



Overview R&D

7 December 2023

Koen, Frédéric, Jorren, Jan





nsx

NORMALIZED SYSTEMS

1. **Angular Expanders**
2. **Enum Types**
3. **Mapping: Imports**
4. **Expanders Assert**
5. **ProcessAutomation**
6. **IntelliJ Plugin**
7. **Java 21**
8. **REST Expanders**
9. **QuerySearch Expanders**
10. **JEE stack housekeeping**
11. **Planned features for 2024H1**

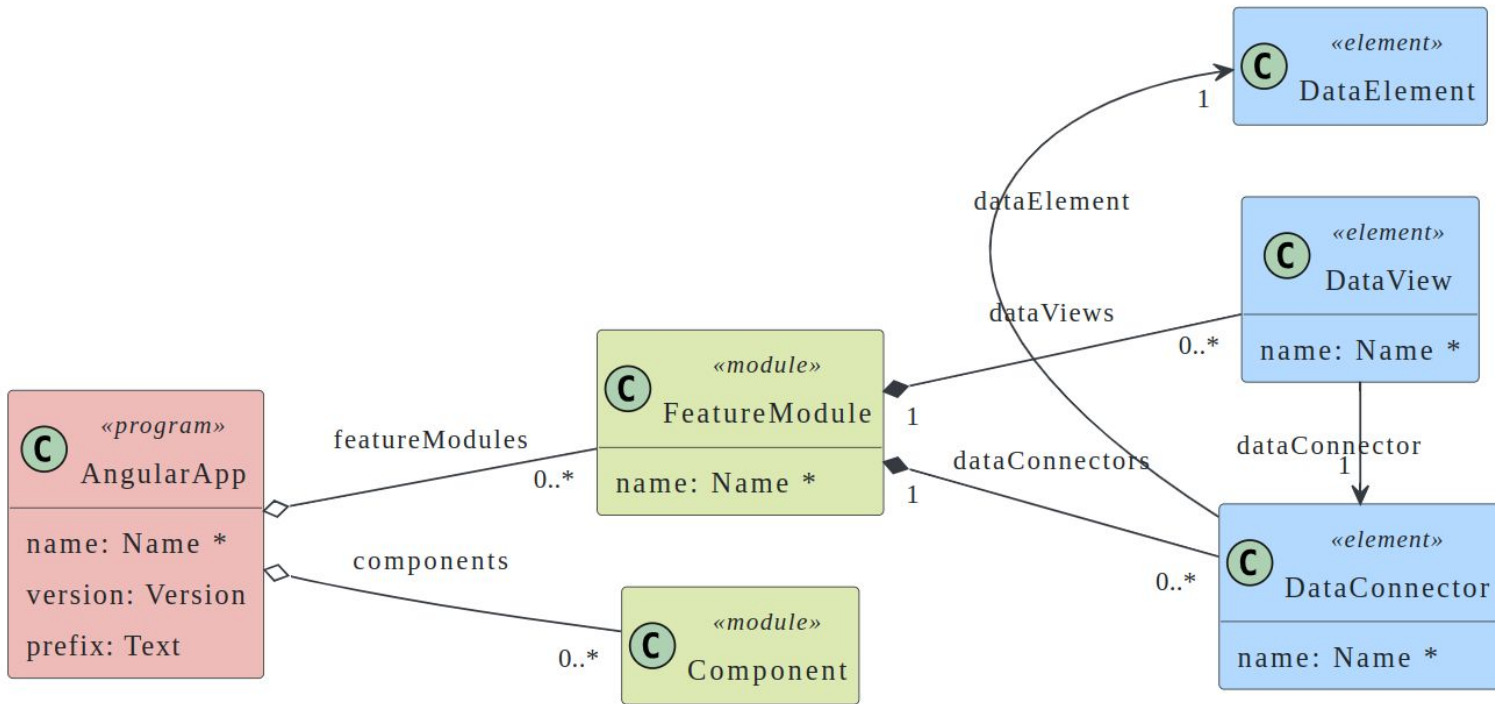


Angular-expanders





Angular MetaModel



New UI design



MENU LEVEL 001

Menu item 001

Menu item 002

DemoComp

Link item

Link item

Customers

[/ Pages / Customers](#)

Export

Search...



Settings

	Name ↓	First name	Last name	Job title	Language	
<input type="checkbox"/>	Support CE support@connecting-expertise.com	Support	CE		ENG	
<input type="checkbox"/>	Support CE support@connecting-expertise.com	Support	CE		ENG	
<input type="checkbox"/>	Support CE support@connecting-expertise.com	Support	CE		ENG	
<input type="checkbox"/>	Support CE support@connecting-expertise.com	Support	CE		ENG	active +
<input type="checkbox"/>	Support CE support@connecting-expertise.com	Support	CE		ENG	active +
<input type="checkbox"/>	Support CE support@connecting-expertise.com	Support	CE		ENG	active +
<input type="checkbox"/>	Support CE support@connecting-expertise.com	Support	CE		ENG	active -

Card view

List view

Comfort display

Comfortable

Normal

Compact



Support CE

Developer SF, Bay Area

support@connecting-expertise.com

Full name

Support CE

Company

Support Company

Contact Phone

+32 485 82 82 82 verified

Company Site

supportcompany.com

Country

Belgium

Communication

Email, Phone

Allow changes

Yes

Edit

Delete



Enum Types

- Subtype of ValueType (next to SimpleValueType)
- Add to Component
- Expanded as Enum
- Currently separate metamodel, will be merged in future

```
<valueType name="FileChangeType"  
  type="enumTypeElements::EnumType">  
  <packageName>org.normalizedsystems.git</packageName>  
  <values>  
    <enumValueType name="ADD">  
      <description>File was created</description>  
    </enumValueType>  
    <enumValueType name="MODIFY">  
      <description>File was updated</description>  
    </enumValueType>  
    <enumValueType name="DELETE">  
      <description>Files was removed</description>  
    </enumValueType>  
    <enumValueType name="RENAME">  
      <description>File was moved or renamed</description>  
    </enumValueType>  
    <enumValueType name="COPY">  
      <description>File is a copy</description>  
    </enumValueType>  
  </values>  
</valueType>
```

Mapping Files: Imports

- Simplifies import mappings
 - Deduplication
 - Conditional Imports
 - Redundant Imports

```
base() ::= <<
package <class.packageName>;

// <expanderComment>

@anchor:imports

// anchor:imports:start
@imports
// anchor:imports:end

// anchor:custom-imports:start
// anchor:custom-imports:end

public interface <class.className>CrudsLocal {
...
>>
```

```
<mapping>
<artifact this="dataElement.qualifiedName + 'CrudsLocal'"
importStrategy="java"/>

<list name="calculatedFields"
eval="dataElement.fields" param="field"
filter="field.fieldType.equals('CALCULATED_FIELD')">
<value name="name" eval="field.name"/>
<uses eval="field.javaType"/>
</list>

<list name="statusFields"
eval="dataElement.fields" param="field"
filter="field.getOption('isStatusField').defined">
<value name="name" eval="field.name"/>
<uses eval="'net democritus.state.StateUpdate'"/>
</list>

<uses eval="'net democritus.sys.DataRef'"/>
<uses eval="'net democritus.sys.ProjectionRef'"/>
<uses eval="'net democritus.sys.CrudsResult'"/>
<uses eval="'net democritus.sys.SearchResult'"/>
<uses eval="'net democritus.sys.ParameterContext'"/>
<uses eval="'net democritus.sys.search.SearchDetails'"/>
</mapping>
```




Expanders Assert

- Replacement for *expanders-test-utils*
- Focus on fixing test traceability
 - ClassPath conflicts
 - ExpansionContext
 - Finding Elements
- JUnit 5
- AssertJ
- Used in:
 - rest-expanders
 - angular-expanders
 - enum-type-expanders
 - micro-radiant-expanders
 - standard-types

```
@ExpanderTest
public class SampleExpanderTest {

    @TestModel(select = "testComp::City")
    Spec baseModel() {
        return component("testComp",
            dataElement("City"));
    }

    @Test
    void testArtifactPath(TestExpansion<DataElementComposite> expansion) {
        expansion.assertThatExpander()
            .hasArtifactPath("C:/MODULE_ROOT/net/demo/cities/City.java");
    }

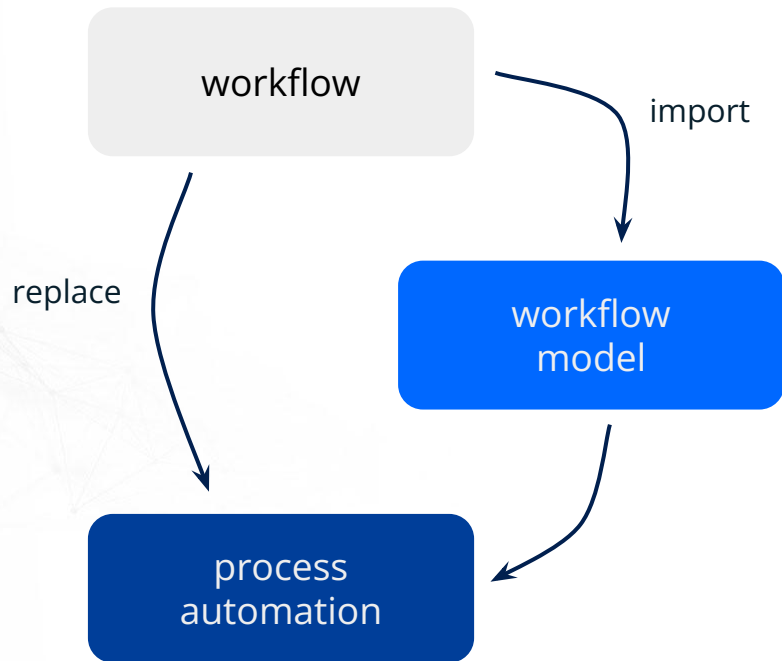
    @Test
    void testIsApplicable(TestExpansion<DataElementComposite> expansion) {
        expansion.assertThatExpander().isNotApplicable();
        expansion.extendModel(option("example.option"))
            .assertThatExpander().isApplicable();
    }

    @Test
    void test_base(TestExpansion<DataElementComposite> expansion) {
        expansion.assertThatExpander().baseContent().matchesTemplate();
    }

    @Test
    void test_anchor_fields(TestExpansion<DataElementComposite> expansion) {
        expansion
            .extendModel(
                field("name: String"),
                field("status: String"))
            .assertThatExpander().anchor("fields")
            .matchesTemplate();
    }
}
```

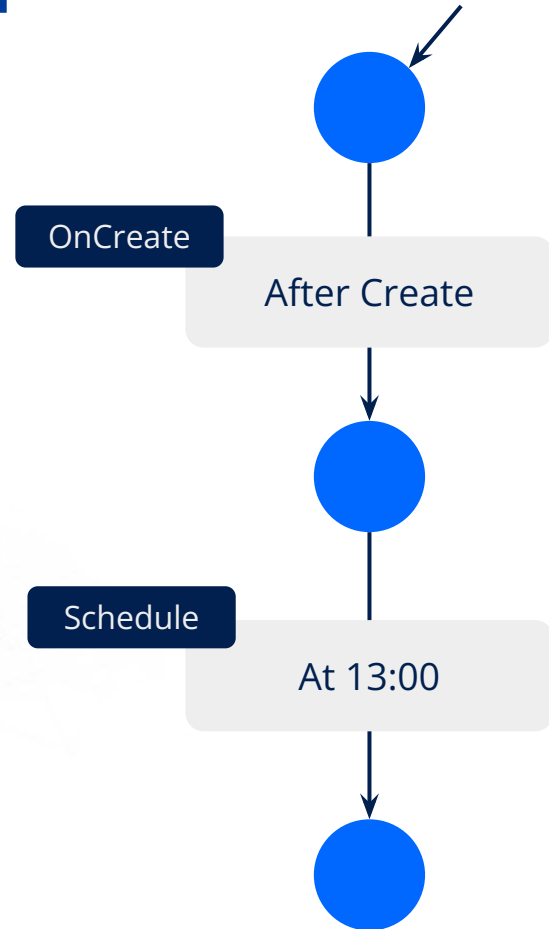

Workflows: ProcessAutomation

- Evolution of 'workflow' component
 - Model-based state-machines
 - Queue-backed scheduling
 - Triggers and Events
- Well defined migration path
- Preview releases 0.x.x



Workflows: ProcessAutomation

- Model behaviour with triggers
 - OnCreate / OnModify
 - OnTransition
 - Schedule
 - FlowEngine
- Fully extendable
 - Define your own triggers
 - Implement & configure different queues



IntelliJ: NSX-Support Plugin

- Developer Support
 - Reordering custom imports
 - Java

```
import net.democritus.sys.TaskResult;
import net.democritus.sys.ParameterContext;
import net.democritus.sys.UserContext;
import net.democritus.sys.Context;

// @anchor:imports:start
import net.palver.logging.LoggerFactory;
import net.palver.logging.Logger;

import java.util.Map;
import java.util.HashMap;
// @anchor:imports:end
// anchor:custom-imports:start
// anchor:custom-imports:end
```

ctrl+s

```
import net.democritus.sys.TaskResult;
import net.democritus.sys.ParameterContext;
import net.democritus.sys.UserContext;
import net.democritus.sys.Context;

// @anchor:imports:start
import net.palver.logging.LoggerFactory;
import net.palver.logging.Logger;
// @anchor:imports:end
// anchor:custom-imports:start
import java.util.Map;
import java.util.HashMap;
// anchor:custom-imports:end
```

IntelliJ: NSX-Support Plugin

- Expander Support
- Improved support for metamodels
 - Loading maven dependencies
- Expander template updates
 - SpecBuilder → TestModelBuilder
 - <artifact> and @imports
 - Junit5 (Jupiter)
 - Expanders Assert
- Expander Mapping Debugger
 - Breakpoints in mapping.xml
 - Inspect contexts
 - Evaluate ognl

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <mapping xmlns="https://schemas.normalizedsystems.org/xsd/expanders/2023/1/0/mapping">
3   <value name="packageName" eval="dataElement.packageName"/>
4   <value name="dataElement" eval="dataElement.name"/>
5
6   <list name="fields" eval="dataElement.fields" param="field">
7     <value name="name" eval="field.name"/>
8   </list>
9 </mapping>
```

```
ognl.eval("dataElement.fields.{? name eq 'status' }.{ name }")
result = {ArrayList@3875} size = 1
  > 0 = "status"
  > context = {DebugMappingContext@3763}
  > ognl = {DebugMappingOgnlEvaluator@3764}
  > scope = {LinkedHashMap@3765} size = 20
    > "classBuilder" -> {ClassBuilder@3792}
    > "businessLogicSettings" -> {BusinessLogicSettingsComposite@3794} "BusinessLogicSettingsCom
    > "applicationInstance" -> {ApplicationInstanceComposite@3796} "ApplicationInstanceComposite
    > "module" -> {ModuleComposite@3798} "ModuleComposite{testComp}"
    > "filters" -> {Filters@3800}
    > "dataElement" -> {DataElementComposite@3802} "DataElementComposite{testComp::TestElem
      > key = "dataElement"
      > value = {DataElementComposite@3802} "DataElementComposite{testComp::TestElement}"
      > mName = "TestElement"
```

Java 21

- Released in September 2023
 - Fully supported for use in NS JEE applications
- Notable features since Java 17
 - Sequences collections API
 - Pattern matching for switch statements
 - Deconstruction of records

For an overview of new features since Java 8:

<https://bit.ly/487I2ly>





REST Expanders

- QuerySearch metamodel is now explicitly used
 - Fields for QueryFilter can be modelled
 - Fields in QueryFilter can be exposed on GETlist
- Implicit metamodel for rest-expanders
 - Already used to expand all Exception handlers
- Ready to support Jakarta EE



QuerySearch Expanders

- Introduction of explicit joins (left inner & left outer)

```
QueryBuilder queryBuilder= select("o")
    .from("com.example.city.City", "o"
        // @anchor:from-arguments:start
        , join("o.country", "o_country").leftOuter()
            .condition(() -> sortFields.stream().map(SortField::getFieldName)
                .anyMatch("country.name":equals)),
        join("o.province", "o_province").leftOuter()
            .condition(() -> sortFields.stream().map(SortField::getFieldName)
                .anyMatch("province.name":equals))
        // @anchor:from-arguments:end
        // anchor:custom-from-arguments:start
        // anchor:custom-from-arguments:end
    )
    .where(buildConstraints(searchParameter.construct(queryFilter)))
    .orderBy(sortFields)
    .useDefaultOrder(true);
```


JEE stack housekeeping

To improve the quality of our JEE stack, we cleaned up a lot of legacy code in our runtime library and expanded code in 2023

- The use of `Option<T>` was replaced with `Optional<T>`
- `UserContext` as main context transport was removed
- `CrudsResult<T>` was completely rebuilt
- `Mindmap` support was removed
- Methods deprecated for >1 year were removed
- Many deprecated model options were removed
- Maven groupIds made unique and consistent
- `FinderBean` with reflection is no longer default

Planned features for 2024H1

 In Progress

REST metamodel (implicit)
& Expanded REST clients

JAX-RS API to replace Struts

Angular: contribution

Expanders: tags

Implicit name rework

Angular: New Designs

DataCommand redesign

ProcessAutomation 1.0.0

Angular: Backend Clients

μRadiant plugins & rework

Mac support for NSX tools

NS Initializer modules