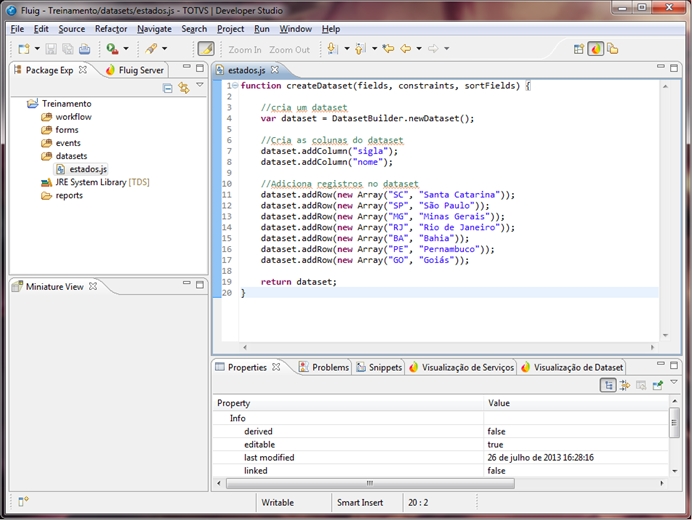
Para adicionar os registros no *Dataset*, utilize o método ***addRow.***



Esse método recebe um *array* como parâmetro. Esse *array* deve conter os valores para cada uma das colunas adicionadas, respeitando a ordem de definição das colunas.

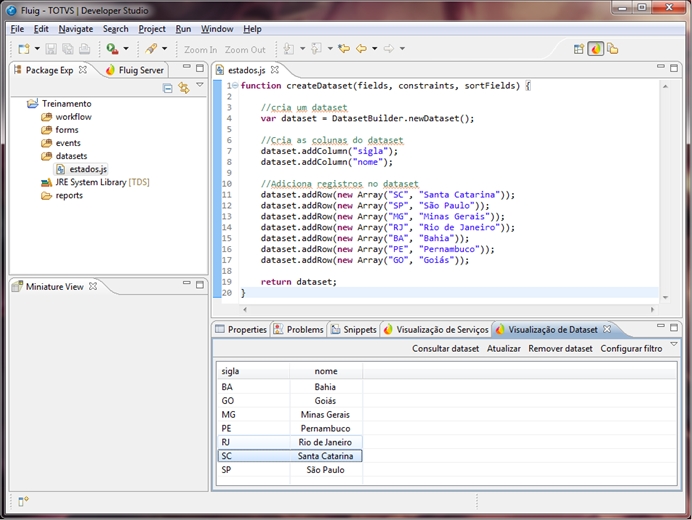
**Passo 9**

E como último passo, defina como retorno da função ***createDataset*** o objeto *dataset* criado.



**Passo 10**

Depois de finalizada a codificação do *Dataset*, é possível exportá-lo para o servidor do **Fluig,** conforme verificado no treinamento ***Studio* I.**



Agora é possível consultar o *Dataset* criado e seus dados.

# Utilizar Dataset

É possível utilizar os ***Datasets*** em vários eventos globais, de processos, de formulário ou mesmo em outros *datasets*.

O acesso ao *Dataset* é realizado por meio da chamada, pelo ***JavaScript DatasetFactory.getDataset***, no qual seus parâmetros são:

* **Nome do Dataset:** Nome do *Dataset* para acesso.
* **Campos:** *Array* com os campos do *Dataset* que serão retornados. Caso seja informado *null*, retorna todos os campos do *Dataset*.
* **Constraints:** *Array* com as condições de busca do *Dataset*. Caso seja informado *null*, retorna todos os registros do *Dataset*.
* **Ordem:** *Array* com os campos para ordenação dos registros do *Dataset*. Caso seja informado *null*, retorna os registros com a ordenação padrão.

A classe *DatasetFactory* é a porta de entrada para o acesso a qualquer *Dataset*, no qual permite a busca de um *dataset* e navegar em seus dados.

# 4.4.1. Via Ponto de Customização

Para a criação de condições de busca é utilizado o método ***DatasetFactory.createConstraint***, no qual seus parâmetros são:

* + **Campo**: Nome do campo que será filtrado.
  + **ValorInicial**: Valor inicial da faixa de valores do filtro.
  + **ValorFinal**: Valor final da faixa de valores do filtro.
  + **Tipo**: Tipo da condição, no qual pode ser:
* ***MUST***: indica que todos os registros do *Dataset* devem satisfazer a esta condição.
* ***SHOULD***: indica que os registros do *Dataset* podem ou não atender à condição. Esse tipo é mais comum caso necessite que um mesmo campo tenha valores A ou B – no qual cada um será uma condição de busca com tipo *SHOULD.*
* ***MUST\_NOT***: indica que nenhum dos registros pode satisfazer a condição.

A seguir visualize uma implementação de exemplo da utilização de um ***Dataset*** em um ***script* de customização.**



**Linha 1**

É criada as condições de busca através do método ***DatasetFactory.createConstraint.***

**Linha 2**

É criado um ***array*** com as condições de busca criadas anteriormente.

**Linha 3**

É criado um ***array*** com os campos para ordenação dos registros do ***Dataset***.

Observe que mesmo sendo apenas um campo é necessário criar um ***array***, pois o método ***DatasetFactory.getDataset*** espera um ***array*** como parâmetro. Esta mesma situação ocorre para as condições de busca, como visto na linha 2, e para os campos de retorno.

**Linha 4**

Na linha 4 é buscado o ***Dataset***, através do método ***DatasetFactory.getDataset.*** Para o parâmetro dos campos de retorno foi informado *null*, o que fará que sejam retornados todos os campos do *Dataset*.

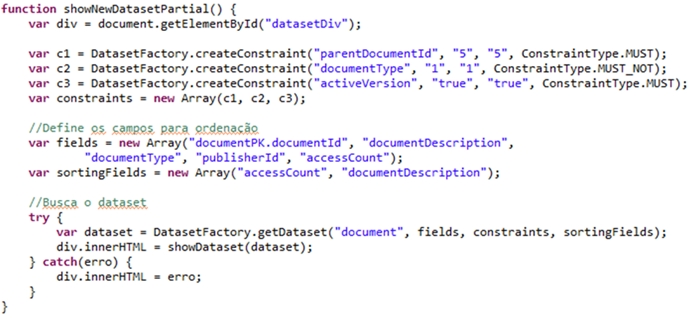
**Linha 5**

Na linha 5 é utilizado o método *getValue* do *Dataset* retornado para pegar os valores presentes nele. Este método possui dois parâmetros, o primeiro é o **índice do registro,** e o segundo o **nome do campo desejado.**

# 4.4.2. Via JavaScript Formulário

É possível acessar os *Datasets* via código *JavaScript* no *HTML* de um formulário utilizando a biblioteca *vcXMLRPC.js.*

Esta biblioteca não é definida automaticamente pelo **Fluig.** Para sua utilização é necessário incluir ela no *HTML* do formulário, adicionando a linha de código conforme disposta.



A forma de utilização via *JavaScript* em formulário é similar ao método via ponto de customização, por meio da utilização do *DatasetFactory.*

Neste exemplo, foi criado uma função *JavaScript* que utiliza o *dataset* document para retornar informações referentes as pastas abaixo da pasta de código 5. As informações de retorno são então adicionadas ao formulário para visualização do usuário.

# 4.4.3. Via Tag Formulário

É possível especificar um *Dataset* em um campo ***Select****.* Quando isto é feito, as opções do *combo* são automaticamente criadas de acordo com os dados do *Dataset.*

Para utilizar um *Dataset* na *tag Select* é necessário informar três propriedades adicionais à *tag:*

• ***Dataset***: determina o *Dataset* utilizado.

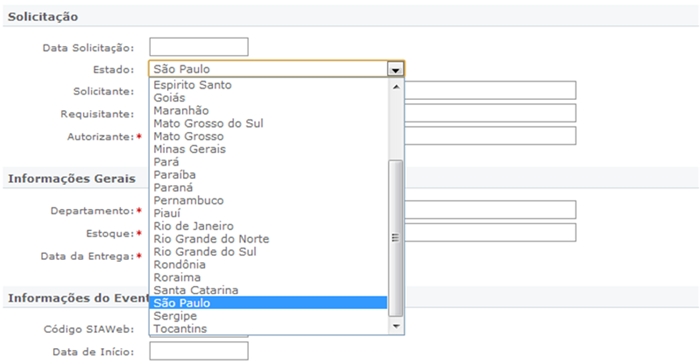
• ***Datasetkey:*** determina qual coluna do *Dataset* será utilizada como valor do campo.

• ***Datasetvalue:*** determina qual coluna do *Dataset* será utilizada como rótulo das opções do combo.



No exemplo de utilização apresentado o campo ***codEstado*** foi configurado para que as opções do combo sejam criadas a partir do *Dataset* estados e como valor do campo será utilizado o valor da coluna sigla, e para o usuário será apresentado o valor da coluna nome.

**Exemplo:**



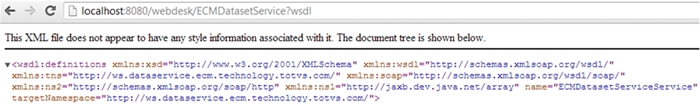
Neste exemplo, foi utilizado um *dataset* customizado de nome estados, via *tag* de formulário, retornando os estados brasileiros como opções do *combobox.*

É imprescindível indexar o repositório do **Fluig** antes do uso de *Dataset* em *tags* de formulários, para que o **Fluig** retorne todos os dados corretamente.

# 4.4.4. Via WebService

O **Fluig** disponibiliza um conjunto de *Web Services* que permitem integrar aplicações de terceiros. Dentre os serviços disponibilizados, está o serviço *DatasetService*, no qual permite realizar o acesso a um *Dataset.*

A forma de uso do *Web Service* varia de acordo com a tecnologia utilizada pela aplicação cliente. Consulte a documentação da tecnologia escolhida pra obter detalhes sobre como acessar *WebServices.*



Para mais informações sobre a utilização dos *WebServices* disponibilizados pelo **Fluig,** acesse a página Utilização de *WebServices,* no endereço **dev.fluig.com**.

# 4.4.5. Guia de Referência

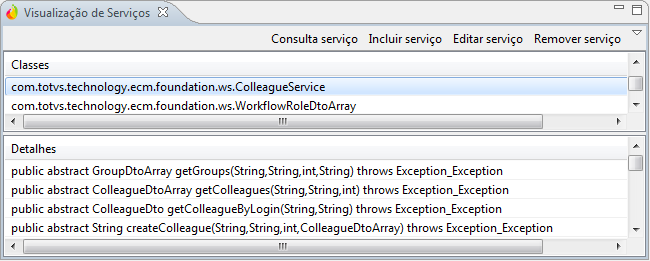
Caso tenha interesse sobre mais informações de customização e utilização de *datasets,* consulte o **Guia de Referência de Customização de Datasets** disponível no diretório de instalação do **Fluig.**

# Criar e Utilizar Serviços

O Serviço é um componente de *software* utilizado para a integração e comunicação entre diferentes aplicações.

A utilização de serviços no **Fluig** permite a troca de informações do **Fluig** com outras aplicações e também para a utilização de funcionalidades disponibilizadas por estas outras aplicações.

A principal forma de integração do **Fluig** é através de *Webservices.* Outra forma de integração disponível é via chamadas ao *Progress Open AppServer* e é indicado para usuários que necessitam realizar integrações com aplicativos desenvolvidos nesta plataforma.



Um *WebService* é uma tecnologia utilizada para a comunicação entre diferentes aplicações e sistemas.

A integração de sistemas via *WebServices* permite que aplicações e sistemas desenvolvidos em plataformas totalmente diferentes possam trocar informações entre si de forma transparente.

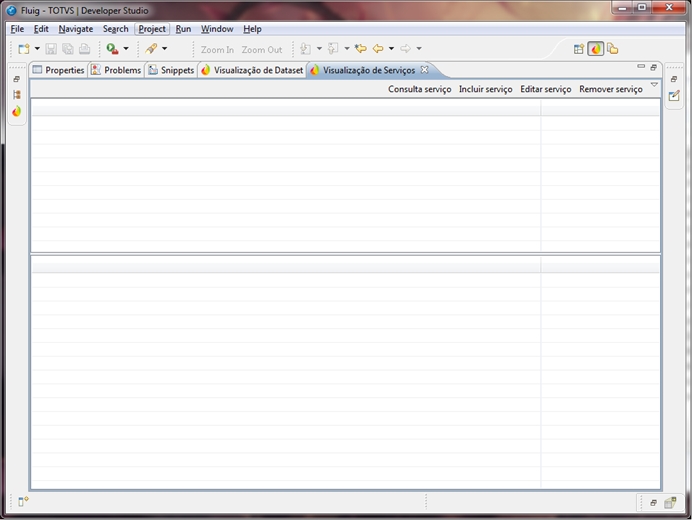
Possui uma linguagem ***WSDL*** baseada em *XML* usada para descrever como acessá-lo e quais as operações ou métodos disponíveis.

# Passo a Passo

A seguir visualize o passo a passo de como **adicionar** e **visualizar** um **Serviço no *Studio*.**

**Passo 1**

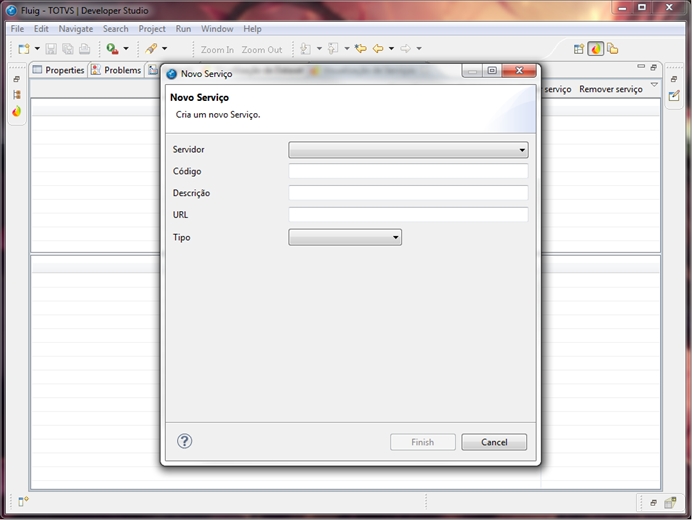
O gerenciamento de serviços é realizado no ***Studio*** através da visão **Visualização de Serviços.**



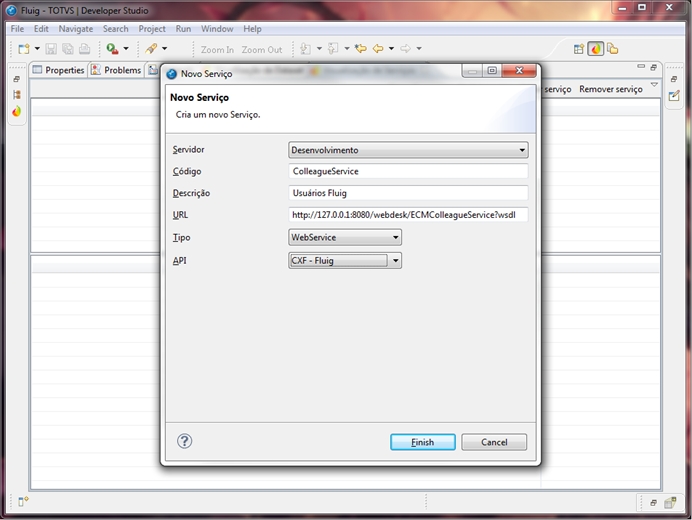
Para incluir um serviço no **Fluig,** clique em **Incluir Serviço.**

**Passo 2**

O assistente **Novo Serviço** é aberto.



**Passo 3**



Neste exemplo é incluído o ***Webservice ColleagueService*** do próprio **Fluig** como um serviço.

No campo **Servidor** selecione o servidor **Fluig** onde será adicionado o serviço.

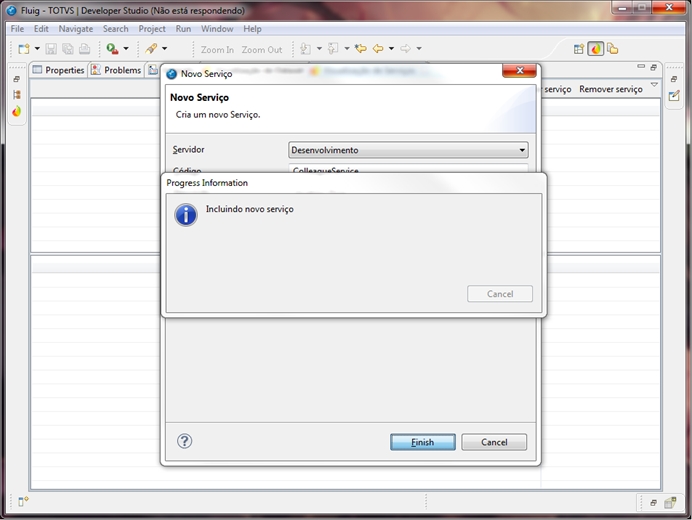
Informe o código e a descrição do serviço. O código informado não pode ser igual ao de um **Serviço** existente.

No campo *URL* informe o endereço do serviço.

Selecione ***Webservice*** como tipo e clique em ***Finish*** para finalizar a inclusão do serviço.

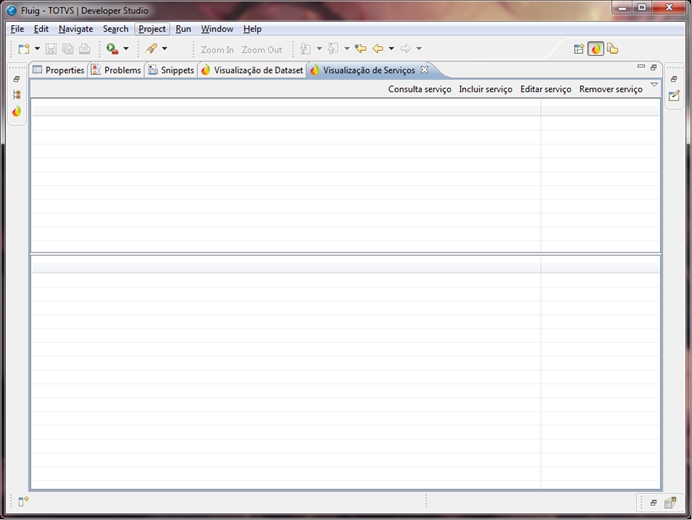
**Passo 4**

Aguarde a mensagem de informação de progresso.



**Passo 5**

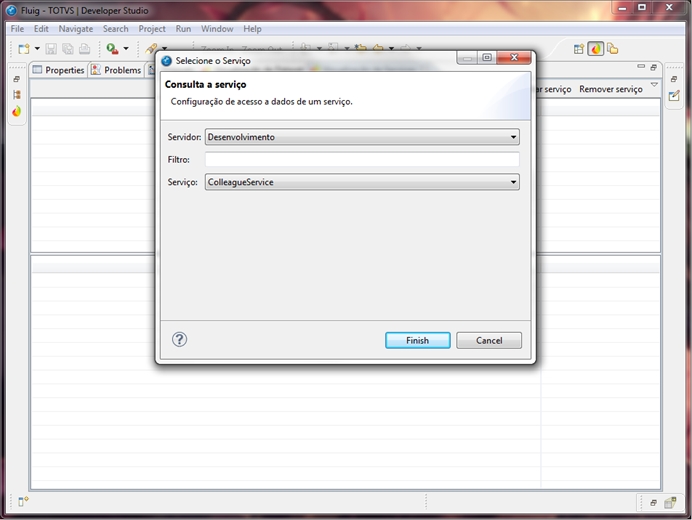
O serviço foi adicionado.



Para consultar um serviço clique em **Consulta Serviço.**

**Passo 6**

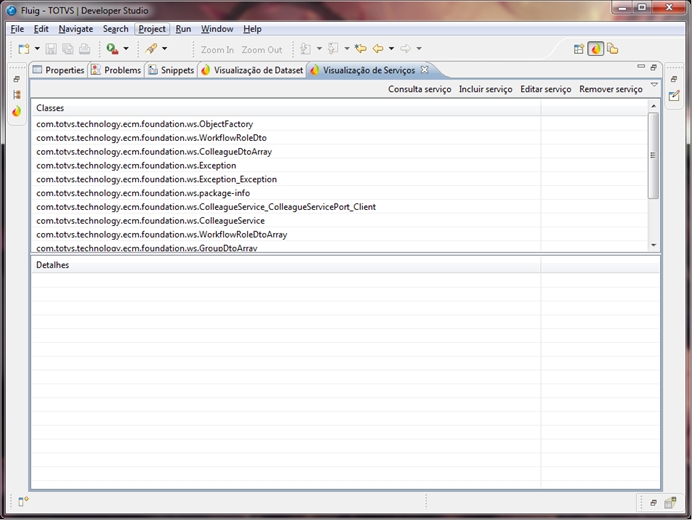
O assistente **Consulta a Serviço** é aberto.



Selecione o servidor e o serviço desejado e clique em ***Finish*.**

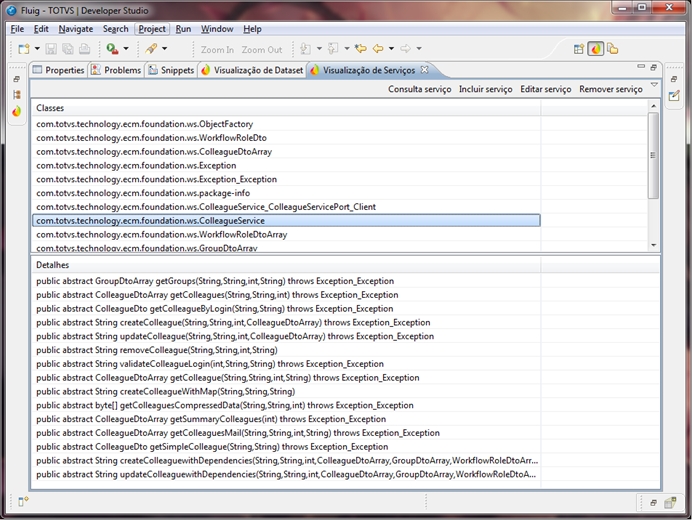
**Passo 7**

As classes do serviço são apresentadas.



**Passo 8**

Ao clicar em uma das classes listadas são apresentados os detalhes dela, sendo possível conhecer os métodos disponíveis para utilização da classe em questão.



# Implementação

Os serviços adicionados no **Fluig** podem ser instanciados e utilizados nos pontos onde o **Fluig** permite customização utilizando-se *JavaScript*, como nos *scripts* para eventos de processos, eventos de formulário ou *datasets*.

O código demonstrado apresenta uma implementação de exemplo do uso de um serviço em um *script* de customização.



**Linha 1**

Na **linha 1** é carregado o serviço cadastrado no **Fluig**, através do método ***ServiceManager.getService*.**

O nome passado como parâmetro deve ser o mesmo nome utilizado quando cadastrado o serviço.

**Linha 2**

Na **linha 2** é utilizado o método ***getBean*** do serviço carregado para retornar um utilitário para acesso às classes do serviço.

**Linha 3**

Na **linha 3** é utilizado o método ***instantiate*** do utilitário para instanciar a classe que possuir o localizador do serviço.

As classes que podem ser instanciadas por este método são as classes listadas quando é realizada a consulta de um serviço, como vimos anteriormente neste treinamento.

**Linha 4**

Na **linha 4** é utilizado o método responsável em **instanciar** o serviço.

**Linha 5**

Na **linha 5** é utilizado o serviço. Os métodos que podem ser utilizados são listados quando é realizada a consulta de uma classe do serviço, como vimos anteriormente neste treinamento.

Possuindo o objeto utilitário do serviço instanciado, as classes que devem ser instanciadas e os métodos que devem ser invocados dependem de cada *WebService* utilizado, e deve-se recorrer à sua documentação para mais informações.

Para mais informações sobre a criação e utilização de serviços, acesse a página Integração com Aplicativos Externos, no endereço dev.fluig.com.

# Conclusão

Assim, você terminou o treinamento ***Studio* IV** do Curso ***Studio* do Fluig**.

Neste treinamento você conheceu mais sobre:

* Utilizar *Logs.*
* Eventos Globais.
* Customizar Evento.
* *Dataset.*
* Tipos de *Dataset.*
* Consultar *Dataset.*
* *Dataset* Customização.
* Utilizar *Dataset.*
* Criar e Utilizar Serviços.
* Adicionar e Visualizar um serviço no *Studio.*