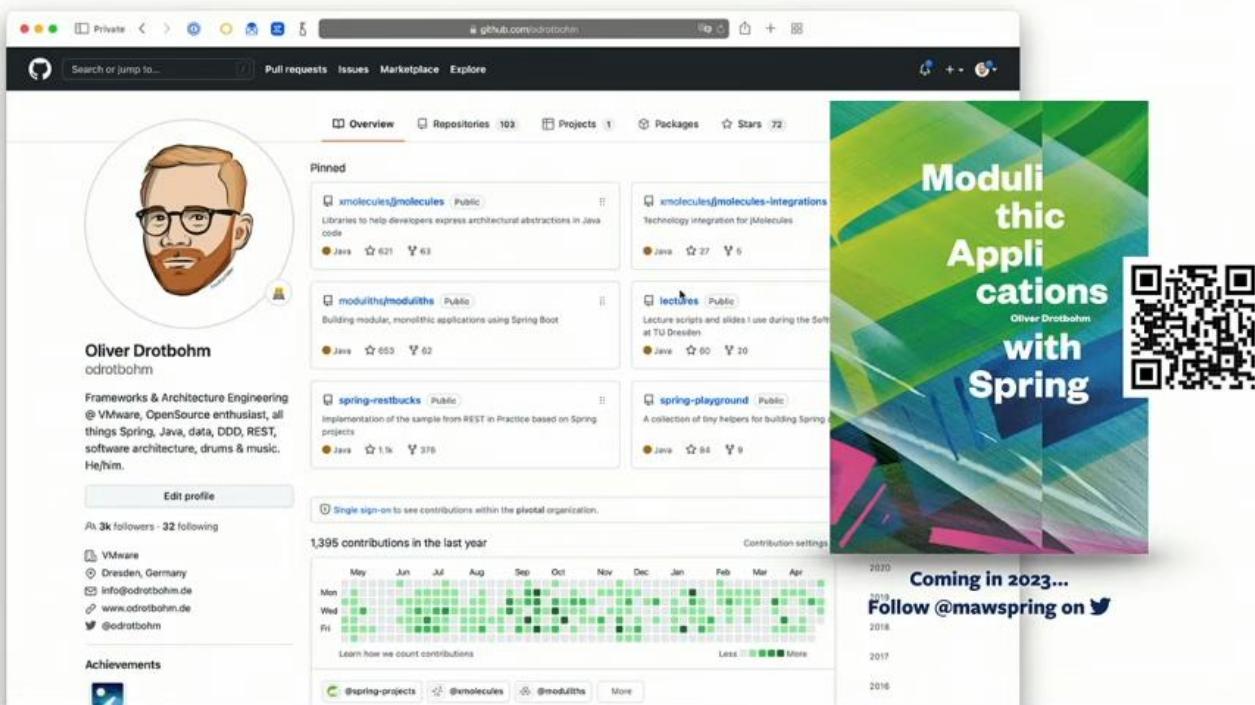


SPRING FOR ARCHITECTURALLY CURIOS DEVELOPERS

Oliver Drotbohm  odrotbohm  odrotbohm@vmware.com

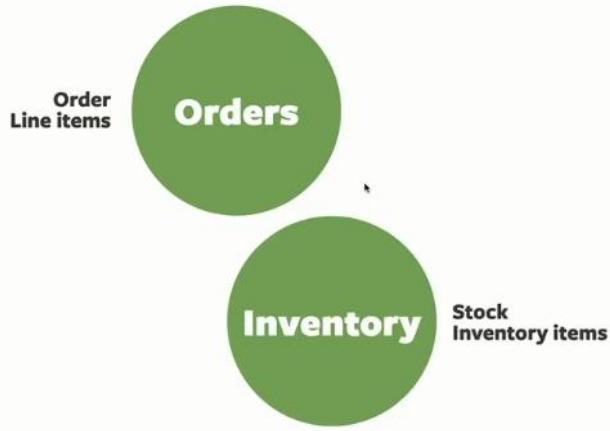
The architecture and design of an application significantly influences its maintainability, testability, and quality in general. Spring has always been a versatile tool that supports architects in implementing the practices and patterns that have emerged to align technical building blocks in their code bases with domain concepts and boundaries. It lets developers build applications that clearly reflect architectural ideas and that are ultimately more amendable and maintainable. Key aspects of that are code organization for encapsulation, the Spring bean relationship arrangement, and how to use domain events to decouple logically individual parts of the application even within a singular artifact. This talk discusses those patterns and approaches to introduce libraries, such as **jMolecules** and **Moduliths** to, in turn, show how they uniquely position Spring developers to build better structured, more maintainable applications.



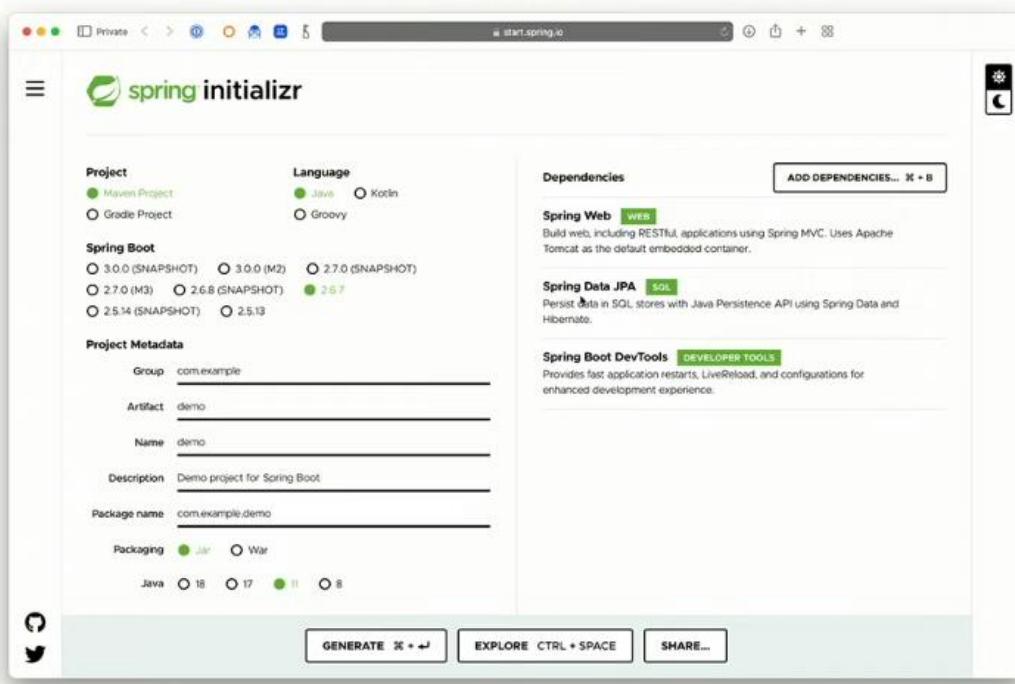
A screenshot of a GitHub profile page for Oliver Drotbohm. The profile picture shows a cartoon version of him with glasses and a beard. The bio section says: "Frameworks & Architecture Engineering @ VMware, OpenSource enthusiast, all things Spring, Java, data, DDD, REST, software architecture, drums & music. He/Him." Below the bio is a "Edit profile" button. The "Achievements" section shows 1 achievement. The "Contributions" section shows a heatmap of contributions from May to April of the previous year. On the right side of the GitHub page, there is a large image of the book "Modular Applications with Spring" by Oliver Drotbohm. The book cover has a colorful geometric design and the text "Modular Applications with Spring" and "Coming in 2023... Follow @mawspring on Twitter". A QR code is also visible next to the book cover.

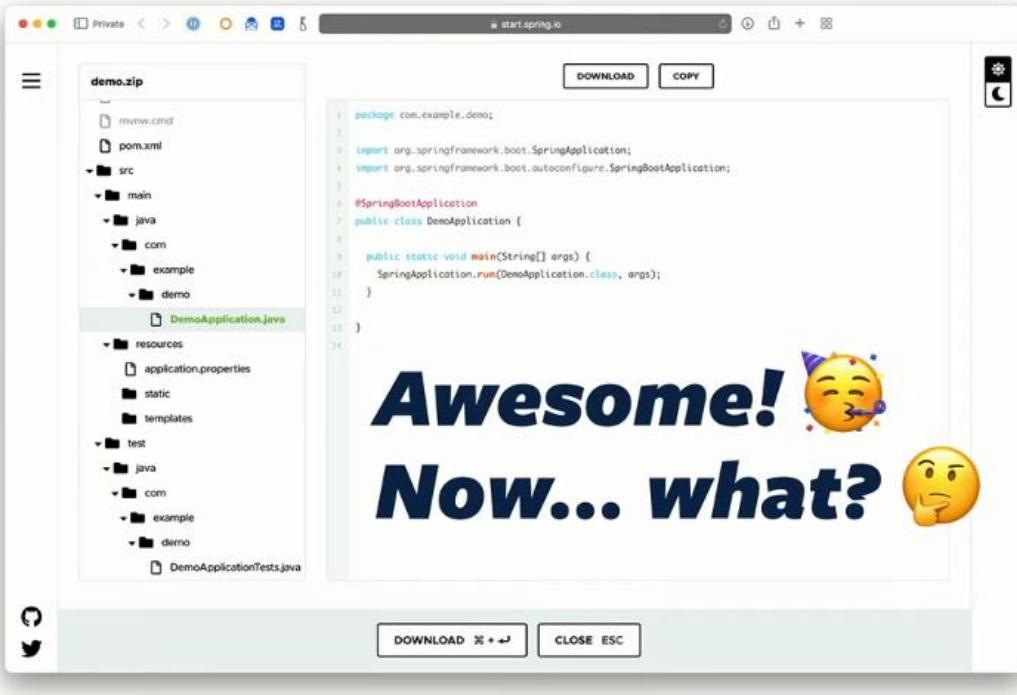
The Domain

Let us see how to structure a spring application to support development work



***Unfortunately, post-its
don't run in production...***





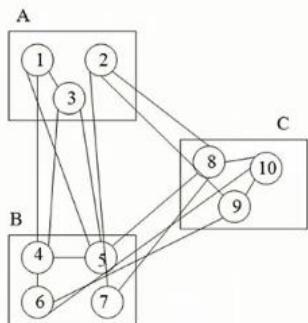
How do you structure your system as recommended by the spring team?

“ **How do we represent functional architecture in a codebase?**

Let's get started...

src/main/java

- 📦acme.myproject
- 📦acme.myproject.domain
- 📦acme.myproject.persistence
- 📦acme.myproject.service
- 📦acme.myproject.web



Bad modularization:
low cohesion, high coupling

via <https://devopedia.org/cohesion-vs-coupling>

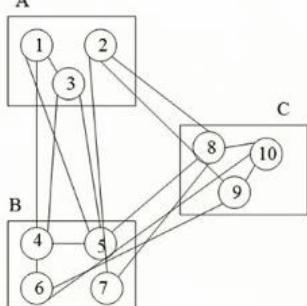


*via Dan North – CUPID—for joyful coding
Video cast episode @ Software Architektur im Stream (english)*

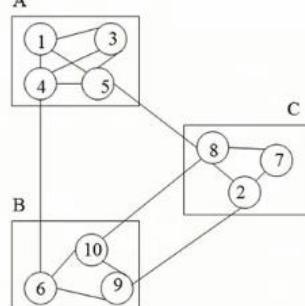
Much better...

📁 **src/main/java**
 ⚡acme.myproject
 ⚡acme.myproject.customer
 ⚡acme.myproject.inventory
 ⚡acme.myproject.order





Bad modularization:
low cohesion, high coupling



Good modularization:
high cohesion, low coupling

via <https://devopedia.org/cohesion-vs-coupling>



Can the spring framework help support this functional approach?

Sample code

🔗 <https://github.com/odrotbohm/arch-evident-spring>
 ↗ events/230316-voxxed-days-zurich



Demo time! 

The screenshot shows a Java IDE interface with two main panes. On the left is the 'Project Explorer' pane, displaying the project structure for 'arch-evident-spring 0.0.1-SNAPSHOT [boot] [steps/0 5ec2d71 - arch-evide]'. The structure includes 'src/main/java' with packages like 'example', 'example.customer', 'example.inventory', 'example.order', and 'src/test/java' with 'exampleTests.java' and 'ModularityTests.java'. On the right is the 'Code Editor' pane, showing the source code for 'ModularityTests.java'. The code defines a test class 'ModularityTests' that uses annotations to verify modularity, specifically testing the 'OrderManagement' module.

```

2* Copyright 2022 the original author or authors.
16 package example;
17
18 import org.junit.jupiter.*;
19
20 /**
21 * @author Oliver Drotbohm
22 */
23
24
25 /**
26 * @author Oliver Drotbohm
27 */
28 class ModularityTests {
29
30     /**
31      * ApplicationModules module
32     */
33     @Test
34     void verifyModularity() {
35         // → Module model
36         // modules.forEach(s)
37     }
38 }

```

These is all about the spring beans, components and how we arrange them.

This screenshot shows the same Java IDE environment. The 'Code Editor' pane now displays the 'Order.java' code, which implements an 'AggregateRoot<Order, OrderIdentifier>'. It includes methods for creating an order, adding line items, and completing it. The 'Project Explorer' pane shows the project structure again, with 'Order.java' highlighted in the 'src/main/java/example/order' package.

```

2* Copyright 2022 the original author or authors.
16 package example.order;
17
18 import example.customer.Customer;
19
20 /**
21 * @author Oliver Drotbohm
22 */
23
24 /**
25 * @Getter
26 * @Table(name = "MyOrder")
27 */
28 public class Order implements AggregateRoot<Order, OrderIdentifier> {
29
30     private final OrderIdentifier id;
31     private final Association<Customer, CustomerIdentifier> customer;
32     private Status status;
33
34     private final List<LineItem> lineItems = new ArrayList<>();
35
36     public Order(CustomerIdentifier customerId) {
37
38         this.id = new OrderIdentifier(UUID.randomUUID());
39         this.status = Status.OPEN;
40         this.customer = Association.forName(customerId);
41     }
42
43     Order complete() {
44
45         this.status = Status.COMPLETED;
46
47         return this;
48     }
49
50     Order add(LineItem item) {
51
52         this.lineItems.add(item);
53
54         return this;
55     }
56
57     public Order create(CustomerIdentifier customerId) {
58
59         return new Order(customerId);
60     }
61
62     public Order complete(Order order) {
63
64         return order;
65     }
66 }

```

This screenshot shows the Java IDE again. The 'Code Editor' pane now contains the 'OrderManagement.java' code, which is annotated with '@Service' and '@RequiredArgsConstructor'. It has methods for creating an order and completing an existing one. The 'Project Explorer' pane shows the project structure, with 'OrderManagement.java' highlighted in the 'src/main/java/example/order' package.

```

2* Copyright 2022 the original author or authors.
16 package example.order;
17
18 import example.customer.Customer.CustomerIdentifier;
19
20 /**
21 * @author Oliver Drotbohm
22 */
23
24 /**
25 * @Transactional
26 */
27
28 @Service
29 @RequiredArgsConstructor
30 public class OrderManagement {
31
32     public Order create(CustomerIdentifier customerId) {
33
34         return new Order(customerId);
35     }
36
37     public Order complete(Order order) {
38
39         return order;
40     }
41 }

```

```

Project Explorer X Package Explorer
ModularityTests.java Order.java OrderManagement.java Inventory.java
+ Arch-Evident-Spring
  + arch-evident-spring 0.0.1-SNAPSHOT [boot] [steps/0 sec0207] - arch-evident-spring
    + example
      + Application.java
      + example.customer
        + Customer.java
      + example.inventory
        + Inventory.java
      + example.order
        + Order.java
        + OrderManagement.java
    + srcTest/java
      + example
        + ApplicationTests.java
        + ModularityTests.java
      + example.inventory
        + example.order
      + srcTestResources
    + Maven Dependencies
    + JRE System Library [JavaSE-17]
    + target/generated-sources/annotations
    + target/generated-test-sources/test-annotations
    + etc
    + arc
      + main
        + encoder
        + index.adoc
    + target
    + maven
    + mvnw.cmd
    + pom.xml
    + README.adoc
  + moduluses
  + modules
  + spring-modulith 0.0.0-SNAPSHOT [main - spring-modulith]
  Spring HATEOAS [main - spring-hateoas]
  Other Projects

```

```

16 package example.inventory;
17
18 import lombok.RequiredArgsConstructor;
19
20 /**
21 * Copyright 2022 the original author or authors.
22 */
23 * @author Oliver Drotbohm
24 */
25 @Service
26 @RequiredArgsConstructor
27 public class Inventory {
28
29 /**
30 * Updates the stock for all line items contained in the order.
31 */
32 public void updateStock() {
33
34 }
35
36

```

```

Project Explorer X Package Explorer
ModularityTests.java Order.java OrderManagement.java Inventory.java
+ Arch-Evident-Spring
  + arch-evident-spring 0.0.1-SNAPSHOT [boot] [steps/0 sec0207] - arch-evident-spring
    + srcMain/java
      + Application.java
    + example
      + example.customer
        + Customer.java
      + example.inventory
        + Inventory.java
      + example.order
        + Order.java
        + OrderManagement.java
    + srcTest/java
      + example
        + ModularityTests.java
    + srcTestResources
    + Maven Dependencies
    + JRE System Library [JavaSE-17]
    + target/generated-sources/annotations
    + target/generated-test-sources/test-annotations
    + etc
    + arc
      + main
        + encoder
        + index.adoc
    + target
    + maven
    + mvnw.cmd
    + pom.xml
    + README.adoc
  + moduluses
  + modules
  + spring-modulith 0.0.0-SNAPSHOT [main - spring-modulith]
  Spring HATEOAS [main - spring-hateoas]
  Other Projects

```

```

24 * Copyright 2022 the original author or authors.
25 */
26 * @author Oliver Drotbohm
27 */
28 class ModularityTests {
29
30 ApplicationModules modules = ApplicationModules.of(Application.class);
31
32 @Test
33 void verifyModularity() {
34
35 // → Module model
36 modules.forEach(System.out::println);
37
38 // → Trigger verification
39 modules.verify();
40 }
41
42 @Test
43 void renderDocumentation() throws Exception {
44
45 var canvasOptions = CanvasOptions.defaults()
46
47 // → Optionally enable linking of JavaDoc
48 // .withApiBase("https://foobar.something")
49
50 ;
51
52 var docOptions = DiagramOptions.defaults()
53
54 // .withStyle(DiagramStyle.UML);
55
56 new Documenter(modules) // .
57 .writeDocumentation(docOptions, canvasOptions);
58

```

```

Project Explorer X Package Explorer
ModularityTests.java Order.java OrderManagement.java Inventory.java Application.java
+ Arch-Evident-Spring
  + arch-evident-spring 0.0.1-SNAPSHOT [boot] [steps/0 sec0207] - arch-evident-spring
    + example
      + Application.java
    + example.customer
      + Customer.java
    + example.inventory
      + Inventory.java
    + example.order
      + Order.java
      + OrderManagement.java
    + srcTest/java
      + example
        + ApplicationTests.java
        + ModularityTests.java
      + example.inventory
        + example.order
      + srcTestResources
    + Maven Dependencies
    + JRE System Library [JavaSE-17]
    + target/generated-sources/annotations
    + target/generated-test-sources/test-annotations
    + etc
    + arc
      + main
        + encoder
        + index.adoc
    + target
    + maven
    + mvnw.cmd
    + pom.xml
    + README.adoc
  + moduluses
  + modules
  + spring-modulith 0.0.0-SNAPSHOT [main - spring-modulith]
  Spring HATEOAS [main - spring-hateoas]
  Other Projects

```

```

1 package example;
2
3 import org.springframework.boot.SpringApplication;
4
5 /**
6 * Spring for the architecturally curious developer.
7 *
8 * @author Oliver Drotbohm (@odrotbohm).
9 */
10 @SpringBootApplication
11 @ConfigurationPropertiesScan
12 public class Application {
13
14 public static void main(String[] args) {
15     SpringApplication.run(Application.class, args);
16 }
17
18 }
19
20

```

The screenshot shows a Java project structure in the Project Explorer:

- Arch-Evident-Spring
- src/main/java:
 - example
 - example.customer
 - example.inventory
 - example.order
 - example.test
 - ModularityTests.java
- src/test/java:
 - example
 - example.inventory
 - example.order
 - example.orderManagement
 - example.test
 - ModularityTests.java
- Maven Dependencies
- target/generated-sources/annotations
- target/generated-test-sources/test-annotations
- etc
- src/main/resources
- src/main/webapp
- target
- mmr
- mimiccmd
- pom.xml
- readme.adoc
- src/main
- modul
- spring-modulith 0.0.0-SNAPSHOT (main - spring-modulith)
- Spring-HATEOAS (main - Spring-HATEOAS)
- Other Projects

The code editor displays the `ModularityTests.java` file:

```
2+ * Copyright 2022 the original author or authors.
16 package example;
17
18 import org.junit.jupiter.api.Test;
19
20 /**
21 * @author Oliver Drotbohm
22 */
23
24 class ModularityTests {
25
26     ApplicationModules modules = ApplicationModules.of(Application.class);
27
28     @Test
29     void verifyModularity() {
30
31         // → Module model
32         // modules.forEach(System.out::println);
33
34         // → Trigger verification
35         modules.verify();
36     }
37
38     @Test
39     void renderDocumentation() throws Exception {
40
41         var canvasOptions = CanvasOptions.defaults()
42
43             // → Optionally enable linking of JavaDoc
44             // .withApiBase("https://foobar.something")
45
46         ;
47
48         var docOptions = DiagramOptions.defaults()
49             .withStyle(DiagramStyle.UML);
50
51         new Documenter(modules) //
52             .writeDocumentation(docOptions, canvasOptions);
53     }
54
55 }
```

The screenshot shows the same Java project structure in the Project Explorer.

A context menu is open over the line 56 of the code editor, listing options such as "Run JUnit Test", "Run As", "Copy", "Cut", etc.

The code editor displays the `ModularityTests.java` file:

```
2+ * Copyright 2022 the original author or authors.
16 package example;
17
18 import org.junit.jupiter.api.Test;
19
20 /**
21 * @author Oliver Drotbohm
22 */
23
24 class ModularityTests {
25
26     ApplicationModules modules = ApplicationModules.of(Application.class);
27
28     @Test
29     void verifyModularity() {
30
31         // → Module model
32         modules.forEach(System.out::println);
33
34         // → Trigger verification
35         modules.verify();
36     }
37
38     @Test
39     void renderDocumentation() throws Exception {
40
41         var canvasOptions = CanvasOptions.defaults()
42
43             // → Optionally enable linking of JavaDoc
44             // .withApiBase("https://foobar.something")
45
46         ;
47
48         var docOptions = DiagramOptions.defaults()
49             .withStyle(DiagramStyle.UML);
50
51         new Documenter(modules) //
52             .writeDocumentation(docOptions, canvasOptions);
53     }
54
55 }
```

The screenshot shows a Java project named "Arch Evident Spring" in the Project Explorer. The code editor displays a test class named "ModularityTests.java". The code is annotated with JUnit Jupiter annotations and includes a log message from Logback.

```
* Copyright 2022 the original author or authors.
16 package example;
17
18 import org.junit.jupiter.api.Test;
24
25 /**
26 * @author Oliver Drotbohm
27 */
28 class ModularityTests {
29
30 ApplicationModules modules = ApplicationModules.of(Application.class);
31
32 @Test
33 void verifyModularity() {
34
# Customer
> Logical name: customer
> Base package: example.customer
> Spring beans: none

# Order
> Logical name: order
> Base package: example.order
> Spring beans:
+ ...OrderManagement

# Inventory
> Logical name: inventory
> Base package: example.inventory
> Spring beans:
+ ...Inventory
}
```

The screenshot shows the same Java project "Arch Evident Spring" in the Project Explorer. The code editor displays the same "ModularityTests.java" file, which contains a test method "verifyModularity".

```
* Copyright 2022 the original
16 package example;
17
18 import org.junit.jupiter.api.
24
25 /**
26 * @author Oliver Drotbohm
27 */
28 class ModularityTests {
29
30 » ApplicationModules modules
31
32 » @Test
33 » void verifyModularity() {
34
}
```

Project Explorer X Package Explorer

```

37     // → Trigger verification
38     modules.verify();
39 }
40 }
41
42 @Test
43 void renderDocumentation() throws Exception {
44     var canvasOptions = CanvasOptions.defaults()
45
46     // → Optionally enable linking of JavaDoc
47     // .withApiBase("https://foobar.something")
48
49
50 ;
51

```

Problems Error Log Progress Search Call Hierarchy Coverage JUnit Boot Dashboard Maven Repositories History Console Properties

Run: 5/1 Errors: 0 Failures: 0

ModularityTests [Runner: JUnit 5] (0.127 s) Failure Trace

Project Explorer X Package Explorer

```

1= Developer documentation
2
3:moduliths: ../../target/spring-modulith-docs
4
5= Overview
6plantuml::{moduliths}/components.puml[format=svg,]
7
8= Customer
9plantuml::{moduliths}/module-customer.puml[format=
10include::{moduliths}/module-customer.adoc[]
11
12= Inventory
13plantuml::{moduliths}/module-inventory.puml[format=
14include::{moduliths}/module-inventory.adoc[]
15

```

Please wait, your document preview is being initialized

Problems Error Log Progress Search Call Hierarchy Coverage JUnit Boot Dashboard Maven Repositories History Console Properties

Run: 5/1 Errors: 0 Failures: 0

ModularityTests [Runner: JUnit 5] (0.127 s) Failure Trace

Project Explorer X Package Explorer

```

1= Developer documentation
2
3:moduliths: ../../target/spring-modulith-docs
4
5= Overview
6plantuml::{moduliths}/components.puml[format=svg,title="Overall c
7
8= Customer
9plantuml::{moduliths}/module-customer.puml[format=svg,title="Inv
10include::{moduliths}/module-customer.adoc[]
11
12= Inventory
13plantuml::{moduliths}/module-inventory.puml[format=svg,title="Inv
14include::{moduliths}/module-inventory.adoc[]
15
16= Orders
17plantuml::{moduliths}/module-order.puml[format=svg,title="Order c
18include::{moduliths}/module-order.adoc[]
19

```

Developer documentation

Overview

Application

- + Component: Module - Order
- + depends on
- + Component: Module - Customer

Customer

Customer

Base package example.customer

Aggregate roots e.c.Customer

Inventory

```

1 [Developer documentation]
2
3 :moduliths: ../../target/spring-modulith-docs
4
5 == Overview
6 plantuml:::{moduliths}/components.puml[format=svg,title="Overall c
7
8 == Customer
9 plantuml:::{moduliths}/module-customer.puml[format=svg,title="Inve
10 include:::{moduliths}/module-customer.adoc[]
11
12 == Inventory
13 plantuml:::{moduliths}/module-inventory.puml[format=svg,title="Inv
14 include:::{moduliths}/module-inventory.adoc[]
15
16 == Orders
17 plantuml:::{moduliths}/module-order.puml[format=svg,title="Order c
18 include:::{moduliths}/module-order.adoc[]
19

```

Customer

| | |
|-----------------|------------------|
| Base package | example.customer |
| Aggregate roots | e.c.Customer |

Inventory

| | |
|-------------------|-------------------|
| Base package | example.inventory |
| Spring components | Services |
| | • e.i.Inventory |

Orders

| | |
|-------------------|-----------------------|
| Base package | example.order |
| Spring components | Services |
| | • e.o.OrderManagement |
| Aggregate roots | e.o.Order |

```

1 [Developer documentation]
2
3 :moduliths: ../../target/spring-modulith-docs
4
5 == Overview
6 plantuml:::{moduliths}/components.puml[format=svg,title="Overall c
7
8 == Customer
9 plantuml:::{moduliths}/module-customer.puml[format=svg,title="Inve
10 include:::{moduliths}/module-customer.adoc[]
11
12 == Inventory
13 plantuml:::{moduliths}/module-inventory.puml[format=svg,title="Inv
14 include:::{moduliths}/module-inventory.adoc[]
15
16 == Orders
17 plantuml:::{moduliths}/module-order.puml[format=svg,title="Order c
18 include:::{moduliths}/module-order.adoc[]
19

```

Customer

| | |
|-----------------|------------------|
| Base package | example.customer |
| Aggregate roots | e.c.Customer |

Inventory

| | |
|-------------------|-------------------|
| Base package | example.inventory |
| Spring components | Services |
| | • e.i.Inventory |

Orders

| | |
|-------------------|-----------------------|
| Base package | example.order |
| Spring components | Services |
| | • e.o.OrderManagement |
| Aggregate roots | e.o.Order |

Order

| | |
|-------------------|-----------------------|
| Base package | example.order |
| Spring components | Services |
| | • e.o.OrderManagement |
| Aggregate roots | e.o.Order |

Customer

| | |
|-----------------|------------------|
| Base package | example.customer |
| Aggregate roots | e.c.Customer |

Project Explorer X Package Explorer

ModularityTests.java Order.java OrderManagement.java Inventory.java index.adoc

```

1 = Developer documentation
2
3:moduliths: ../../target/spring-modulith-docs
4
5 = Overview
6 plantuml::{moduliths}/components.puml[format=svg,1
7
8 = Customer
9 plantuml::{moduliths}/module-customer.puml[format=
10 include::{moduliths}/module-customer.adoc[]
11
12 = Inventory
13 plantuml::{moduliths}/module-inventory.puml[format=
14 include::{moduliths}/module-inventory.adoc[]
15

```

Base package example.inventory

Spring components Services

- e.i.Inventory

Orders

Order

+Component: Module+ Order

Installed asciidoctor: Processing file:/Users/odrotbohm/Documents/workspace/arch-evident-spring/src/main/asciidoc/index.adoc

2023-03-16 10:50:32.245 java[58756:1579205] CoreText note: Client requested name ".Helvetica Neue DeskInterface-Regular", it will get T

2023-03-16 10:50:32.245 java[58756:1579205] CoreText note: Set a breakpoint on CTFontLogSystemFontNameRequest to debug.

2023-03-16 10:50:32.246 java[58756:1579205] CoreText note: Client requested name ".Helvetica Neue DeskInterface-UltralightP2", it wi

2023-03-16 10:50:32.246 java[58756:1579205] CoreText note: Client requested name ".Helvetica Neue DeskInterface-Thin", it will get T

2023-03-16 10:50:32.247 java[58756:1579205] CoreText note: Client requested name ".Helvetica Neue DeskInterface-MediumP4", it will g

2023-03-16 10:50:32.247 java[58756:1579205] CoreText note: Client requested name ".Helvetica Neue DeskInterface-Bold", it will get T

2023-03-16 10:50:32.320 java[58756:1579205] CoreText note: Client requested name ".Helvetica Neue DeskInterface-Regular", it will get T

2023-03-16 10:50:32.321 java[58756:1579205] CoreText note: Client requested name ".Helvetica Neue DeskInterface-Bold", it will get T

Installed asciidoctor: Processing file:/Users/odrotbohm/Documents/workspace/arch-evident-spring/src/main/asciidoc/index.adoc

Installed asciidoctor: Processing file:/Users/odrotbohm/Documents/workspace/arch-evident-spring/src/main/asciidoc/index.adoc

2023-03-16 11:45:27.724 java[65323:15855724] CoreText note: Client requested name ".Helvetica Neue DeskInterface-Regular", it will get T

2023-03-16 11:45:27.724 java[65323:15855724] CoreText note: Set a breakpoint on CTFontLogSystemFontNameRequest to debug.

2023-03-16 11:45:27.725 java[65323:15855724] CoreText note: Client requested name ".Helvetica Neue DeskInterface-UltralightP2", it wi

2023-03-16 11:45:27.725 java[65323:15855724] CoreText note: Client requested name ".Helvetica Neue DeskInterface-Thin", it will get T

2023-03-16 11:45:27.725 java[65323:15855724] CoreText note: Client requested name ".Helvetica Neue DeskInterface-MediumP4", it will g

2023-03-16 11:45:27.725 java[65323:15855724] CoreText note: Client requested name ".Helvetica Neue DeskInterface-Bold", it will get T

2023-03-16 11:45:27.779 java[65323:15855724] CoreText note: Client requested name ".Helvetica Neue DeskInterface-Regular", it will get T

2023-03-16 11:45:27.779 java[65323:15855724] CoreText note: Client requested name ".Helvetica Neue DeskInterface-Bold", it will get T

Project Explorer X Package Explorer

ModularityTests.java Order.java OrderManagement.java Inventory.java index.adoc

```

1 = Developer documentation
2
3:moduliths: ../../target/spring-modulith-docs
4
5 = Overview
6 plantuml::{moduliths}/components.puml[format=svg,1
7
8 = Customer
9 plantuml::{moduliths}/module-customer.puml[format=
10 include::{moduliths}/module-customer.adoc[]
11
12 = Inventory
13 plantuml::{moduliths}/module-inventory.puml[format=
14 include::{moduliths}/module-inventory.adoc[]
15
16 = Orders
17 plantuml::{moduliths}/module-order.puml[format=sv
18 include::{moduliths}/module-order.adoc[]
19

```

Base package example.order

Spring components Services

- e.o.OrderManagement

Orders

Order

+Component: Module+ Order

Depends on

+Component: Module+ Customer

Figure 4. Order component overview

Base package example.order

Spring components Services

- e.o.OrderManagement

Customer

AbstractAggregateRoot CustomerIdentifier

Customer

- id: CustomerIdentifier
- address: String
- userAccount: UserAccount
- Customer
- Customer userAccount: UserAccount, address: String
- getid: CustomerIdentifier

Writetable Insert 6 / 12 / 88 1308M of 2008M

Project Explorer X Package Explorer

```

src/main/java
  example
    Application.java
    example.customer
      Customer.java
    example.inventory
      Inventory.java
        Order.java
        OrderManagement.java
    example.order
      OrderTests.java
    example.inventory
      InventoryTests.java
    example.order
  src/test/java
  src/main/resources
  Maven Dependencies
  JRE System Library [JavaSE-17]
  target/generated-sources/annotations
  target/generated-test-sources/test-annotations
  etc
  src
    main
      example
        Inventory.java
        Order.java
        OrderManagement.java
      example.customer
        Customer.java
      example.inventory
        Inventory.java
        Order.java
        OrderManagement.java
      example.order
        OrderTests.java
    test
    resources
  pom.xml
  README.adoc

```

Inventory.java

```

2* Copyright 2022 the original author or authors.
16 package example.inventory;
17
18 import lombok.RequiredArgsConstructor;
19
20 /**
21 * @author Oliver Drotbohm
22 */
23 @Service
24 @RequiredArgsConstructor
25 public class Inventory {
26
27     /**
28      * Updates the stock for all line items contained in the order.
29      */
30     public void updateStock() {
31
32     }
33
34 }
35
36

```

Problems Error Log Progress Search PlantUML Call Hierarchy Coverage JUnit Boot Dashboard Maven Repositories History Console Properties

workspace - arch-evident-spring/src/main/java/example/inventory/InventoryRepository.java - Spring Tool Suite 4

```

src/main/java
  example
    Application.java
    example.customer
      Customer.java
    example.inventory
      Inventory.java
      InventoryRepository.java
    example.order
      Order.java
      OrderManagement.java
    example.order
  src/test/java
  src/main/resources
  Maven Dependencies
  JRE System Library [JavaSE-17]
  target/generated-sources/annotations
  target/generated-test-sources/test-annotations
  etc
  src
    main
      example
        Inventory.java
        Order.java
        OrderManagement.java
      example.customer
        Customer.java
      example.inventory
        Inventory.java
        InventoryRepository.java
      example.order
        Order.java
        OrderManagement.java
    test
    resources
  pom.xml
  README.adoc

```

InventoryRepository.java

```

2* Copyright 2023 the original author or authors.
16 package example.inventory;
17
18 /**
19 * @author Oliver Drotbohm
20 */
21
22 public class InventoryRepository {
23
24 }
25

```

Problems Error Log Progress Search PlantUML Call Hierarchy Coverage JUnit Boot Dashboard Maven Repositories History Console Properties

Project Explorer X | Package Explorer

Inventory.java | InventoryRepository.java

```
2+ * Copyright 2023 the original author or authors.
16 package example.inventory;
17
18 import org.springframework.stereotype.Repository;
19
20 /**
21 *
22 * @author Oliver Drotbohm
23 */
24 @Repository
25 public class InventoryRepository {
26
27 }
```

InventoryRepository

Arch-Evident-Spring

src/main/java

- example
- Application.java
- example.customer
- Customer.java
- example.inventory
- Inventory.java
- InventoryRepository.java
- example.order
- Order.java
- OrderManagement.java
- srcTestJava
- ApplicationTests.java
- ModularityTests.java
- example.inventory
- example.order
- srcTestResources
- Maven Dependencies
- JRE System Library [JavaSE-17]
- target/generated-sources/annotations
- target/generated-test-sources/test-annotations
- etc
- src
- main
- execdac
- index.adoc
- test
- target
- maven
- maven.cmd
- pom.xml
- README.adoc
- modules
- modules
- spring-modult 0.0.0-SNAPSHOT [main - spring-modult]
- Spring HATEOAS [main - spring-hateoas]
- Other Projects

Problems Error Log Progress Search Call Hierarchy Coverage JUnit Root Dashboard Maven Repositories History Console Properties

Project Explorer X | Package Explorer

Inventory.java | InventoryRepository.java

```
2+ * Copyright 2022 the original author or authors.
16 package example.inventory;
17
18 import lombok.RequiredArgsConstructor;
19
20 /**
21 * @author Oliver Drotbohm
22 */
23
24 @Service
25 @RequiredArgsConstructor
26 private final InventoryRepository repository;
27
28 /**
29 * Updates the stock for all line items contained in the order.
30 */
31
32 public void updateStock() {
33 }
34
35
36 }
```

Inventory

repository: InventoryRepository

updateStock(): void

Inventory(repository: InventoryRepository)

Arch-Evident-Spring

src/main/java

- example
- Application.java
- example.customer
- Customer.java
- example.inventory
- Inventory.java
- InventoryRepository.java
- example.order
- Order.java
- OrderManagement.java
- srcTestJava
- ApplicationTests.java
- ModularityTests.java
- example.inventory
- example.order
- srcTestResources
- Maven Dependencies
- JRE System Library [JavaSE-17]
- target/generated-sources/annotations
- target/generated-test-sources/test-annotations
- etc
- src
- main
- execdac
- index.adoc
- test
- target
- maven
- maven.cmd
- pom.xml
- README.adoc
- modules
- modules
- spring-modult 0.0.0-SNAPSHOT [main - spring-modult]
- Spring HATEOAS [main - spring-hateoas]
- Other Projects

Problems Error Log Progress Search Call Hierarchy Coverage JUnit Root Dashboard Maven Repositories History Console Properties

Project Explorer X Package Explorer

```

1 = Developer documentation
2
3 :moduliths: ../../target/spring-modulith-docs
4
5 == Overview
6 plantuml::{moduliths}/components.puml[format=svg,1
7
8 == Customer
9 plantuml::{moduliths}/module-customer.puml[format=
10 include ::{moduliths}/module-customer.adoc[]
11
12 == Inventory
13 plantuml::{moduliths}/module-inventory.puml[format=
14 include ::{moduliths}/module-inventory.adoc[]
15
16 == Orders
17 plantuml::{moduliths}/module-order.puml[format=sv
18 include ::{moduliths}/module-order.adoc[]
19

```

Please wait, your document preview is being initialized

Problems Error Log Progress Search PlantUML Call Hierarchy Coverage JUNIT Boot Dashboard Maven Repositories History Console Properties

Runs: 1/1 Errors: 0 Failed: 0

ModularityTests [Runner: JUnit 5] (0,109 s) Failure Trace

Project Explorer X Package Explorer

```

1 = Developer documentation
2
3 :moduliths: ../../target/spring-modulith-docs
4
5 == Overview
6 plantuml::{moduliths}/components.puml[format=svg,1
7
8 == Customer
9 plantuml::{moduliths}/module-customer.puml[format=
10 include ::{moduliths}/module-customer.adoc[]
11
12 == Inventory
13 plantuml::{moduliths}/module-inventory.puml[format=
14 include ::{moduliths}/module-inventory.adoc[]
15
16 == Orders
17 plantuml::{moduliths}/module-order.puml[format=sv
18 include ::{moduliths}/module-order.adoc[]
19

```

Inventory

Component: Module+ Inventory

| | |
|-------------------|---------------------------|
| Base package | example.inventory |
| Spring components | Services |
| | • e.i.Inventory |
| | Repositories |
| | • e.i.InventoryRepository |

Figure 3. Inventory component overview

Orders

Component: Module+ Order

Order

Installed asciidoctor: Processing file:/Users/odrothohn/Documents/workspace/arch-evident-spring/src/main/asciidoc/index.adoc

```

16 package example.order;
17
18+ import example.customer.Customer.CustomerIdentifier;
19
20+ /**
21 * @author Oliver Drotbohm
22 */
23 @Transactional
24 @Service
25 @RequiredArgsConstructor
26 public class OrderManagement {
27
28     public Order create(CustomerIdentifier customerId) {
29         return new Order(customerId);
30     }
31
32     public Order complete(Order order) {
33         return order;
34     }
35
36 }
37
38 }
39 }
40

```

```

18+ import example.customer.Customer.CustomerIdentifier;
19 import example.inventory.InventoryRepository;
20 import jakarta.transaction.Transactional;
21 import lombok.RequiredArgsConstructor;
22
23 import org.jmoleculesddd.annotation.Service;
24
25+ /**
26 * @author Oliver Drotbohm
27 */
28 @Transactional
29 @Service
30 @RequiredArgsConstructor
31 public class OrderManagement {
32
33     private final InventoryRepository repository;
34
35     public Order create(CustomerIdentifier customerId) {
36         return new Order(customerId);
37     }
38
39     public Order complete(Order order) {
40         return order;
41     }
42 }
43

```

```

16 package example.inventory;
17
18 import org.springframework.stereotype.Repository;
19
20+ /**
21 * @author Oliver Drotbohm
22 */
23 @Repository
24 public class InventoryRepository {
25
26 }
27

```

```

16 package example.inventory;
17
18 import org.springframework.stereotype.Repository;
19
20+ /**
21 * @author Oliver Drotbohm
22 */
23 @Repository
24 public class InventoryRepository {
25
26 }
27

```

Make the class private so that it can't be accessed from another package, since we have arranged the code by domain

Project Explorer X Package Explorer

```

import example.customer.CustomerIdentifier;
import example.inventory.InventoryRepository;
import jakarta.transaction.Transactional;
import lombok.RequiredArgsConstructor;
import org.jmoleculesddd.annotation.Service;

/**
 * @author Oliver Drotbohm
 */
@Transactional
@Service
@RequiredArgsConstructor
public class OrderManagement {

    private final InventoryRepository repository;

    public Order create(CustomerIdentifier customerId) {
        return new Order(customerId);
    }

    public Order complete(Order order) {
        return order;
    }
}

```

Project Explorer X Package Explorer

```

* Copyright 2022 the original author or authors.
package example.order;

import example.customer.CustomerIdentifier;
import jakarta.transaction.Transactional;
import lombok.RequiredArgsConstructor;
import org.jmoleculesddd.annotation.Service;

/**
 * @author Oliver Drotbohm
 */
@Transactional
@Service
@RequiredArgsConstructor
public class OrderManagement {

    public Order create(CustomerIdentifier customerId) {
        return new Order(customerId);
    }

    public Order complete(Order order) {
        return order;
    }
}

```



Project Explorer X Package Explorer

```

2 * Copyright 2023 the original author or authors.
16 package example.inventory.impl;
17
18 import org.springframework.stereotype.Repository;
19
20 /**
21 * @author Oliver Drotbohm
22 */
23 @Repository
24 public class InventoryRepository {
25
26 }
27

```

Project Explorer X Package Explorer

```

2 * Copyright 2022 the original author or authors.
16 package example;
17
18 import org.junit.jupiter.api.Test;
19
20 /**
21 * @author Oliver Drotbohm
22 */
23 class ModularityTests {
24
25     ApplicationModules modules = ApplicationModules.of(Application.class);
26
27     @Test
28     void verifyModularity() {
29
30         // → Module model
31         modules.forEach(System.out::println);
32
33         // → Trigger verification
34         modules.verify();
35     }
36
37     @Test
38     void renderDocumentation() throws Exception {
39
40         var canvasOptions = CanvasOptions.defaults()
41
42     }
43
44 }
45

```

Problems Error Log Progress Search PLANTUML Call Hierarchy Coverage JUnit boot Dashboard Maven Repositories History Console Properties

Run: 0% Errors: 0 Failures: 0

ModularityTests (Runner: JUnit 5) (0.160 s)

Application.java

```

example.customer
Customer.java
> example.inventory
Inventory.java
> example.inventory.impl
InventoryRepository.java
example.order
Order.java
OrderManagement.java
> src/test/java
> example
ApplicationTests.java
ModularityTests.java
example.inventory
example.order
src/test/resources
Maven Dependencies
JRE System Library [JavaSE-17]
target/generated-sources/annotations
target/generated-test-sources/test-annotations
etc
> src
> main
asciidoc
index.adoc
> test
target
mvnw

```

3 :moduliths: .. / .. / .. / target/spring

4

5 = Overview

6 plantuml:::{moduliths}/components.p

7

8 = Customer

9 plantuml:::{moduliths}/module-custo

10 include :: {moduliths}/module-custo

11

12 = Inventory

13 plantuml:::{moduliths}/module-inven

14 include :: {moduliths}/module-inven

15

16 = Orders

17 plantuml:::{moduliths}/module-order

18 include :: {moduliths}/module-order.

19

Spring 0.0.1-SNAPSHOT [boot] [steps/0 sec/207] - arch-evid

```

1 = Developer documentation
2
3 :moduliths: ../../target/spring-modulith-docs
4
5 = Overview
6 plantuml::{moduliths}/components.puml[format=svg,1]
7
8 = Customer
9 plantuml::{moduliths}/module-customer.puml[format=svg,1]
10 include::{moduliths}/module-customer.adoc[]
11
12 = Inventory
13 plantuml::{moduliths}/module-inventory.puml[format=svg,1]
14 include::{moduliths}/module-inventory.adoc[]
15
16 = Orders
17 plantuml::{moduliths}/module-order.puml[format=svg,1]
18 include::{moduliths}/module-order.adoc[]
19

```

Aggregate roots * e.c.customer

Inventory

Inventory

Component: Module
Inventory

Figure 3. Inventory component overview

| | |
|-------------------|-------------------|
| Base package | example.inventory |
| Spring components | Services |
| • e.i.Inventory | |

Orders

Order

Component: Module
Order

Landing [MathJax]/extensions/MathZoom.js

Problems Error Log Progress Search PlantUML Call Hierarchy Coverage JUnit Boot Dashboard Maven Repositories History Console Properties

2023-03-16 11:51:23.381 java[66098:15864381] CoreText note: Set a breakpoint on CTFontLogSystemFontNameRequest to debug.

2023-03-16 11:51:23.383 java[66098:15864381] CoreText note: Client requested name ".Helvetica Neue DeskInterface-UltralightP2", it will get "Ultralight" as result.

2023-03-16 11:51:23.383 java[66098:15864381] CoreText note: Client requested name ".Helvetica Neue DeskInterface-Thin", it will get "Thin" as result.

2023-03-16 11:51:23.383 java[66098:15864381] CoreText note: Client requested name ".Helvetica Neue DeskInterface-MediumP4", it will get "Medium" as result.

2023-03-16 11:51:23.383 java[66098:15864381] CoreText note: Client requested name ".Helvetica Neue DeskInterface-Bold", it will get "Bold" as result.

2023-03-16 11:51:23.439 java[66098:15864381] CoreText note: Client requested name ".Helvetica Neue DeskInterface-Regular", it will get "Regular" as result.

2023-03-16 11:51:23.439 java[66098:15864381] CoreText note: Client requested name ".Helvetica Neue DeskInterface-Bold", it will get "Bold" as result.

Project Explorer X Package Explorer

Inventory.java InventoryRepository.java OrderManagement.java ModularityTests.java index.adoc

Arch-Evident-Spring

src/main/java
 - example
 - Application.java
 - Customer.java
 - Inventory.java
 - example.inventory.impl
 - InventoryRepository.java
 - Order.java
 - OrderManagement.java
 - example.inventory
 - example.order
 - ApplicationTests.java
 - ModularityTests.java
 - example.inventory
 - example.order
 - resources
 - src
 - main
 - esodoc
 - index.adoc
 - test
 - OrderManagementTest.java
 - target
 - pom.xml
 - resources
 - moduliths
 - source-modulith 0.0.1-SNAPSHOT [main - spring-modulith]
 - Spring HATEOAS [main - spring-hateoas]
 - Other Projects

2+ * Copyright 2022 the original author or authors.
16 package example;
17
18+ import org.junit.jupiter.api.Test;
24
25 /**
26 * @author Oliver Drotbohm
27 */
28 class ModularityTests {
29
30 ApplicationModules modules = ApplicationModules.of(Application.class);
31
32+ @Test
33 void verifyModularity() {
34
35 // → Module model
36 modules.forEach(System.out::println);
37 }

Problems Error Log Progress Search PlantUML Call Hierarchy Coverage JUnit Boot Dashboard Maven Repositories History Console Properties

Run: 1/1 0 Errors: 1 0 Failures: 0

Failure Trace

1. org.springframework.modulith.core.Violations - Module "order" depends on non-exported type example.inventory.impl.InventoryRepository within module "inventory". OrderManagement declares constructor OrderManagement(OrderManagementInventoryRepository) in OrderManagement.java(5)
 - Module "order" depends on non-exported type example.inventory.impl.InventoryRepository within module "inventory".
 - Module "order" depends on non-exported type example.inventory.impl.InventoryRepository in OrderManagement.java(0)
 - Module "order" depends on non-exported type example.inventory.impl.InventoryRepository within module "inventory".
 - Constructor <example.order.OrderManagement> in example.inventory.impl.InventoryRepository in OrderManagement.java(5)
 at org.springframework.modulith.core.Violations.verifyViolations(Violations.java:151)
 at java.base/java.util.stream.ReduceOps\$3ReduceSink.accept(ReduceOps.java:80)
 at java.base/java.util.stream.ReferencePipeline\$3\$1.accept(ReferencePipeline.java:197)
 at java.base/java.util.HashMap\$ValueIterator.forEachRemaining(HashMap.java:179)
 at java.base/java.util.stream.AbstractPipeline.copyInto(AbstractPipeline.java:490)
 at java.base/java.util.stream.ReduceOps\$ReduceOp.evaluateSequential(ReduceOps.java:91)
 at java.base/java.util.stream.AbstractPipeline.evaluate(AbstractPipeline.java:254)
 at java.base/java.util.stream.ReferencePipeline.reduce(ReferencePipeline.java:657)
 at org.springframework.modulith.core.ApplicationModules.detectViolations(ApplicationModules.java:379)
 at org.springframework.modulith.core.ApplicationModules.verify(ApplicationModules.java:344)
 at org.springframework.modulith.core.ApplicationModules.verifyModularity(ApplicationModules.java:344)
 at java.base/java.lang.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
 at java.base/java.lang.reflect.NativeMethodAccessorImpl.invoke(NativeMethod.invoke)
 at java.base/java.lang.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
 at java.base/java.lang.reflect.Method.invoke(Method.java:661)
 at org.junit.platform.commons.util.ReflectionUtils.invokeMethod(ReflectionUtils.java:727)
 at org.junit.jupiter.engine.execution.MethodInvocation.proceed(MethodInvocation.java:60)

work.modulith.core.Violations: - Module 'order' depends on non-exposed type example.inventory.impl.InventoryRepository within module 'inventory'!

```

eclares constructor OrderManagement(InventoryRepository) in (OrderManagement.java:0)
ends on non-exposed type example.inventory.impl.InventoryRepository within module 'inventory'
r.OrderManagement.repository> has type <example.inventory.impl.InventoryRepository> in (OrderManagement.java:0)
ends on non-exposed type example.inventory.impl.InventoryRepository within module 'inventory'
le.order.OrderManagement.<init>(example.inventory.impl.InventoryRepository) has parameter of type <example.inventory.impl.InventoryRepository>
ework.modulith.core.Violations.and(Violations.java:115)
.util.stream.ReduceOps$1ReducingSink.accept(ReduceOps.java:80)
.util.stream.ReferencePipeline$3$1.accept(ReferencePipeline.java:197)
.util.HashMap$ValueSpliterator.forEachRemaining(HashMap.java:1779)

```

Project Explorer X Package Explorer

```

37
38     // → Trigger verification
39     modules.verify();
40 }
41
42 @Test
43 void renderDocumentation() throws Exception {
44     var canvasOptions = CanvasOptions.defaults()
45
46     // → Optionally enable linking of JavaDoc
47     // .withApiBase("https://foobar.something")
48
49 ;
50
51
52 var docOptions = DiagramOptions.defaults()
53     .withStyle(DiagramStyle.UML);

```

Problems Error Log Progress Search Platform Call Hierarchy Coverage JUnit Boot Dashboard Maven Repositories History Console Properties

Failure Trace

work.modulith.core.Violations: - Module 'order' depends on non-exposed type example.inventory.impl.InventoryRepository within module 'inventory'

OrderManagement declares constructor OrderManagement(InventoryRepository) in (OrderManagement.java:0)

Module 'order' depends on non-exposed type example.inventory.impl.InventoryRepository within module 'inventory'

Field <example.order.OrderManagement>.repository has type <example.inventory.impl.InventoryRepository> in (OrderManagement.java:0)

Module 'order' depends on non-exposed type example.inventory.impl.InventoryRepository within module 'inventory'

Constructor <example.order.OrderManagement>.<init>(example.inventory.impl.InventoryRepository) has parameter of type <example.inventory.impl.InventoryRepository> in (OrderManagement.java:0)

at example.order.OrderManagement.<init>(example.inventory.impl.InventoryRepository) (OrderManagement.java:0)

at java.base/java.util.stream.ReduceOps\$1ReducingSink.accept(ReduceOps.java:80)

at java.base/java.util.stream.ReferencePipeline\$3\$1.accept(ReferencePipeline.java:197)

at java.base/java.util.stream.AbstractPipeline\$Terminator.evaluate(AbstractPipeline.java:509)

at java.base/java.util.stream.AbstractPipeline.wrapAndCopyInto(AbstractPipeline.java:499)

at java.base/java.util.stream.ReduceOps\$ReduceOp.evaluateSequential(ReduceOps.java:921)

at java.base/java.util.stream.AbstractPipeline.evaluate(AbstractPipeline.java:234)

at java.base/java.util.stream.ReferencePipeline.collect(ReferencePipeline.java:577)

at org.springframework.modulith.core.ApplicationModules.detectViolations(ApplicationModules.java:378)

at org.springframework.modulith.core.ApplicationModules.verify(ApplicationModules.java:344)

at example.ModularityTests.verifyModularityTests(ModularityTests.java:39)

at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method)

at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:77)

at java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)

at java.base/java.lang.reflect.Method.invoke(Method.java:160)

at org.junit.platform.common.util.ReflectionUtils.invokeMethod(ReflectionUtils.java:727)

at org.junit.jupiter.engine.execution.MethodInvocation.proceed(MethodInvocation.java:69)

Project Explorer X Package Explorer

Arch-Evident-Spring 0.0.1-SNAPSHOT [root] [steps/0 Sec2d7] - arch-evident-spring

```

1 = Developer documentation
2
3 :moduliths: ../../..../target/spring-modulith-docs
4
5 = Overview
6 plantuml::{moduliths}/components.puml[format=svg,1] Orders
7
8 = Customer
9 plantuml::{moduliths}/module-customer.puml[format=svg,1]
10 include::{moduliths}/module-customer.adoc[]
11
12 = Inventory
13 plantuml::{moduliths}/module-inventory.puml[format=svg,1]
14 include::{moduliths}/module-inventory.adoc[]
15
16 = Orders
17 plantuml::{moduliths}/module-order.puml[format=svg,1]
18 include::{moduliths}/module-order.adoc[]

```

Base package example.order

Figure 4. Order component overview

```

# Customer
> Logical name: customer
> Base package: example.customer
> Spring beans: none

# Order
> Logical name: order
> Base package: example.order
> Spring beans:
+ ....OrderManagement

# Inventory
> Logical name: inventory
> Base package: example.inventory
> Spring beans:
+ ....Inventory
+ ....impl.InventoryRepository

```

Project Explorer X Package Explorer

Arch-Evident-Spring 0.0.1-SNAPSHOT [root] [steps/0 Sec2d7] - arch-evident-spring

```

1 package example;
2
3 import org.springframework.boot.SpringApplication;
4
5 /**
6 * Spring for the architecturally curious developer
7 */
8 * @author Oliver Drotbohm (@odrotbohm)
9 */
10 @SpringBootApplication
11 @ConfigurationPropertiesScan
12 public class Application {
13
14     public static void main(String[] args) {
15         SpringApplication.run(Application.class, args);
16     }
17 }

```

Project Explorer X Package Explorer

Arch-Evident-Spring 0.0.1-SNAPSHOT [root] [steps/0 Sec2d7] - arch-evident-spring

```

2+ * Copyright 2022 the original author or authors.
3+
4 package example.inventory;
5
6 import org.junit.jupiter.api.Test;
7
8 /**
9 */
10 * @author Oliver Drotbohm
11 */
12 @ApplicationModuleTest
13 class InventoryIntegrationTests {
14
15     @Test
16     void bootstrapsInventoryModule() {
17
18     }
19 }

```

```
# Problems ⚠ Error Log ⚡ Progress ⚡ Search ⚡ PlanUML ⚡ Call Hierarchy ⚡ Coverage ⚡ JUnit ⚡ Boot Dashboard ⚡ Maven Repositories ⚡ History ⚡ Console X ⚡ Properties
terminated: InventoryIntegrationTests [1] [/Users/odrotbohm/Library/Java/JavaVirtualMachines/jdk-17/Contents/Home/bin/java] [16 Mar 2023, 11:57:48 - 11:57:51 (pid: 67186)]
# 8,888 |-INFO in ch.qos.logback.core.model.processor.ImplicitModelHandler - Assuming default type [ch.qos.logback.classic.encoder.PatternLayout]
# 8,903 |-INFO in ch.qos.logback.classic.model.processor.RootLoggerModelHandler - Setting level of ROOT logger to INFO
# 8,903 |-INFO in ch.qos.logback.core.model.processor.AppenderRefModelHandler - Attaching appender named [CONSOLE] to Logger[ROOT]
# 8,903 |-INFO in ch.qos.logback.classic.model.processor.LoggerModelHandler - Setting level of logger [com.tngtech.archunit] to WARN
# 8,903 |-INFO in ch.qos.logback.classic.model.processor.LoggerModelHandler - Setting level of logger [org.springframework.modulith] to DEBUG
# 8,903 |-INFO in ch.qos.logback.core.model.processor.DefaultProcessor@2bd08376 - End of configuration.
# 8,904 |-INFO in ch.qos.logback.classic.joran.JoranConfigurator@e70f13a - Registering current configuration as safe fallback point

8,939 -     main : Could not detect default configuration classes for test class [example.inventory.InventoryIntegrationTests]: InventoryIntegrationTests
8,987 -     main : Found @SpringBootConfiguration example.Application for test class example.inventory.InventoryIntegrationTests

```
 ing Boot :: (v3.1.0-M1)
```
9,749 -     main : Bootstrapping @org.springframework.modulith.test.ApplicationModuleTest for Inventory in mode STANDALONE (class example.
9,749 -     main :
9,750 -     main : # Inventory
9,750 -     main : > Logical name: inventory
9,750 -     main : > Base package: example.inventory
9,750 -     main : > Direct module dependencies: none
9,750 -     main : > Spring beans:
9,751 -     main : + ...Inventory
9,751 -     main : + ...impl.InventoryRepository
9,751 -     main :
9,754 -     main : Starting InventoryIntegrationTests using Java 17.0.6 with PID 67186 (started by odrotbohm in /Users/odrotbohm/Documents)
9,755 -     main : No active profile set, falling back to 1 default profile: "default"
9,898 -     main : Re-configuring auto-configuration and entity scan packages to: example.inventory.
9,923 -     main : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
9,933 -     main : Finished Spring Data repository scanning in 7 ms. Found 0 JPA repository interfaces.
10,079 -    main : HikariPool-1 - Starting...
10,146 -    main : HikariPool-1 - Added connection conn0: url=jdbc:h2:mem:7bc320e6-87f6-4674-aaeb-3f40e60ee116 user=SA
10,147 -    main : HikariPool-1 - Start completed.
10,165 -    main : HHH0000204: Processing PersistenceUnitInfo [name: default+]
```

```
# Problems ⚠ Error Log ⚡ Progress ⚡ Search ⚡ PlanUML ⚡ Call Hierarchy ⚡ Coverage ⚡ JUnit ⚡ Boot Dashboard ⚡ Maven Repositories ⚡ History ⚡ Console X ⚡ Properties
terminated: InventoryIntegrationTests [1] [/Users/odrotbohm/Library/Java/JavaVirtualMachines/jdk-17/Contents/Home/bin/java] [16 Mar 2023, 11:57:48 - 11:57:51 (pid: 67186)]
# 49,751 -     main : + ...Inventory
# 49,751 -     main : + ...impl.InventoryRepository
# 49,751 -     main :
# 49,754 -     main : Starting InventoryIntegrationTests using Java 17.0.6 with PID 67186 (started by odrotbohm in /Users/odrotbohm/Documents)
# 49,755 -     main : No active profile set, falling back to 1 default profile: "default"
# 49,898 -     main : Re-configuring auto-configuration and entity scan packages to: example.inventory.
# 49,923 -     main : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
# 49,933 -     main : Finished Spring Data repository scanning in 7 ms. Found 0 JPA repository interfaces.
# 50,079 -    main : HikariPool-1 - Starting...
# 50,146 -    main : HikariPool-1 - Added connection conn0: url=jdbc:h2:mem:7bc320e6-87f6-4674-aaeb-3f40e60ee116 user=SA
# 50,147 -    main : HikariPool-1 - Start completed.
# 50,165 -    main : HHH0000204: Processing PersistenceUnitInfo [name: default]
# 50,185 -    main : HHH0000412: Hibernate ORM core version 6.2.0.CR3
# 50,186 -    main : HHH0000406: Using bytecode reflection optimizer
# 50,233 -    main : HHH000021: Bytecode provider name : bytebuddy
# 50,283 -    main : No LoadTimeWeaver setup: ignoring JPA class transformer
# 50,303 -    main : HHH0000400: Using dialect: org.hibernate.dialect.H2Dialect
# 50,439 -    main : HHH000021: Bytecode provider name : bytebuddy
# 50,686 -    main : HHH0000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
# ate: drop table if exists jpa_event_publication cascade
# ate: create table jpa_event_publication (completion_date timestamp(6) with time zone, publication_date timestamp(6) with time zone, id uuid)
# 50,699 -    main : Initialized JPA EntityManagerFactory for persistence unit 'default'
# 50,735 -    main : Defaulting async shutdown to await termination in 2 seconds.
# 50,785 -  ecutor-1 : Obtaining Spring Modulith application modules...
# 50,788 -    main : Looking up previously pending event publications...
# 50,926 -  ecutor-1 : Detected 3 application modules: [customer, order, inventory]
# ate: select j1_0.id,j1_0.completion_date,j1_0.event_type,j1_0.listener_id,j1_0.publication_date,j1_0.serialized_event from jpa_event_publi
# 50,962 -    main : None found.
# 50,968 -    main : No TaskScheduler/ScheduledExecutorService bean found for scheduled processing
# 50,978 -    main : Started InventoryIntegrationTests in 1.961 seconds (process running for 2,458)
# ate: select j1_0.id,j1_0.completion_date,j1_0.event_type,j1_0.listener_id,j1_0.publication_date,j1_0.serialized_event from jpa_event_publi
# 51,058 -  downHook : No publications outstanding!
# 51,059 -  downHook : Closing JPA EntityManagerFactory for persistence unit 'default'
# 51,059 -  downHook : HHH000477: Starting delayed evictData of schema as part of SessionFactory shut-down'
# ate: drop table if exists jpa_event_publication cascade
# 51,060 -  downHook : HikariPool-1 - Shutdown initiated ...
# 51,061 -  downHook : HikariPool-1 - Shutdown completed.
```

```
23
24 /**
25 * @author Oliver Drotbohm
26 */
27 @Transactional
28 @Service
29 @RequiredArgsConstructor
30 public class OrderManagement {
31     ...
32     public Order create(CustomerIdentifier customerId) {
33         return new Order(customerId);
34     }
35
36     public Order complete(Order order) {
37         return order;
38     }
39 }
40
```

```
22
23 import org.jmolecules.ddd.annotation.Service;
24
25 /**
26 * @author Oliver Drotbohm
27 */
28 @Transactional
29 @Service
30 @RequiredArgsConstructor
31 public class OrderManagement {
32     ...
33     private final Inventory inventory;
34
35     public Order create(CustomerIdentifier customerId) {
36         return new Order(customerId);
37     }
38
39     public Order complete(Order order) {
40 }
```

```
33
34     void verifyModularity() {
35
36         // → Module model
37         modules.forEach(System.out::println);
38
39         // → Trigger verification
40         modules.verify();
41     }
42
43     @Test
44     void renderDocumentation() throws Exception {
45
46         var canvasOptions = CanvasOptions.defaults()
47
48         // → Optionally enable linking of JavaDoc
49         // .withApiBase("https://foobar.something")
50     }
51 }
```

Developer documentation

```

1 = Developer documentation
2
3 :moduliths: ../../target/spring-modulith-docs
4
5 == Overview
6 plantuml ::{moduliths}/components.puml[format=svg,1
7
8 == Customer
9 plantuml ::{moduliths}/module-customer.puml[format=
10 include ::{moduliths}/module-customer.adoc[]
11
12 == Inventory
13 plantuml ::{moduliths}/module-inventory.puml[format=
14 include ::{moduliths}/module-inventory.adoc[]
15
16 == Orders
17 plantuml ::{moduliths}/module-order.puml[format=sv
18 include ::{moduliths}/module-order.adoc[]

```

Application

Customer

Figure 1. Overall component overview

Problems Error Log Progress Search Call Hierarchy Coverage JUnit Boot Dashboard Maven Repositories History Console Properties

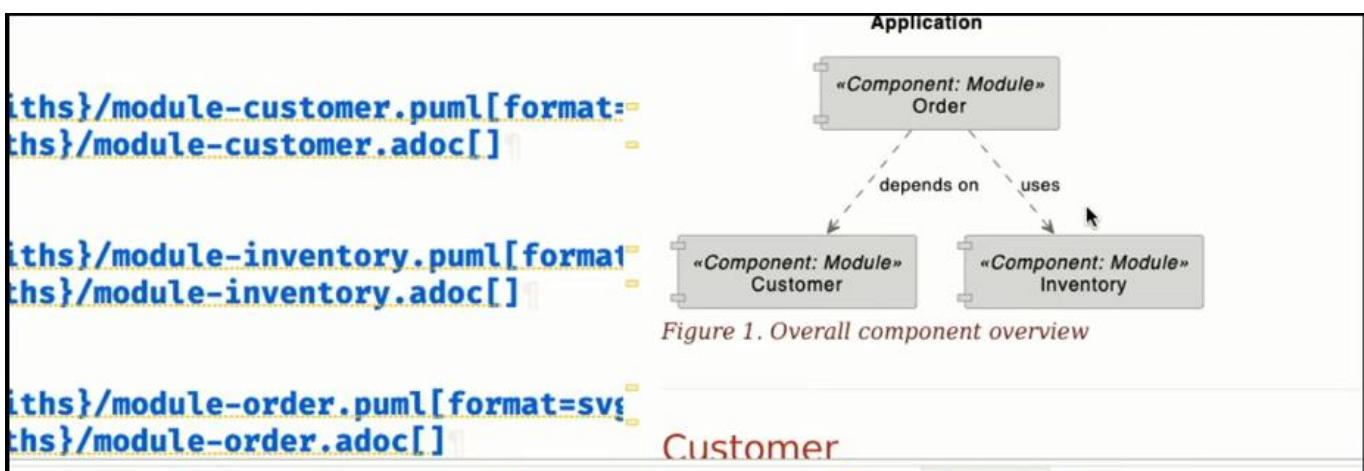
2023-03-16 11:51:23.383 java[66098:15864381] CoreText note: Client requested name ".Helvetica Neue DeskInterface-MediumP4", it will get T
2023-03-16 11:51:23.383 java[66098:15864381] CoreText note: Client requested name ".Helvetica Neue DeskInterface-Bold", it will get T
2023-03-16 11:51:23.439 java[66098:15864381] CoreText note: Client requested name ".Helvetica Neue DeskInterface-Regular", it will get T
2023-03-16 11:51:23.439 java[66098:15864381] CoreText note: Client requested name ".Helvetica Neue DeskInterface-Bold", it will get T

Installed asciidoctor: Processing file:/Users/odrotbohm/Documents/workspace/arch-evident-spring/src/main/asciidoc/index.adoc

Installed asciidoctor: Processing file:/Users/odrotbohm/Documents/workspace/arch-evident-spring/src/main/asciidoc/index.adoc

Installed asciidoctor: Processing file:/Users/odrotbohm/Documents/workspace/arch-evident-spring/src/main/asciidoc/index.adoc

2023-03-16 11:59:27.928 java[67574:15876898] CoreText note: Client requested name ".Helvetica Neue DeskInterface-Regular", it will get T
2023-03-16 11:59:27.928 java[67574:15876898] CoreText note: Set breakpoint on CTFFontLogSystemFontNameRequest to debug.
2023-03-16 11:59:27.929 java[67574:15876898] CoreText note: Client requested name ".Helvetica Neue DeskInterface-UltralightP2", it will get T
2023-03-16 11:59:27.929 java[67574:15876898] CoreText note: Client requested name ".Helvetica Neue DeskInterface-Thin", it will get T
2023-03-16 11:59:27.930 java[67574:15876898] CoreText note: Client requested name ".Helvetica Neue DeskInterface-MediumP4", it will get T
2023-03-16 11:59:27.930 java[67574:15876898] CoreText note: Client requested name ".Helvetica Neue DeskInterface-Bold", it will get T
2023-03-16 11:59:27.987 java[67574:15876898] CoreText note: Client requested name ".Helvetica Neue DeskInterface-Regular", it will get T
2023-03-16 11:59:27.987 java[67574:15876898] CoreText note: Client requested name ".Helvetica Neue DeskInterface-Bold", it will get T



Customer

Figure 1. Overall component overview

Problems Error Log Progress Search Call Hierarchy Coverage JUnit Boot Dashboard Maven Repositories History Console Properties

2023-03-16 11:51:23.383 java[66098:15864381] CoreText note: Client requested name ".Helvetica Neue DeskInterface-MediumP4", it will get T
2023-03-16 11:51:23.383 java[66098:15864381] CoreText note: Client requested name ".Helvetica Neue DeskInterface-Bold", it will get T
2023-03-16 11:51:23.439 java[66098:15864381] CoreText note: Client requested name ".Helvetica Neue DeskInterface-Regular", it will get T
2023-03-16 11:51:23.439 java[66098:15864381] CoreText note: Client requested name ".Helvetica Neue DeskInterface-Bold", it will get T

Installed asciidoctor: Processing file:/Users/odrotbohm/Documents/workspace/arch-evident-spring/src/main/asciidoc/index.adoc

Installed asciidoctor: Processing file:/Users/odrotbohm/Documents/workspace/arch-evident-spring/src/main/asciidoc/index.adoc

Installed asciidoctor: Processing file:/Users/odrotbohm/Documents/workspace/arch-evident-spring/src/main/asciidoc/index.adoc

2023-03-16 11:59:27.928 java[67574:15876898] CoreText note: Client requested name ".Helvetica Neue DeskInterface-Regular", it will get T
2023-03-16 11:59:27.928 java[67574:15876898] CoreText note: Set breakpoint on CTFFontLogSystemFontNameRequest to debug.
2023-03-16 11:59:27.929 java[67574:15876898] CoreText note: Client requested name ".Helvetica Neue DeskInterface-UltralightP2", it will get T
2023-03-16 11:59:27.929 java[67574:15876898] CoreText note: Client requested name ".Helvetica Neue DeskInterface-Thin", it will get T
2023-03-16 11:59:27.930 java[67574:15876898] CoreText note: Client requested name ".Helvetica Neue DeskInterface-MediumP4", it will get T
2023-03-16 11:59:27.930 java[67574:15876898] CoreText note: Client requested name ".Helvetica Neue DeskInterface-Bold", it will get T
2023-03-16 11:59:27.987 java[67574:15876898] CoreText note: Client requested name ".Helvetica Neue DeskInterface-Regular", it will get T
2023-03-16 11:59:27.987 java[67574:15876898] CoreText note: Client requested name ".Helvetica Neue DeskInterface-Bold", it will get T

anDefinitionException: No qualifying bean of type 'example.inventory.Inventory' available: expected at least 1 bean which qualifies as autowitable. Did you mean 'example.Inventory'?
DefaultListableBeanFactory.raiseNoMatchingBeanFound(DefaultListableBeanFactory.java:1812)
DefaultListableBeanFactory.doResolveDependency(DefaultListableBeanFactory.java:1371)
DefaultListableBeanFactory.resolveDependency(DefaultListableBeanFactory.java:1325)
ConstructorResolver.resolveAutowiredArgument(ConstructorResolver.java:885)
ConstructorResolver.createArgumentArray(ConstructorResolver.java:789)

Project Explorer X Package Explorer

```

src/main/java
  -> Arch-Evident-Spring
    -> arch-evident-spring
      -> example
        -> Application.java
        -> Customer.java
        -> example.inventory
          -> Inventory.java
          -> InventoryRepository.java
        -> example.order
          -> Order.java
          -> OrderManagement.java
      -> sqnTests.java
        -> ApplicationTests.java
        -> ModularityTests.java
      -> example.inventory
        -> InventoryIntegrationTests.java
      -> example.order
        -> OrderIntegrationTests.java
      -> OrderManagementTests.java
      -> sqnTestResources
      -> Maven Dependencies
      -> JRE System Library [JavaSE-17]
      -> target/generated-sources
      -> target/generated-test-sources/test-annotations
    -> etc
    -> main
      -> exclude
        -> index.adoc
  -> target
  -> mvnw
  -> mvnw.cmd

```

```

Inventory.java  InventoryRepository.java  OrderManagement.java  ModularityTests.java  index.adoc  InventoryIntegrationTests.java  Application.java  OrderIntegrationTests.java
16 package example.order;
17
18+ import example.inventory.Inventory;
19
20+ /**
21+ * Copyright 2022 the original author or authors.
22+
23+ */
24
25+ /*
26+ * @author Oliver Drotbohm
27+ */
28 @ApplicationModuleTest
29 @RequiredArgsConstructor
30 class OrderIntegrationTests {
31
32     @MockBean Inventory inventory;
33
34+ @Test
35     void bootstrapsOrderModule() {
36
37     }
38 }
39

```

Project Explorer X Package Explorer

```

src/main/java
  -> Arch-Evident-Spring
    -> arch-evident-spring
      -> example
        -> Application.java
        -> Customer.java
        -> example.inventory
          -> Inventory.java
          -> InventoryRepository.java
        -> example.order
          -> Order.java
          -> OrderManagement.java
      -> sqnTests.java
        -> ApplicationTests.java
        -> ModularityTests.java
      -> example.inventory
        -> InventoryIntegrationTests.java
      -> example.order
        -> OrderIntegrationTests.java
      -> OrderManagementTests.java
      -> sqnTestResources
      -> Maven Dependencies
      -> JRE System Library [JavaSE-17]
      -> target/generated-sources
      -> target/generated-test-sources/test-annotations
    -> etc
    -> main
      -> exclude
        -> index.adoc
  -> target
  -> mvnw
  -> mvnw.cmd

```

```

Inventory.java  InventoryRepository.java  OrderManagement.java  ModularityTests.java  index.adoc  InventoryIntegrationTests.java  Application.java  OrderIntegrationTests.java
20+ * Copyright 2022 the original author or authors.
21+
22+ import lombok.RequiredArgsConstructor;
23+
24+ import org.junit.jupiter.api.Test;
25+ import org.springframework.modulith.test.ApplicationModuleTest;
26+ import org.springframework.modulith.test.ApplicationModuleTest.BootstrapMode;
27+
28+ /**
29+ * @author Oliver Drotbohm
30+
31+ @ApplicationModuleTest(mode = BootstrapMode.DIRECT_DEPENDENCIES)
32+ @RequiredArgsConstructor
33+ class OrderIntegrationTests {
34
35     void bootstrapsOrderModule() {
36
37     }
38 }
39

```

Project Explorer X Package Explorer

```

src/main/java
  -> Arch-Evident-Spring
    -> arch-evident-spring
      -> example
        -> Application.java
        -> Customer.java
        -> example.customer
          -> CustomerManagement.java
        -> example.inventory
          -> Inventory.java
          -> InventoryRepository.java
        -> example.order
          -> Order.java
          -> OrderManagement.java
      -> sqnTests.java
        -> ApplicationTests.java
        -> ModularityTests.java
      -> example.inventory
        -> InventoryIntegrationTests.java
      -> example.order
        -> OrderIntegrationTests.java
      -> OrderManagementTests.java
      -> sqnTestResources
      -> Maven Dependencies
      -> JRE System Library [JavaSE-17]
      -> target/generated-sources/annotations
      -> target/generated-test-sources/test-annotations
    -> etc
    -> main
      -> exclude
        -> index.adoc
  -> target
  -> mvnw
  -> mvnw.cmd
  -> pom.xml
  -> README.adoc
  -> modules
    -> spring-modulith (0.0.1-SNAPSHOT (main - spring-modulith))
    -> Spring-HATEOAS (main - spring-hateoas)
  -> Other Projects

```

```

Inventory.java  InventoryRepository.java  OrderManagement.java  ModularityTests.java  index.adoc  InventoryIntegrationTests.java  Application.java  OrderIntegrationTests.java
20+ * Copyright 2022 the original author or authors.
21+
22+ import lombok.RequiredArgsConstructor;
23+
24+ import org.junit.jupiter.api.Test;
25+ import org.springframework.modulith.test.ApplicationModuleTest;
26+ import org.springframework.modulith.test.ApplicationModuleTest.BootstrapMode;
27+
28+ /**
29+ * @author Oliver Drotbohm
30+
31+ @ApplicationModuleTest(mode = BootstrapMode.DIRECT_DEPENDENCIES)
32+ @RequiredArgsConstructor
33+ class OrderIntegrationTests {
34
35     void bootstrapsOrderModule() {
36
37     }
38 }
39

```

Spring Boot :: (v3.1.0-M1)

```

2:01:44.405 -      main : Bootstrapping @org.springframework.modulith.test.ApplicationModuleTest for Order integration tests
2:01:44.405 -      main :
2:01:44.407 -      main : # Order
2:01:44.407 -      main : > Logical name: order
2:01:44.407 -      main : > Base package: example.order
2:01:44.407 -      main : > Direct module dependencies: inventory
2:01:44.407 -      main : > Spring beans:
2:01:44.407 -      main : + ...OrderManagement
2:01:44.407 -      main :
2:01:44.407 -      main : Included dependencies:
2:01:44.408 -      main :
2:01:44.408 -      main : # Inventory
2:01:44.408 -      main : > Logical name: inventory
2:01:44.408 -      main : > Base package: example.inventory
2:01:44.408 -      main : > Direct module dependencies: none
2:01:44.408 -      main : > Spring beans:
2:01:44.408 -      main : + ...Inventory

```

```

package example.order;

import lombok.RequiredArgsConstructor;
import org.junit.jupiter.api.Test;
import org.springframework.modulith.test.ApplicationModuleTest;
import org.springframework.modulith.test.ApplicationModuleTest.BootstrapMode;

/**
 * Copyright 2022 the original author or authors.
 */
import lombok.RequiredArgsConstructor;
import org.junit.jupiter.api.Test;
import org.springframework.modulith.test.ApplicationModuleTest;
import org.springframework.modulith.test.ApplicationModuleTest.BootstrapMode;

/**
 * @author Oliver Drotbohm
 */
@ApplicationModuleTest(mode = BootstrapMode.DIRECT_DEPENDENCIES)
@RequiredArgsConstructor
class OrderIntegrationTests {
}

```

```

arch-evident-spring @ (5ec2d71...) $ 
arch-evident-spring @ (5ec2d71...) $ git co steps/2
Previous HEAD position was 5ec2d71 Spring Modulith project skeleton.
HEAD is now at 6734c56 Break module cycle using events.
arch-evident-spring @ (6734c56...) $

```

```

import com.modulith.ACCESSLEVEL;
import lombok.Value;
import lombok.RequiredArgsConstructor;
import org.springframework.modulith.ConfigurationProperties;
import org.springframework.modulith.OnConstructorBinding;

class InventorySettings {
    int stockThreshold;
}

main :
main : Included dependencies:
main :
main : # Inventory
main : > Logical name: inventory
main : > Base package: example.inventory
main : > Direct module dependencies: none
main : > Spring beans:
main : + ...Inventory
main : + ...impl.InventoryRepository
main :
main : Starting OrderIntegrationTests using Java 17.0.6 with PID 67953 (started by odrotbohm in /Users/o
main : No active profile set, falling back to 1 default profile: "default"
main : Re-configuring auto-configuration and entity scan packages to: example.order, example.inventory.
main : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
main : Finished Spring Data repository scanning in 11 ms. Found 0 JPA repository interfaces.
main : HikariPool-1 - Starting...
main : HikariPool-1 - Added connection conn0: url=jdbc:h2:mem:240db9ab-3a7f-4384-b2fb-43237bfdb506 user=
main : HikariPool-1 - Start completed.

```

Project Explorer X Package Explorer

```

arch-evident-spring 0.0.1-SNAPSHOT [root] [steps/0.8734c56 · arch-evident]
  +-- example
    +-- Application.java
    +-- example.customer
      +-- Customer.java
    +-- example.inventory
      +-- Inventory.java
      +-- InventoryRepository.java
    +-- example.settings
      +-- InventorySettings.java
    +-- example.order
      +-- Order.java
      +-- OrderManagement.java
      +-- OrderRepository.java
  +-- src/test/java
    +-- example
      +-- ApplicationTests.java
      +-- ModularityTests.java
    +-- example.inventory
      +-- InventoryIntegrationTests.java
    +-- example.order
      +-- OrderIntegrationTests.java
      +-- OrderRepositoryTests.java
  +-- .mvn
  +-- Maven Dependencies
  +-- JRE System Library [JavaSE-17]
  +-- target/generated-sources/annotations
  +-- target/generated-test-sources/test-annotations
  +-- test
  +-- target

```

InventorySettings.java [ModularityTests.java]

```

2 * Copyright 2022 the original author or authors.
16 package example;
17
18 import org.junit.jupiter.api.Test;
19
20 /**
21  * @author Oliver Drotbohm
22 */
23 class ModularityTests {
24
25     ApplicationModules modules = ApplicationModules.of(Application.class);
26
27     @Test
28     void verifyModularity() {
29
30         // → Module model
31
32     }
33 }

```

Problems Error Log Progress Search Part Hierarchy Call Hierarchy Coverage JUnit Root Dashboard Maven Repositories History Console Properties Run: 2/2 Errors: 0 Failures: 0 Failure Trace

Project Explorer X Package Explorer

```

arch-evident-spring 0.0.1-SNAPSHOT [root] [steps/0.8734c56 · arch-evident]
  +-- src/main/java
    +-- example
      +-- Application.java
      +-- example.customer
        +-- Customer.java
      +-- example.inventory
        +-- Inventory.java
        +-- InventoryRepository.java
      +-- example.settings
        +-- InventorySettings.java
    +-- example.order
      +-- Order.java
      +-- OrderManagement.java
      +-- OrderRepository.java
  +-- src/test/java
    +-- example
      +-- ApplicationTests.java
      +-- ModularityTests.java
    +-- example.inventory
      +-- InventoryIntegrationTests.java
    +-- example.order
      +-- OrderIntegrationTests.java
      +-- OrderRepositoryTests.java
  +-- .mvn
  +-- Maven Dependencies
  +-- JRE System Library [JavaSE-17]
  +-- target/generated-sources/annotations
  +-- target/generated-test-sources/test-annotations
  +-- test
  +-- target

```

InventorySettings.java [ModularityTests.java]

```

1 = Developer documentation
2
3 :moduliths: ../../target/spring-modulith-docs
4
5 = Overview
6 plantuml::{moduliths}/components.puml[format=svg,]
7
8 = Customer
9 plantuml::{moduliths}/module-customer.puml[format=svg,]
10 include::{moduliths}/module-customer.adoc[]
11
12 = Inventory
13 plantuml::{moduliths}/module-inventory.puml[format=svg,]
14 include::{moduliths}/module-inventory.adoc[]
15
16 = Orders

```

Problems Error Log Progress Search Part Hierarchy Call Hierarchy Coverage JUnit Root Dashboard Maven Repositories History Console Properties Run: 2/2 Errors: 0 Failures: 0 Failure Trace

InventorySettings.java [ModularityTests.java]

```

1 = Developer documentation
2
3 :moduliths: ../../target/spring-modulith-docs
4
5 = Overview
6 plantuml::{moduliths}/components.puml[format=svg,title="Overall component overview"]
7
8 = Customer
9 plantuml::{moduliths}/module-customer.puml[format=svg,title="Customer component overview"]
10 include::{moduliths}/module-customer.adoc[]
11
12 = Inventory
13 plantuml::{moduliths}/module-inventory.puml[format=svg,title="Inventory component overview"]
14 include::{moduliths}/module-inventory.adoc[]
15
16 = Orders
17 plantuml::{moduliths}/module-order.puml[format=svg,title="Order component overview"]
18 include::{moduliths}/module-order.adoc[]
19

```

Developer documentation

Overview

Figure 1. Overall component overview

Customer

Figure 2. Customer component overview

| | |
|-----------------|------------------|
| Base package | example.customer |
| Aggregate roots | e.c.Customer |

Inventory

Figure 3. Inventory component overview

InventorySettings.java

```

1 = Developer documentation
2
3 :moduliths: ../../target/spring-modulith-docs
4
5 == Overview
6 plantuml:::{moduliths}/components.puml[format=svg,title="Overall component overview"]
7
8 == Customer
9 plantuml:::{moduliths}/module-customer.puml[format=svg,title="Customer component overview"]
10 include:::{moduliths}/module-customer.adoc[]
11
12 == Inventory
13 plantuml:::{moduliths}/module-inventory.puml[format=svg,title="Inventory component overview"]
14 include:::{moduliths}/module-inventory.adoc[]
15
16 == Orders
17 plantuml:::{moduliths}/module-order.puml[format=svg,title="Order component overview"]
18 include:::{moduliths}/module-order.adoc[]
19

```

Aggregate roots • e.c.Customer

Inventory

Inventory

+Component: Module+
Inventory

Figure 3. Inventory component overview

| | |
|--------------------|---|
| Base package | example.inventory |
| Spring components | Services • e.i.Inventory listening to e.o.OrderCompleted |
| Events listened to | • e.o.OrderCompleted |
| Properties | • example.inventory.stock-threshold — java.lang.Integer, default 0. Some Javadoc. |

Orders

Order

+Component: Module+
Order

Figure 4. Order component overview

We are now using event-based communication between the different modules

Project Explorer X Package Explorer

Arch Evident Spring 0.0.1-SNAPSHOT [boot] [step/0/8734e56 - arch-evident-spring]

```

20 @WeldISDContainer
29 @Service
30 @RequiredArgsConstructor
31 public class OrderManagement {
32
33     private final OrderRepository orders;
34
35     public Order create(CustomerIdentifier customerId) {
36         return new Order(customerId);
37     }
38
39     public Order complete(Order order) {
40         return orders.save(order.complete());
41     }
42 }
43

```

Problems Error Log Progress Search Call Hierarchy Coverage JUnit Root Dashboard Maven Repositories History Console Properties

Installed asciidoc: Processing file:/Users/odrotbohm/Documents/workspace/arch-evident-spring/src/main/asciidoc/index.adoc

Installed asciidoc: Processing file:/Users/odrotbohm/Documents/workspace/arch-evident-spring/src/main/asciidoc/index.adoc

2023-03-16 11:59:27.928 [java[67574:15876898]] CoreText note: Client requested name ".Helvetica Neue DeskInterface-Regular", it will get T
2023-03-16 11:59:27.928 [java[67574:15876898]] CoreText note: Set breakpoint on CTFFontLogSystemFontNameRequest to debug.
2023-03-16 11:59:27.929 [java[67574:15876898]] CoreText note: Client requested name ".Helvetica Neue DeskInterface-UltralightP2", it wi
2023-03-16 11:59:27.929 [java[67574:15876898]] CoreText note: Client requested name ".Helvetica Neue DeskInterface-Thin", it will get T
2023-03-16 11:59:27.930 [java[67574:15876898]] CoreText note: Client requested name ".Helvetica Neue DeskInterface-MediumP4", it will ge
2023-03-16 11:59:27.930 [java[67574:15876898]] CoreText note: Client requested name ".Helvetica Neue DeskInterface-Bold", it will get T
2023-03-16 11:59:27.987 [java[67574:15876898]] CoreText note: Client requested name ".Helvetica Neue DeskInterface-Regular", it will ge
2023-03-16 11:59:27.987 [java[67574:15876898]] CoreText note: Client requested name ".Helvetica Neue DeskInterface-Bold", it will get T

Installed asciidoc: Processing file:/Users/odrotbohm/Documents/workspace/arch-evident-spring/src/main/asciidoc/index.adoc

2023-03-16 12:03:45.255 [java[68423:15883904]] CoreText note: Client requested name ".Helvetica Neue DeskInterface-Regular", it will ge
2023-03-16 12:03:45.255 [java[68423:15883904]] CoreText note: Set breakpoint on CTFFontLogSystemFontNameRequest to debug.
2023-03-16 12:03:45.256 [java[68423:15883904]] CoreText note: Client requested name ".Helvetica Neue DeskInterface-UltralightP2", it wi
2023-03-16 12:03:45.256 [java[68423:15883904]] CoreText note: Client requested name ".Helvetica Neue DeskInterface-Thin", it will get T
2023-03-16 12:03:45.256 [java[68423:15883904]] CoreText note: Client requested name ".Helvetica Neue DeskInterface-MediumP4", it will ge
2023-03-16 12:03:45.256 [java[68423:15883904]] CoreText note: Client requested name ".Helvetica Neue DeskInterface-Bold", it will get T
2023-03-16 12:03:45.301 [java[68423:15883904]] CoreText note: Client requested name ".Helvetica Neue DeskInterface-Regular", it will ge
2023-03-16 12:03:45.301 [java[68423:15883904]] CoreText note: Client requested name ".Helvetica Neue DeskInterface-Bold", it will get T

Project Explorer X Package Explorer

```

54 }
55
56 Order complete() {
57     this.status = Status.COMPLETED;
58
59     registerEvent(new OrderCompleted(id));
60
61     return this;
62 }
63
64
65 Order add(LineItem item) {
66
67     this.lineItems.add(item);
68
69     return this;

```

InventorySettings.java ModularityTests.java Index.html OrderManagement.java Order.java

Project Explorer X Package Explorer

```

29 @Service
30 @RequiredArgsConstructor
31 public class OrderManagement {
32
33     private final OrderRepository orders;
34
35     public Order create(CustomerIdentifier customerId) {
36         return new Order(customerId);
37     }
38
39     public Order complete(Order order) {
40         return orders.save(order.complete());
41     }
42 }

```

InventorySettings.java ModularityTests.java Index.html OrderManagement.java Order.java

Project Explorer X Package Explorer

```

28 @RequiredArgsConstructor
29 public class Inventory {
30
31     @SuppressWarnings("unused") //
32     private final InventoryRepository repository;
33
34     /**
35      * Updates the stock for all line items contained in the order.
36      */
37     @ApplicationModuleListener
38     void updateStock(OrderCompleted order) {
39
40     }
41
42 }

```

InventorySettings.java ModularityTests.java Index.html OrderManagement.java Order.java

Project Explorer X Package Explorer

```

44 @Async
45 @Transactional(propagation = Propagation.REQUIRES_NEW)
46 @TransactionalEventListener
47 @Documented
48 @Target({ ElementType.METHOD, ElementType.ANNOTATION_TYPE })
49 @Retention(RetentionPolicy.RUNTIME)
50 public @interface ApplicationModuleListener {
51
52     /**
53      * Whether the transaction to be run for the event listener is supposed to be read-only (default {@link
54      * @AliasFor(annotation = Transactional.class, attribute = "readOnly")})
55     boolean readOnlyTransaction() default false;
56 }

```

InventorySettings.java ModularityTests.java Index.html OrderManagement.java Order.java

Project Explorer X Package Explorer

```

41 * @see <a href="https://docs.spring.io/spring-modulith/docs/current/reference/html/#events.publication
42 *      Modulith Event Publication Registry - Reference Documentation</a>
43 */
44 @Async
45 @Transactional(propagation = Propagation.REQUIRES_NEW)
46 @TransactionalEventListener
47 @Documented
48 @Target({ ElementType.METHOD, ElementType.ANNOTATION_TYPE })
49 @Retention(RetentionPolicy.RUNTIME)
50 public @interface ApplicationModuleListener {
51
52     /**
53      * Whether the transaction to be run for the event listener is supposed to be read-only (default {@link
54      * @AliasFor(annotation = Transactional.class, attribute = "readOnly")})
55     boolean readOnlyTransaction() default false;
56 }

```

InventorySettings.java ModularityTests.java Index.html OrderManagement.java Order.java

Project Explorer X Package Explorer

```

arch-evident-spring 0.0.1-SNAPSHOT [root] [steps/0/8734d56 - arch-evident-spring]
  +-- example
    +-- example.customer
    +-- example.inventory
    +-- example.order
      +-- Order.java
      +-- OrderManagement.java
      +-- OrderRepository.java
    +-- OrderIntegrationTests.java
  +-- src/test/java
    +-- example
      +-- ApplicationTests.java
      +-- ModularityTests.java
    +-- example.inventory
      +-- InventoryIntegrationTests.java
    +-- example.order
      +-- OrderIntegrationTests.java
      +-- OrderIntegrationTests.java
    +-- OrderIntegrationTests.java
  +-- src/test/resources
  +-- Maven Dependencies

```

```

InventorySettings.java  ModularityTests.java  index.adoc  OrderManagement.java  Order.java  Inventory.java  ApplicationModuleListener.class  OrderIntegrationTests.java
2+ * Copyright 2022 the original author or authors.
16 package example.order;
17
18 import static org.assertj.core.api.Assertions.*;
19
20 /**
21  * @author Oliver Drotbohm
22 */
23 @ApplicationModuleTest
24 @RequiredArgsConstructor
25 class OrderIntegrationTests {
26
27     private final OrderManagement orders;
28     private final OrderRepository repository;
29
30     ...
31 }
32
33 */
34
35
36
37
38     private final OrderManagement orders;
39     private final OrderRepository repository;
40
41

```

Project Explorer X Package Explorer

```

arch-evident-spring 0.0.1-SNAPSHOT [root] [steps/0/8734d56 - arch-evident-spring]
  +-- example
    +-- example.customer
    +-- example.inventory
    +-- example.order
      +-- Order.java
      +-- OrderManagement.java
      +-- OrderRepository.java
    +-- OrderIntegrationTests.java
  +-- src/test/java
    +-- example
      +-- ApplicationTests.java
      +-- ModularityTests.java
    +-- example.inventory
      +-- InventoryIntegrationTests.java
    +-- example.order
      +-- OrderIntegrationTests.java
      +-- OrderIntegrationTests.java
    +-- OrderIntegrationTests.java
  +-- src/test/resources
  +-- Maven Dependencies

```

```

InventorySettings.java  ModularityTests.java  index.adoc  OrderManagement.java  Order.java  Inventory.java  ApplicationModuleListener.class  OrderIntegrationTests.java
56
57
58
59
60
61
62 @Test
63 void completionCausesEventPublished(AssertablePublishedEvents events) {
64
65     var order = new Order(new CustomerIdentifier(UUID.randomUUID()));
66
67     orders.complete(order);
68
69     assertThat(events).contains(OrderCompleted.class)
70         .matching(OrderCompleted::id, order.getId());
71 }

```

Problems Error Log Progress Search Maven Help Call Hierarchy Coverage JUnit Test Dashboard Maven Repositories History Console Properties

Installed asciidoctor: Processing file:/Users/odrotbohm/Documents/workspace/arch-evident-spring/src/main/asciidoc/index.adoc

Installed asciidoctor: Processing file:/Users/odrotbohm/Documents/workspace/arch-evident-spring/src/main/asciidoc/index.adoc

2023-03-16 11:59:27.928 java[67574:15876898] CoreText note: Client requested name ".Helvetica Neue Desktop Interface-Regular", it will get T
2023-03-16 11:59:27.928 java[67574:15876898] CoreText note: Set a breakpoint on CTFontLogSystemFontNameRequest to debug.
2023-03-16 11:59:27.929 java[67574:15876898] CoreText note: Client requested name ".Helvetica Neue Desktop Interface-UltralightP2", it will get T
2023-03-16 11:59:27.930 java[67574:15876898] CoreText note: Client requested name ".Helvetica Neue Desktop Interface-MediumP4", it will get T
2023-03-16 11:59:27.930 java[67574:15876898] CoreText note: Client requested name ".Helvetica Neue Desktop Interface-Bold", it will get T
2023-03-16 11:59:27.987 java[67574:15876898] CoreText note: Client requested name ".Helvetica Neue Desktop Interface-Regular", it will get T
2023-03-16 11:59:27.987 java[67574:15876898] CoreText note: Client requested name ".Helvetica Neue Desktop Interface-Bold", it will get T

Installed asciidoctor: Processing file:/Users/odrotbohm/Documents/workspace/arch-evident-spring/src/main/asciidoc/index.adoc

2023-03-16 12:03:45.255 java[68423:15883904] CoreText note: Client requested name ".Helvetica Neue Desktop Interface-Regular", it will get T
2023-03-16 12:03:45.255 java[68423:15883904] CoreText note: Set a breakpoint on CTFontLogSystemFontNameRequest to debug.

Project Explorer X Package Explorer

```

arch-evident-spring 0.0.1-SNAPSHOT [root] [steps/0/8734d56 - arch-evident-spring]
  +-- example
    +-- example.customer
    +-- example.inventory
    +-- example.order
      +-- Order.java
      +-- OrderManagement.java
      +-- OrderRepository.java
    +-- OrderIntegrationTests.java
  +-- src/test/java
    +-- example
      +-- ApplicationTests.java
      +-- ModularityTests.java
    +-- example.inventory
      +-- InventoryIntegrationTests.java
    +-- example.order
      +-- OrderIntegrationTests.java
      +-- OrderIntegrationTests.java
    +-- OrderIntegrationTests.java
  +-- src/test/resources
  +-- Maven Dependencies

```

```

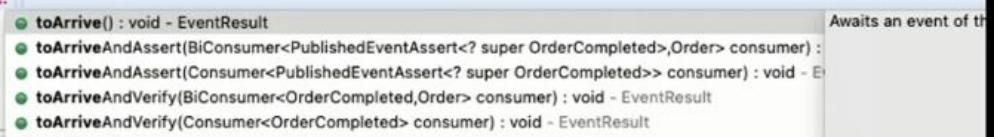
InventorySettings.java  ModularityTests.java  index.adoc  OrderManagement.java  Order.java  Inventory.java  ApplicationModuleListener.class  OrderIntegrationTests.java
82
83
84 @Test
85 void completionCausesEventPublished(Scenario scenario) {
86
87     var order = new Order(new CustomerIdentifier(UUID.randomUUID()));
88
89     scenario.stimulate(() -> orders.complete(order))
90         .andwaitForEventType(OrderCompleted.class)
91         .matchingMappedValue(OrderCompleted::id, order.getId())
92         .toArrive();
93 }
94

```

```
44 » @Test
45 » void completionCausesEventPublished(Scenario scenario)
46 »
47 » » var order = new Order(new CustomerIdentifier(UUID.randomUUID()))
48 »
49 » » scenario.stimulate(() -> orders.complete(order))
50 » » » .andWith()
51 » » » .matching()
52 » » » .toArrive()
53 » }
54 }
55 }
```



```
3 »
4 » @Test
5 » void completionCausesEventPublished(Scenario scenario) {
6 »
7 » » var order = new Order(new CustomerIdentifier(UUID.randomUUID()))
8 »
9 » » scenario.stimulate(() -> orders.complete(order))
10 » » » .andWaitForEventOfType(OrderCompleted.class)
11 » » » .matchingMappedValue(OrderCompleted::id, order.getId())
12 » » » .toArrive()
13 » }
14 }
15 }
```



Problems Error Log Progress Search
stalled asciidoctor: Problem

```

16 package example.inventory;
17
18 import org.junit.jupiter.api.Test;
19
20
21 /**
22  * Copyright 2022 the original author or authors.
23 */
24 @ApplicationModuleTest
25 class InventoryIntegrationTests {
26
27     @Test
28     void bootstrapsInventoryModule() {
29
30     }
31 }

```

Installed asciidoctor: Processing file:/Users/odrotbohm/Documents/workspace/arch-evident-spring/src/main/asciidoc/index.adoc

Installed asciidoctor: Processing file:/Users/odrotbohm/Documents/workspace/arch-evident-spring/src/main/asciidoc/index.adoc

2023-03-16 11:59:27.928 java[67574:15876898] CoreText note: Client requested name ".Helvetica Neue Desktop Interface - Regular", it will get "Helvetica Neue Desktop Interface - Regular".

2023-03-16 11:59:27.928 java[67574:15876898] CoreText note: Set a breakpoint on CTFontLogSystemFontNameRequest to debug.

2023-03-16 11:59:27.929 java[67574:15876898] CoreText note: Client requested name ".Helvetica Neue Desktop Interface - UltraLight P2", it will get "Helvetica Neue Desktop Interface - UltraLight P2".

2023-03-16 11:59:27.929 java[67574:15876898] CoreText note: Client requested name ".Helvetica Neue Desktop Interface - Thin", it will get "Helvetica Neue Desktop Interface - Thin".

2023-03-16 11:59:27.930 java[67574:15876898] CoreText note: Client requested name ".Helvetica Neue Desktop Interface - Medium P4", it will get "Helvetica Neue Desktop Interface - Medium P4".

2023-03-16 11:59:27.930 java[67574:15876898] CoreText note: Client requested name ".Helvetica Neue Desktop Interface - Bold", it will get "Helvetica Neue Desktop Interface - Bold".

2023-03-16 11:59:27.987 java[67574:15876898] CoreText note: Client requested name ".Helvetica Neue Desktop Interface - Regular", it will get "Helvetica Neue Desktop Interface - Regular".

2023-03-16 11:59:27.987 java[67574:15876898] CoreText note: Client requested name ".Helvetica Neue Desktop Interface - Bold", it will get "Helvetica Neue Desktop Interface - Bold".

Installed asciidoctor: Processing file:/Users/odrotbohm/Documents/workspace/arch-evident-spring/src/main/asciidoc/index.adoc

2023-03-16 12:03:45.255 java[68423:15883904] CoreText note: Client requested name ".Helvetica Neue Desktop Interface - Regular", it will get "Helvetica Neue Desktop Interface - Regular".

2023-03-16 12:03:45.255 java[68423:15883904] CoreText note: Set a breakpoint on CTFontLogSystemFontNameRequest to debug.

```

arch-evident-spring @ (6734c56 ...) $ 
arch-evident-spring @ (6734c56 ...) $ 
arch-evident-spring @ (6734c56 ...) $ 
arch-evident-spring @ (6734c56 ...) $ git co main
Previous HEAD position was 6734c56 Break module cycle using events.
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
arch-evident-spring @ main $

```

```

43 class EventPublicationRegistryTests {
44
45     private final OrderManagement orders;
46     private final EventPublicationRegistry registry;
47     private final FailingAsyncTransactionalEventListener listener;
48
49     @Test
50     void leavesPublicationIncompleteForFailingListener(Scenario scenario) throws Exception {
51
52         var order = new Order(new CustomerIdentifier(UUID.randomUUID()));
53
54         scenario.stimulate(() -> orders.complete(order))
55             .andwaitForStateChange(() -> listener.getEx())
56             .andVerify(_ -> {
57                 assertThat(registry.findIncompletePublications()).hasSize(1);
58             });
59     }
60
61     static class FailingAsyncTransactionalEventListener { }
62
63 }

```

Spring Modulith

- A convention to map modules to packages
- Simple set of access rules and API to verify
- Test support to bootstrap modules
- Event publication registry
- Documentation support
- Actuator and observability support