

Contents

- [Custom Methods](#)
- [Configure Method](#)
 - [getTransfer\(\)](#)
- [Notes](#)

Custom Methods

It is possible to define custom methods on a [TransferObject](#) within the configuration file for Transfer.

For every 'Function' element that is defined in the [Transfer Configuration File](#), a corresponding ColdFusion method is applied to the generated TransferObject.

For example, given the following definition on a TransferObject:

```
<function name="setNewProperty" access="public" returntype="numeric">
  <argument name="NewProperty" type="string" required="true"/>
  <body>
    <![CDATA[<cfset instanceNewProperty = argumentsNewProperty]]>
  </body>
</function>
```

Will result in the ColdFusion method being written:

```
<cffunction name="setNewProperty" access="public" returntype="numeric" output="false">
  <cfargument name="NewProperty" type="string" required="true"/>
  <cfset instanceNewProperty = argumentsNewProperty>
</cffunction>
```

It should also be noted that the full array of TransferObject functions are available at this scope.

Care should be taken not to overwrite any of the generated methods and/or inner properties of the TransferObject, as errors may occur.

Configure Method

Since the `init()` method is automatically generated for a TransferObject, and cannot be overwritten, you are able to define an optional method named 'configure' that is called immediately after the `init()` method has finished processing.

If the TransferObject called is being requested from the database, `init()` is called *before* the properties of the Object are set from the record values.

The `configure()` method must be named 'configure' and can have no required arguments.

Example:

```
<function name="configure" access="private" returntype="void">
  <body>
    <![CDATA[<cfset setPropertyValue("defaultValue")]]>
  </body>
</function>
```

getTransfer()

Custom methods can call 'getTransfer()' within their bodies to get access to the [transfer.com.Transfer](#) object.

Notes

Its should be noted that similar functionality can be accomplished through the use of [Decorators](#)

📁 Categories:

- [Configuration](#)
- [Decorators](#)