Language editor generation
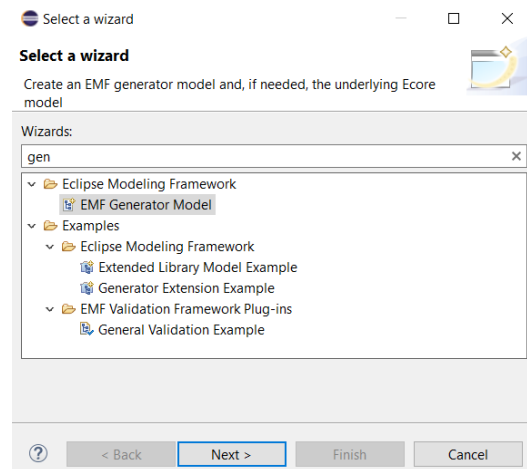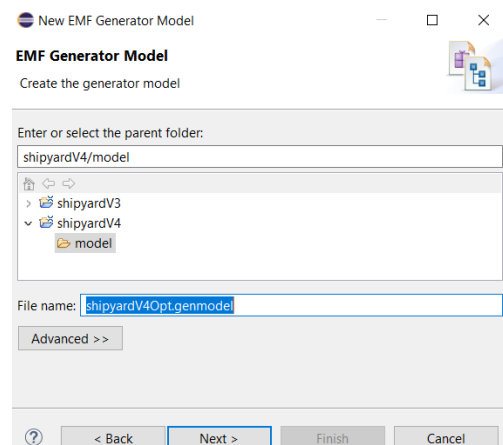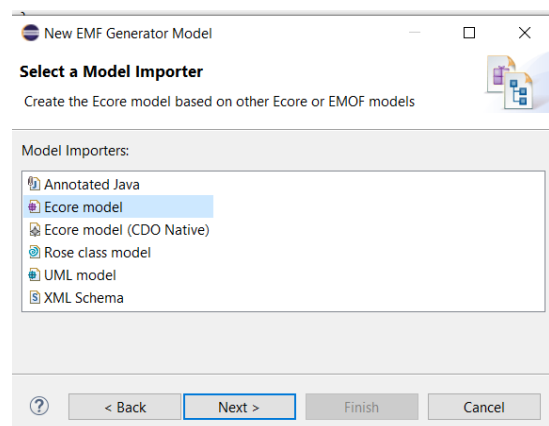
1.	Click on the /model/<name>Opt.ecore file, select *new -> other -> EMF Generator Mode*l.
Click *Next*.

2.	Fill the *File name* field as suggested.
 Click *Next*

3.	Choose *Ecore model* and click *Next*

4.   Click *Load* and the *Next*

----
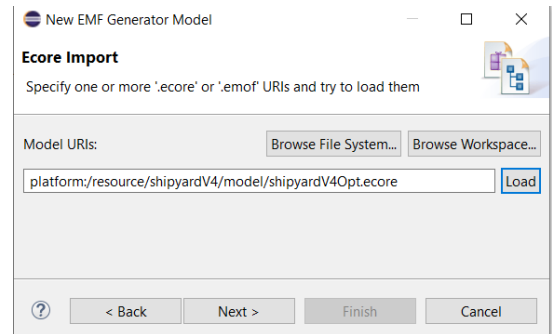
5.   Select packages as in the figure below and click *Finish*

----
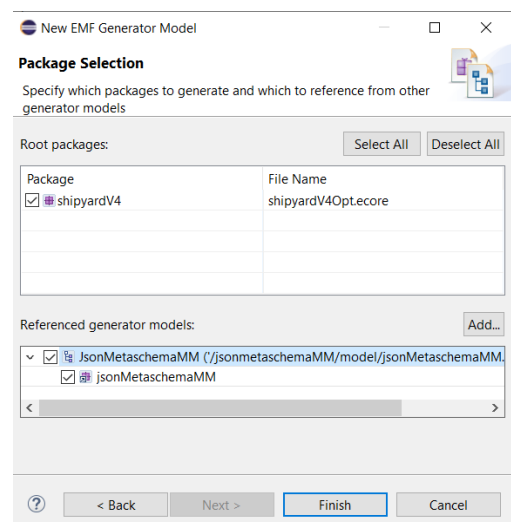
6.   It will generate a file with extension *.genmodel* under the folder model . Open it. A tree will be displayed. Right click it and then select *Generate Model Code*
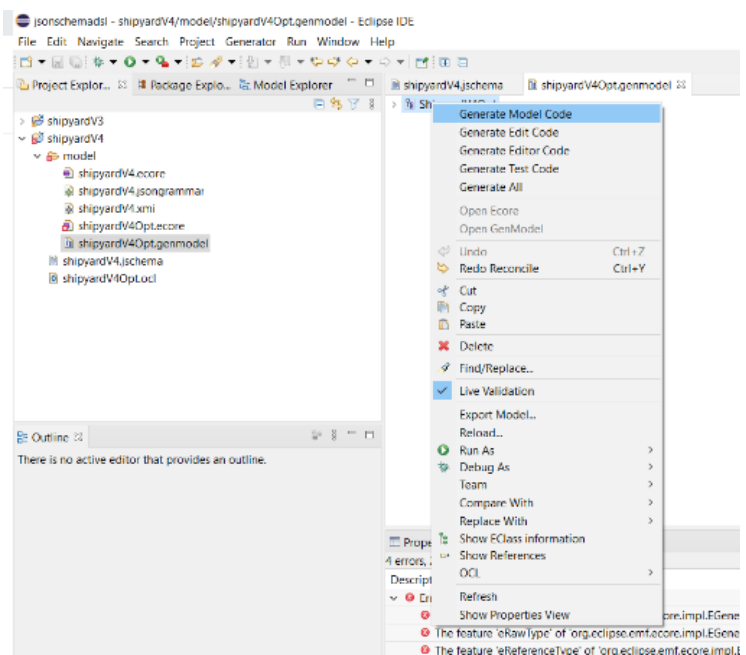
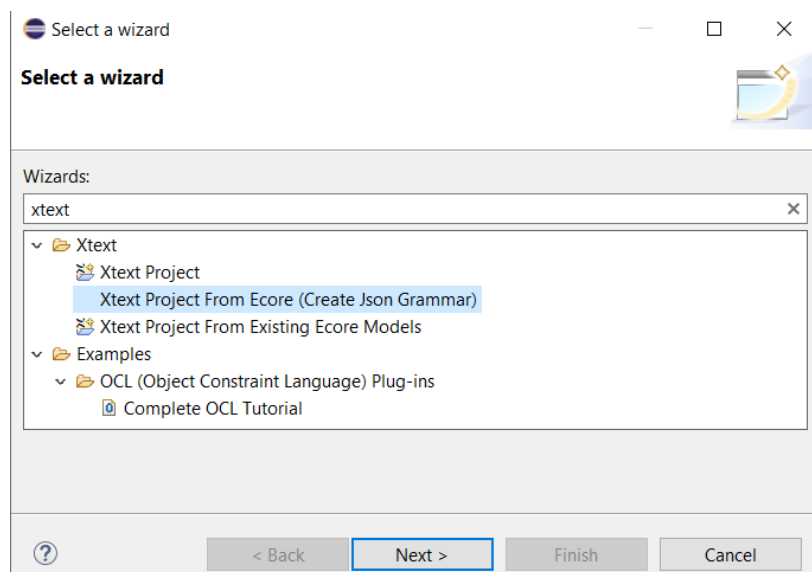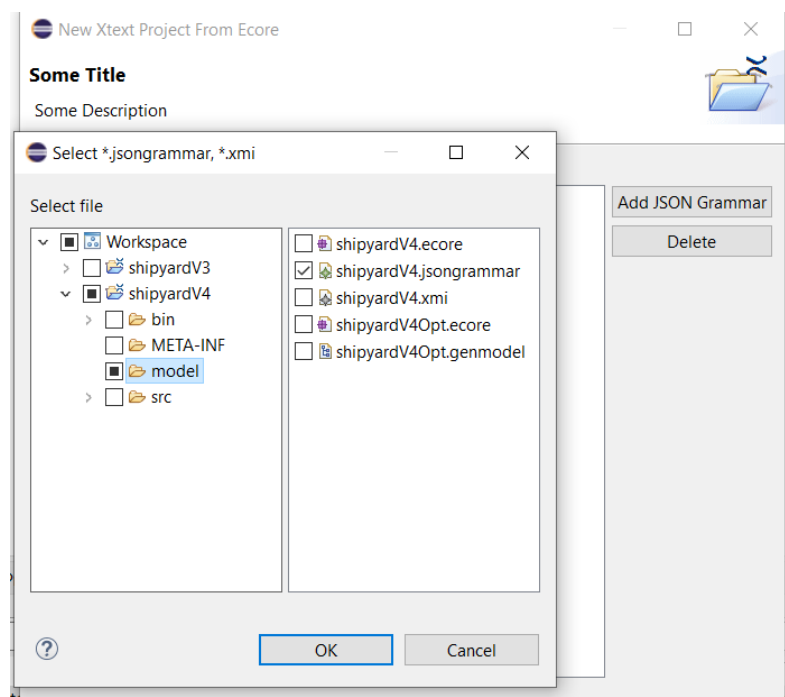
7. A new folder src with code will be generated.

```
shipyardV4
  src
    shipyardV4
    shipyardV4.impl
    shipyardV4.util
  JRE System Library [JavaSE-13]
  Plug-in Dependencies
  META-INF
  model
    shipyardV4.ecore
    shipyardV4.jsongrammar
    shipyardV4.xmi
    shipyardV4Opt.ecore
    shipyardV4Opt.genmodel
```
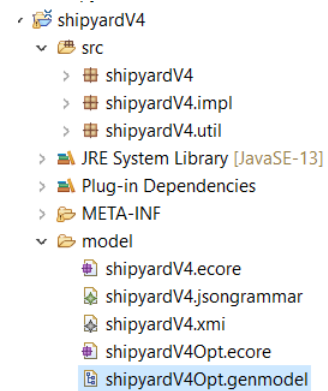
8. Click on *File -> New ->Othe*r in the Eclipse top menu. Choose *Xtext Project From Ecore (Create json Grammar)*.
Click *Next*.

Select a wizard

Select a wizard

Wizards:

```
xtext                                              ×
```

```
Xtext
  Xtext Project
  Xtext Project From Ecore (Create Json Grammar)
  Xtext Project From Existing Ecore Models
Examples
  OCL (Object Constraint Language) Plug-ins
    Complete OCL Tutorial
```

| < Back | Next > | Finish | Cancel |

----

9. Click Add JSON Grammar, and select the file *.jsongrammar* under the folder model of your project.
Click *OK* .

New Xtext Project From Ecore

Some Title

Some Description

Select *.jsongrammar, *.xmi

Select file

```
Workspace                        shipyardV4.ecore
  shipyardV3                   ☑ shipyardV4.jsongrammar
  shipyardV4                     shipyardV4.xmi
    bin                          shipyardV4Opt.ecore
    META-INF                     shipyardV4Opt.genmodel
    model
    src
```

Add JSON Grammar

Delete

| OK | Cancel |

----

10.   Click *Add*.
Select the file with extension *.genmodel* generated in the previous wizard. Click *OK*.

----
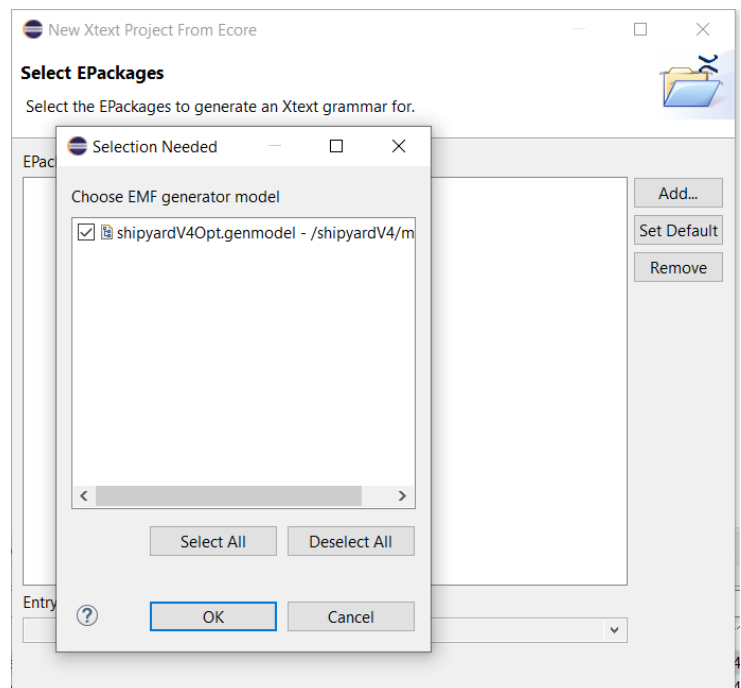
11.   In the *Entry rule* select the root of your metamodel (It has the name of your project plus *Root* Suffix). Click *Next*. **DO NOT** click Finish.

----
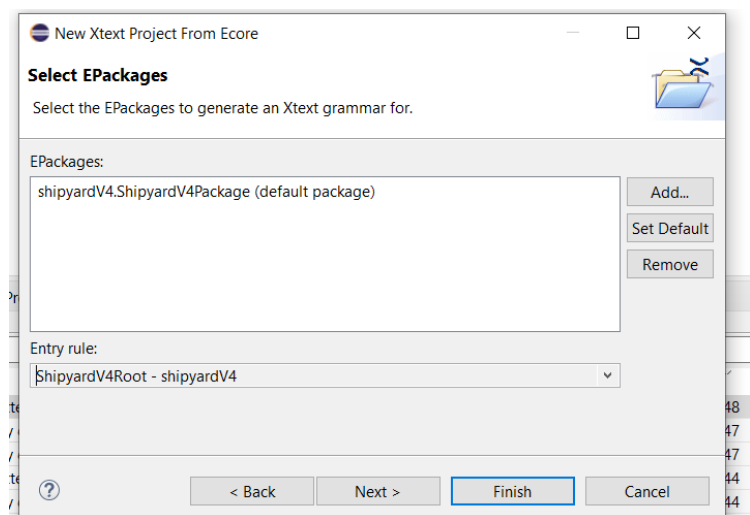
12.   Choose the proper name for your Editor. Above all take care at the *Extensions* field: This will be the extensions of the files that the editor will recognize.. Make sure to use JAVA 11.0.10 or newer..
Click *Next*, **DO NOT** click *Finish*.

----

13.   Fill the Fields as in the picture. But DO NOT check Web Integration: the picture is wrong.
Click *Finish*.
Wait until Eclipse has finished its job.

----

14. You will see a new project created. Open its folders as in the picture until you find a file with extension *.xtext*.

----

15. Open the file. Probably you will see a warning like in the picture. Move your mouse over the warning and apply the suggestion. Save the file if required.

```
1  // automatically generated by Xtext JSON
2  grammar at.jku.bise.shipyardV4.ShipyardV4 with org.eclipse.xtext.common.Terminals
3
4  import "http://jku.bise.at/shipyardV4"
5  import "http://www.eclipse.org/emf/2002/Ecore" as ecore
6  import "http://at.jku.bise/jsonMetaschemaMM" as jsonMetaschemaMM
7
8  //EClass ShipyardV4Root
9  ShipyardV4Root returns ShipyardV4Root:
10             //EReference is not a keyword
11             shipyardV4Root=Shipyard
```

----

16. Right click the file with extension *.xtext* and then click *Run As* and *1. Generate Xext Artifacts*.
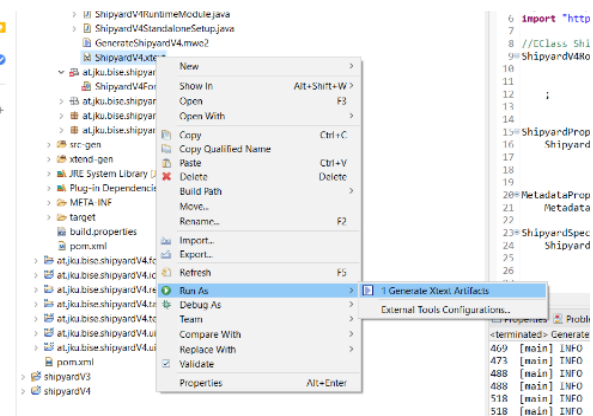
In the console the following INFO will be printed

```
Problems  @ Javadoc  Declaration  Console ⊠
<terminated> GenerateShipyardV4.mwe2 [Mwe2 Launch] C:\Program Files\Java\jdk-11.0.1\bin\javaw.exe  (9 may. 2021 22:22:44 – 22:22:53)
0     [main] INFO  text.xtext.generator.XtextGenerator   - Initializing Xtext generator
12    [main] INFO  lipse.emf.mwe.utils.StandaloneSetup    - Adding generated EPackage 'org.eclipse.xtext.common.types.TypesPackage'
261   [main] INFO  lipse.emf.mwe.utils.StandaloneSetup    - Registering project at.jku.bise.shipyardV4 at 'file:/C:/Users/Alessandro.cola
263   [main] INFO  lipse.emf.mwe.utils.StandaloneSetup    - Registering project at.jku.bise.shipyardV4.tests at 'file:/C:/Users/Alessandr
265   [main] INFO  lipse.emf.mwe.utils.StandaloneSetup    - Registering project at.jku.bise.shipyardV4.ide at 'file:/C:/Users/Alessandro.
266   [main] INFO  lipse.emf.mwe.utils.StandaloneSetup    - Registering project at.jku.bise.shipyardV4.ui at 'file:/C:/Users/Alessandro.c
267   [main] INFO  lipse.emf.mwe.utils.StandaloneSetup    - Registering project at.jku.bise.shipyardV4.ui.tests at 'file:/C:/Users/Alessa
276   [main] INFO  lipse.emf.mwe.utils.StandaloneSetup    - Using resourceSet registry. The registered Packages will not be registered in
549   [main] INFO  clipse.emf.mwe.utils.GenModelHelper    - Registered GenModel 'http://www.eclipse.org/Xtext/Xbase/XAnnotations' from 'p
552   [main] INFO  clipse.emf.mwe.utils.GenModelHelper    - Registered GenModel 'http://www.eclipse.org/xtext/xbase/Xtype' from 'platform
564   [main] INFO  clipse.emf.mwe.utils.GenModelHelper    - Registered GenModel 'http://www.eclipse.org/xtext/xbase/Xbase' from 'platform
564   [main] INFO  clipse.emf.mwe.utils.GenModelHelper    - Registered GenModel 'http://www.eclipse.org/xtext/common/JavaVMTypes' from 'p
596   [main] INFO  clipse.emf.mwe.utils.GenModelHelper    - Registered GenModel 'http://jku.bise.at/shipyardV4' from 'platform:/resource/
596   [main] INFO  clipse.emf.mwe.utils.GenModelHelper    - Registered GenModel 'http://at.jku.bise/jsonMetaschemaMM' from 'platform:/res
955   [main] INFO  text.xtext.generator.XtextGenerator    - Generating at.jku.bise.shipyardV4.ShipyardV4
5650  [main] INFO  text.xtext.generator.XtextGenerator    - Generating common infrastructure
5679  [main] INFO  .emf.mwe2.runtime.workflow.Workflow     - Done.
```
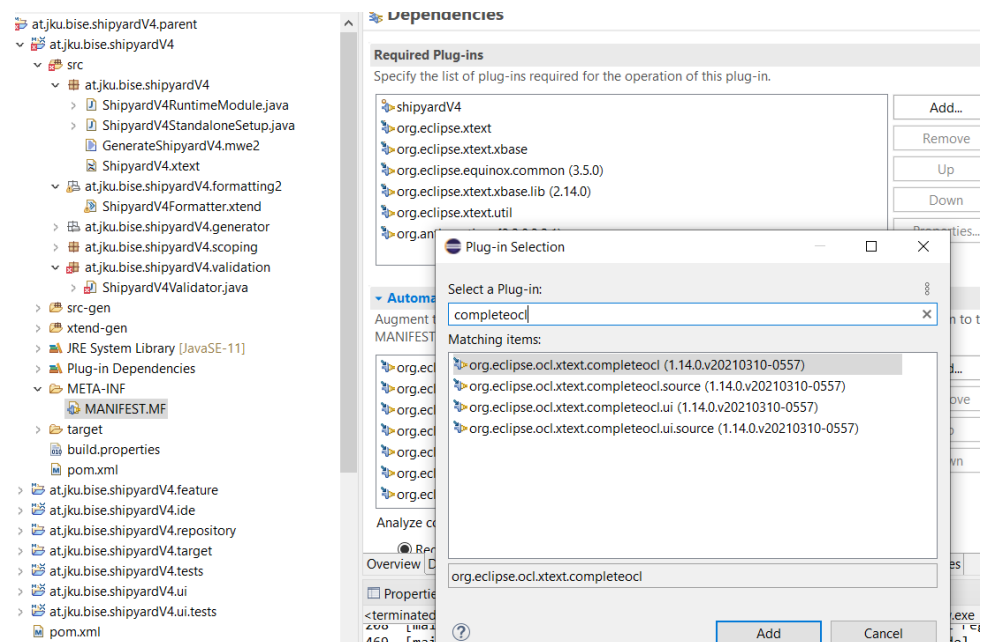


17. Open the file *<YourProjectName>Formatter.xtend*.

If there is a compilation error. Move your mouse over it and apply the suggestion to change the to a name with the first character in lower case.

----
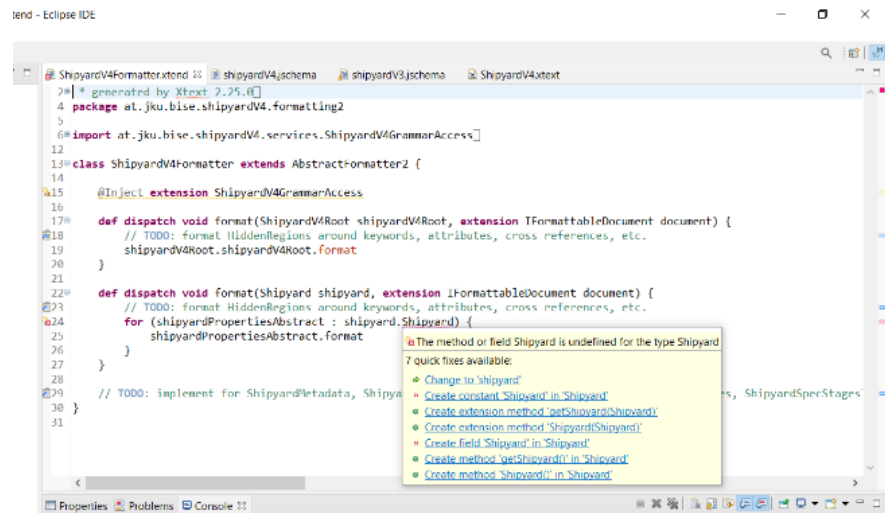
18.   Open the file MANIFEST.MF like in the picture. Select the *Dependencies* tab. Then click *Add*. Select the plugin like in the picture. Click *Add*. Save the file.

----

19.   Open the file like in the picture below.
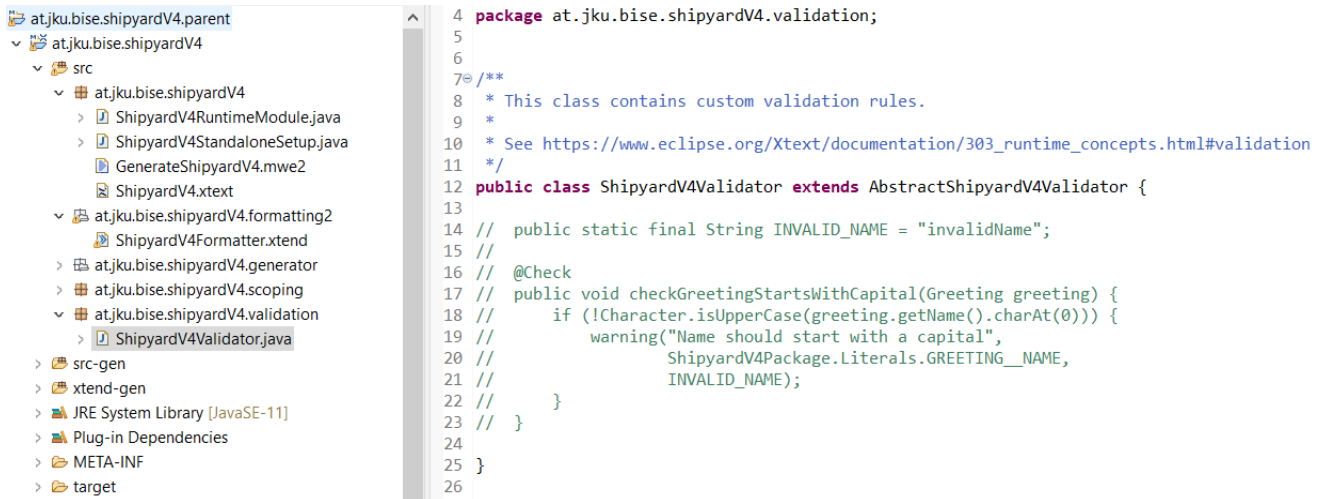
20.   Add the following code snippet:

```
public void register(EValidatorRegistrar registrar) {
     super.register(registrar);
     ShipyardV4Package ePackage = ShipyardV4Package.eINSTANCE;


     URI basicOclURI = URI.createPlatformPluginURI("/shipyardV4/shipyardV4Opt.ocl", true);
     registrar.register(ePackage, new CompleteOCLEObjectValidator(ePackage, basicOclURI));

          }
```
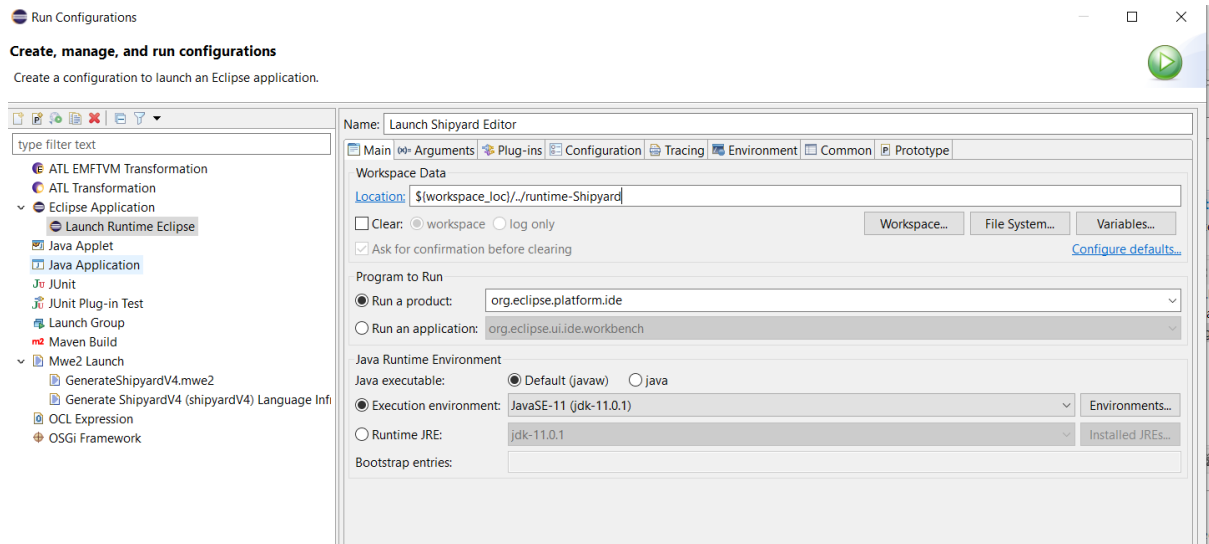
This code is for the case of a project named *shipyardV4*. Take care of the imports. The result of the whole class should be something like the picture below.
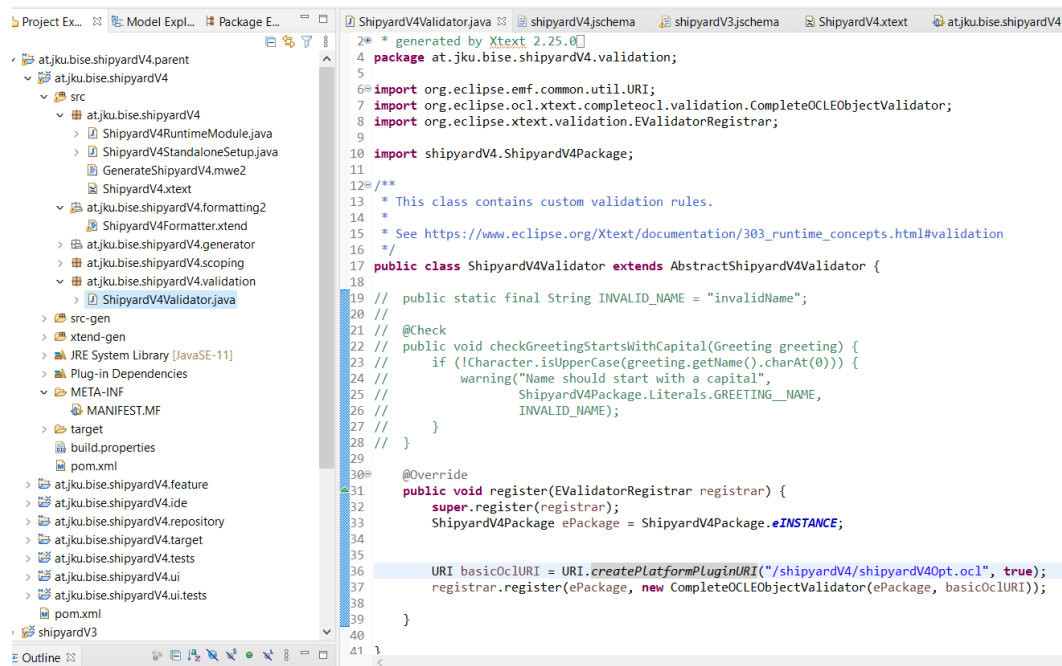

21.   Execute un project clean build.
22.   Regenerate the xtext artifacts
23.   Open the Java Perspective.

24.   In the eclipse top menu click *Run Configurations*, and create your *language editor launcher* like in the picture.

----

25. Run it. Wait that a new eclipse is launched

26. Create your simple plain project; create a file with the extension you chose in the language editor creation. You will be asked if you want to convert the project to an XText project. Answer YES. Open you file
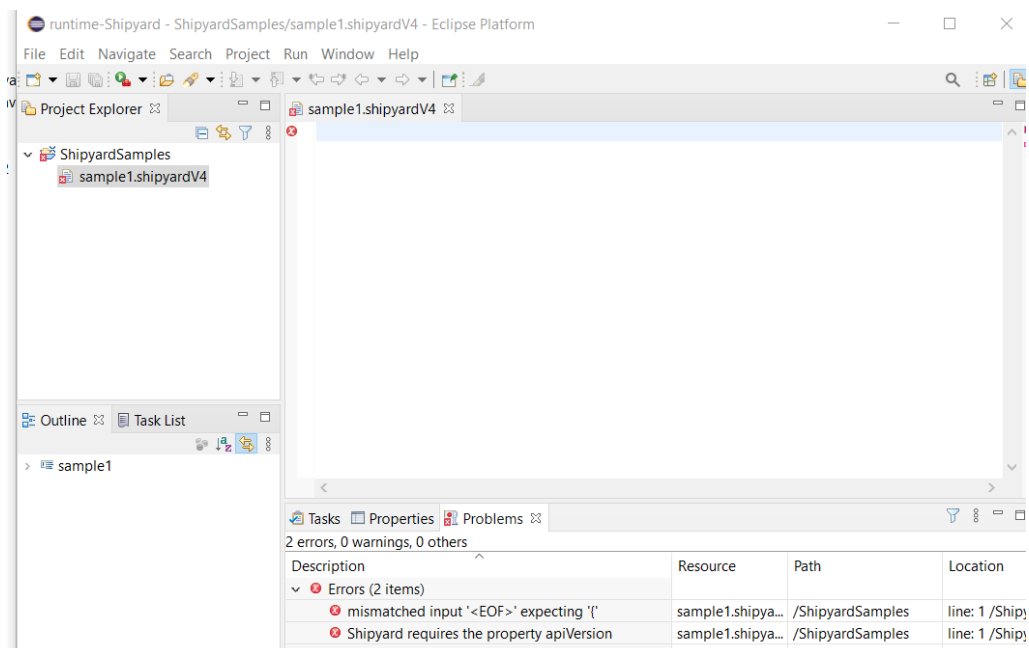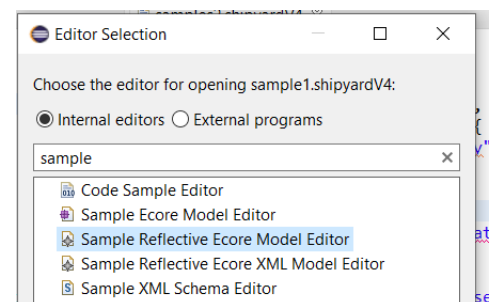

27.   An example:

----
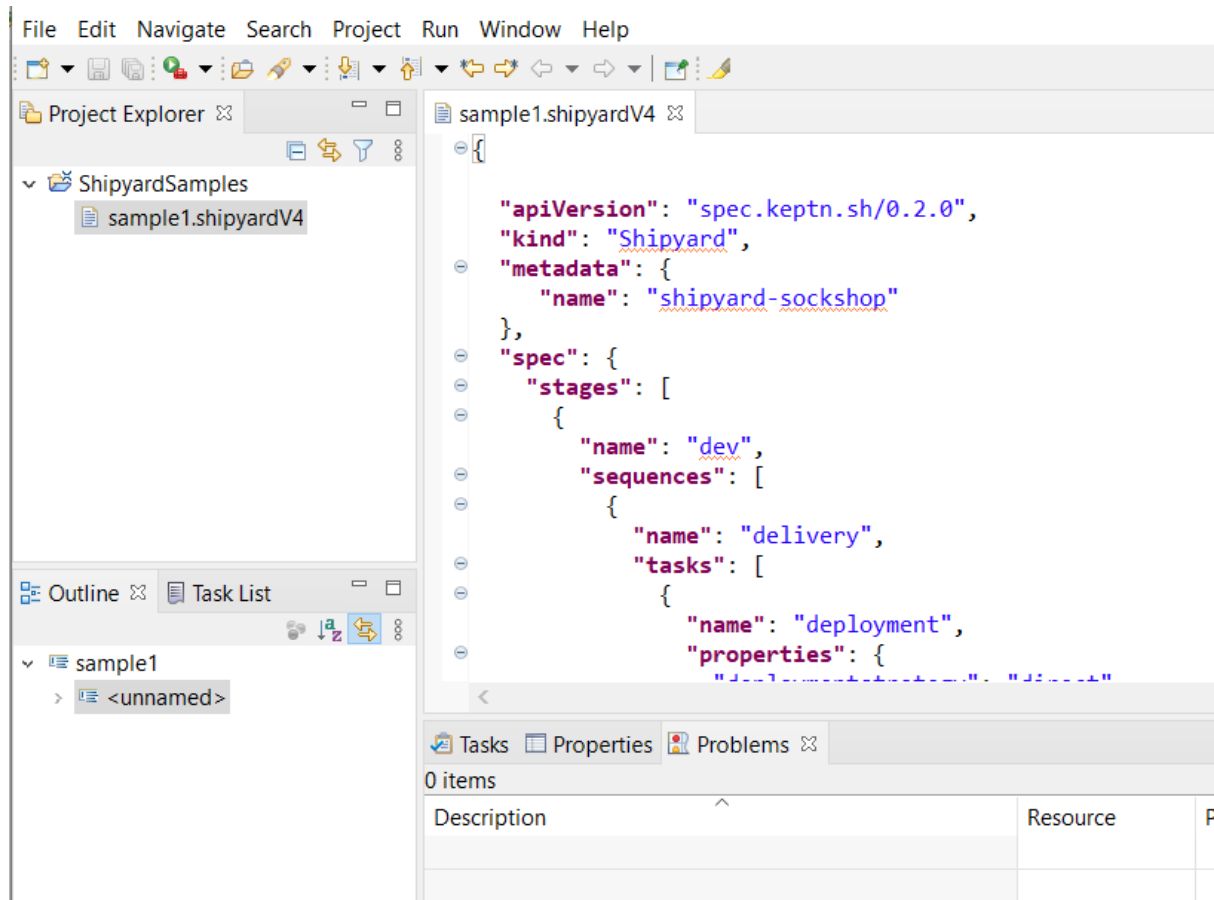
28.   Right click on the file with the extension that you have created (e.g., sample1.shipyardV4) and select *open with -> other*. Choose then the *Sample Reflective Ecore Model Editor* and click *OK*. Wait some second and you will see the tree editor for your file, that is a model.
Open the view properties, to see more details when you select an element of the tree.
Changes applied to the tree are reflected in the file json style and vice versa.

Left Editor Panel

Tab: `sample1.shipyardV4`

```json
1 {
2
3   "apiVersion": "spec.keptn.sh/0.2.0",
4   "kind": "Shipyard",
5   "metadata": {
6       "name": "shipyard-sockshop"
7   },
8   "spec": {
9     "stages": [
10        {
11          "name": "dev",
12          "sequences": [
13            {
14              "name": "delivery",
15              "tasks": [
16                {
17                  "name": "deployment",
18                  "properties": {
19                    "deploymentstrategy": "
20                }
```

Tooltip: `ShipyardSamples/sample1.shipyardV4`

Right Tree Panel

Tab: `sample1.shipyardV4`

- platform:/resource/ShipyardSamples/sample1.shipyardV4
  - Shipyard V4 Root
    - Shipyard
      - Shipyard Api Version spec.keptn.sh/0.2.0
      - Shipyard Kind Shipyard
      - Shipyard Metadata
        - Metadata
          - Metadata Name shipyard-sockshop
      - Shipyard Spec1

Bottom Panel

Tabs: Tasks | Properties | Problems

Property	Value
Name	shipyard-sockshop