BASIC TUTORIAL: ACCELEO

Preparation

Installation

- Install the Acceleo tool
  - Help → Eclipse → Marketplace → Acceleo
- Click on Go
- Click on Install

Import an ecore

- Create a new EMF project.
  - New Project → Other -→ Empty EMF Model
- Import an ecore into model folder
  - Import the sample ecore
- Create code from ecore and called it purchase.genmodel
  - New → Other →EMF Generator Model
- Use genmodel to generate the model code
  - Click right button on the purchase.genmodel file → Generate Model Code
- Select the first entity in the ecore (System) and create dynamic instance (with extension .xmi)
- Create purchase orders and items inside that instance.

Acceleo Project

Creation of a new project

- Create a new Acceleo Project with name PurchaseAcceleo
- New Project → Acceleo Project
  - Name: PurchaseAcceleo
- Click on the green cross
  - Click on Runtime Version
  - Select the purchase.ecore metamodel
- Select the System in Type
- Click on Generate file and Main template
- Click on Finish

Modify the following functions in Generate.java

- In the comments just before `Acceleo.Generate.registerPackages`

```
@generated NOT
```

- At the end of `Acceleo.Generate.registerPackages`

```
if(!isInWorkspace(PurchasePackage.class))
	resourceSet.getPackageRegistry().put(PurchasePackage.eNS_URI,
	PurchasePackage.eINSTANCE);
```

- Note that you need to import the implementation of `PurchasePackage`
- In the comments just before `Acceleo.Generate. registerResourceFactories`

```
@generated NOT
```

- At the end of `Acceleo.Generate. registerResourceFactories`

```
resourceSet.getResourceFactoryRegistry().getExtensionToFactoryMap()
.put("xmi", new XMIResourceFactoryImpl());
```

- Note that you need to import this library:

```
import org.eclipse.emf.ecore.xmi.impl.XMIResourceFactoryImpl;
```

- Check the metamodel path in generate.mtl (it must have a relative path)

Run the Acceleo program for the first time

- Create a folder called outputs
- Select the run configuration
  - Run → Run Configurations
- Select the metamodel
- Select input model (*.xmi)
- Select output folder
- Run as Java Application

Create the first page

- Type the code below in generate.mtl

```
[module generateItem('/ejemplo/purchase.ecore')]

[template public generateElement(aSystem : System)]
[comment @main/]
[file ('output.html', false, 'UTF-8')]
<html>
  <body>
       <div>
       <h2>List of Orders</h2>
[for (anOrder: PurchaseOrder | aSystem.orders)]
   <li>Order billed to [anOrder.billTo /] and shipped to [anOrder.shipTo /]</li>
[/for]
  </div>
  </body>
</html>
[/file]
[/template]
```

- Run to see the result

Generate pages per order

- Create a new package called src/PurchaseAcceleo.files
- Create in this package a new Acceleo Module File
  - Module Name: PurchaseOrderFile
  - Select the ecore metamodel
  - Select Type: PurchaseOrder
  - Click on GenerateFile
- Type the following code

```
[comment encoding = UTF-8 /]
[module PurchaseOrderFile('/ejemplo/purchase.ecore')]


[template public generate(aPurchaseOrder : PurchaseOrder)]
[file (aPurchaseOrder.billTo.concat('.html'), false, 'UTF-8')]
<html>
  <body>
  <h3>Order billed to [ aPurchaseOrder.billTo /]</h3>
  <h3>Shipped to [ aPurchaseOrder.shipTo /]</h3>
  <p>Items of the order:</p>
  <ul>
[for (item : Item | aPurchaseOrder.items)]
       [ item.productName /]
[/for]
  </ul>
  </body>
</html>
[/file]
[/template]
```

- Modify generate.mtl with the following code

```
[module generateItem('/ejemplo/purchase.ecore')]

[import PurchaseAcceleo::files::PurchaseOrderFile /]

[template public generateElement(aSystem : System)]
[comment @main/]
[file ('output.html', false, 'UTF-8')]
<html>
  <body>
    <div>
      <h2>List of Orders</h2>
[[for (order : PurchaseOrder | aSystem.orders)]
        <li>Order billed to <a href="[order.billTo.concat('.html')/]">[order.billTo
/]</a> and shipped to [order.shipTo /]</li>[order.generate() /]
[/for]
  </div>
  </body>
</html>
[/file]
[/template]
```

- Run it

Generate information about the items

- Create the package src/PurchaseAcceleo.common
- Create in this package a new Acceleo Module File (Right Click)
  - Module Name: generateItem
  - Select the ecore metamodel
  - Select Type: Item
- Type the following code

```
[module generateItem('/ejemplo/purchase.ecore')]

[template public generateElement(anItem : Item)]
       <li>[anItem.quantity/] [anItem.productName/] - [anItem.price/] euros.</li>
[/template]
```

- Modify PurchaseOrderFile.mtl with the following code

```
[comment encoding = UTF-8 /]
[module PurchaseOrderFile('/ejemplo/purchase.ec')]

[import PurchaseAcceleo::common::generateItem /]

[template public generate(aPurchaseOrder : PurchaseOrder)]
[file (aPurchaseOrder.billTo.concat('.html'), false, 'UTF-8')]
<html>
  <body>
  <h3>Order billed to [ aPurchaseOrder.billTo /]</h3>
  <h3>Shipped to: [ aPurchaseOrder.shipTo /]</h3>
  <p>Items of the order:</p>
  <ul>
[for (item : Item | aPurchaseOrder.items)]
    [ item.generateElement() /]
[/for]
  </ul>
  </body>
</html>
[/file]
[/template]
```

- Run it

Generate more information about the items

- In the package src/PurchaseAcceleo.common
- Create a new Acceleo Module File (Right Click)
  - Module Name: generateDiscountedItem
  - Select the ecore metamodel
  - Select Type: DiscountedItem
- Type the following content

```
[module generateDiscountedItem('/ejemplo/purchase.ecore')]

[import PurchaseAcceleo::queries::ItemQueries /]

[template public generateElement(aDiscountedItem : DiscountedItem)]
<li>Offer [aDiscountedItem.discount/]%!! [aDiscountedItem.quantity/]
[aDiscountedItem.productName/] - [aDiscountedItem.price/] euros.</li>
[/template]
```

- Include this template in PurchaseOrderFile.mtl
- Run it.

Generate a query

- Create the package src/PurchaseAcceleo.queries
- Create in this package a new Acceleo Module File (Right Click)
  - Module Name: ItemQueries
  - Select the ecore metamodel
  - Select Type: Item
  - Write the following content

```
[comment encoding = UTF-8 /]
[module ItemQueries('/ejemplo/purchase.ecore')]

[query public totalPrice(anItem : Item) : Real =
	anItem.quantity * anItem.price /]

[template public kindOfProduct(anItem : Item)]
[if (anItem.price < 10)]
	Cheap product
[elseif( anItem.price < 20 )]
	Regular product
[else]
	Expensive product
[/if]
[/template]
```

- Modify generateDiscountedItem.mtl with the following code

```
[module generateDiscountedItem('/ejemplo/purchase.ecore')]

[import PurchaseAcceleo::queries::ItemQueries /]

[template public generateElement(aDiscountedItem : DiscountedItem)]
<li>Offer [aDiscountedItem.discount/]%!! [aDiscountedItem.quantity/]
[aDiscountedItem.productName/] - [aDiscountedItem.price/] euros. Total:
[aDiscountedItem.totalPrice()/] euros</li>
[/template]
```

- Modify generateItem.mtl with the following code

```
module generateItem('/ejemplo/purchase.ecore')]

[import PurchaseAcceleo::queries::ItemQueries /]

[template public generateElement(anItem : Item)]
	<li>[anItem.quantity/] [anItem.productName/] - [anItem.price/] euros
([anItem.kindOfProduct()/]). Total: [anItem.totalPrice()/] euros </li>
[/template]
```

- Run it.