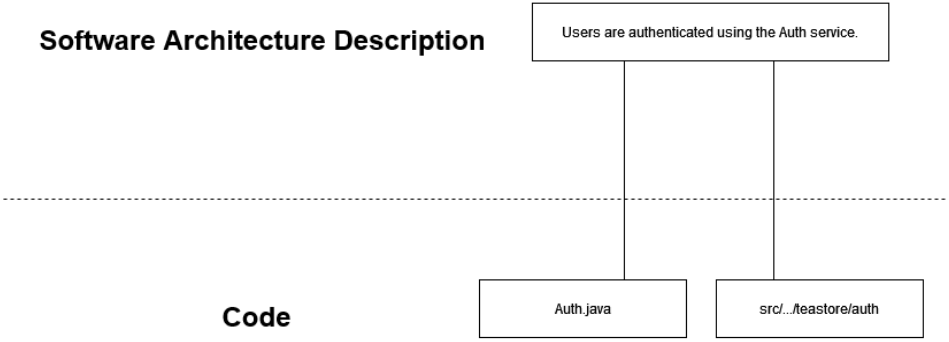
Karlsruher Institut für Technologie

Visualisierung von Trace Links

Praktikum Werkzeuge für Agile Modellierung

Philip Klemens / Dominik Fuchß | 18. März 2024

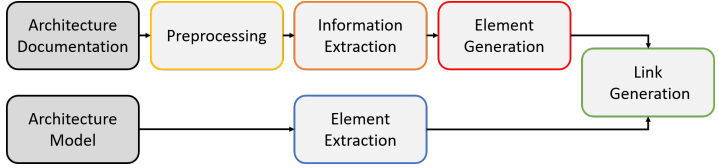


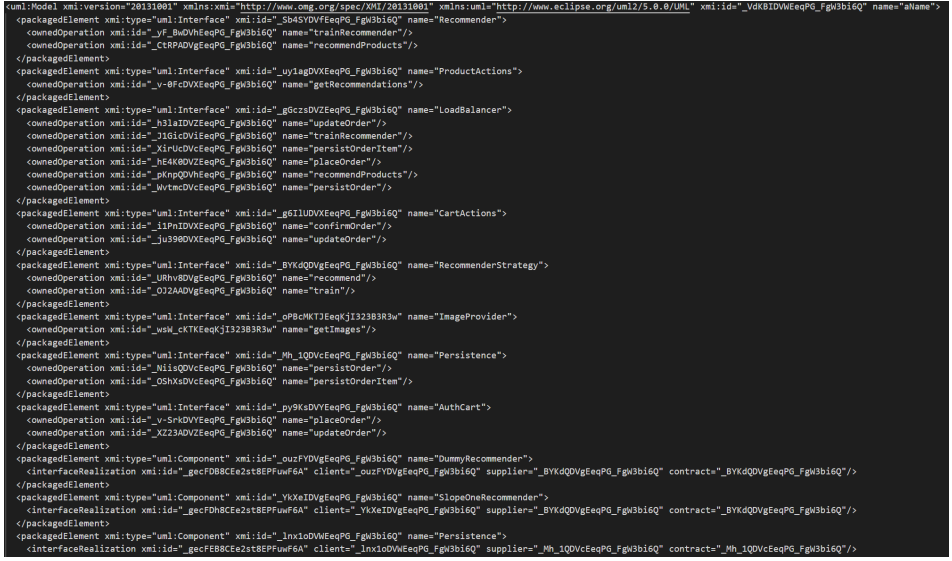
Trace-Links:

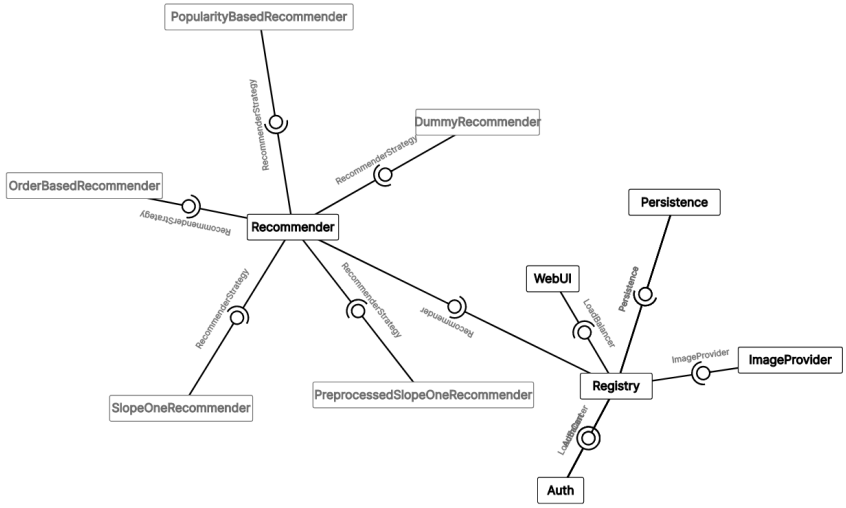
{ Users are authenticated using the Auth service , *Auth.class* }

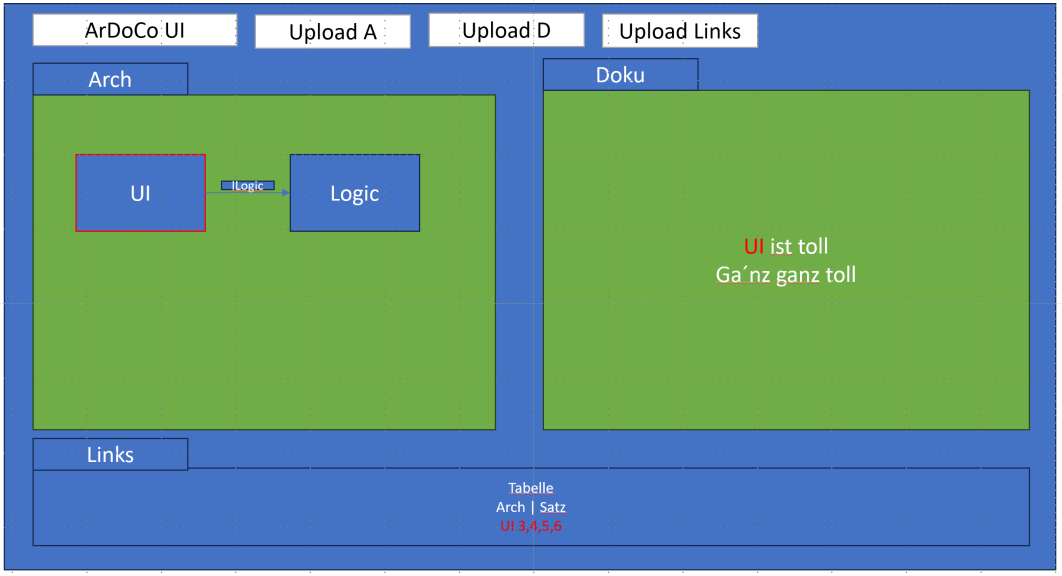
{ Users are authenticated using the Auth service , *src*/.../*teastore*/*auth* }

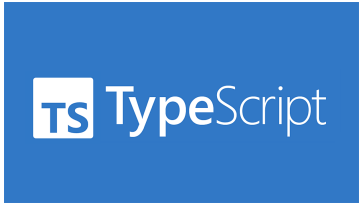
ArDoCo


Architecture Documentation
Preprocessing
Information Extraction
Element Generation
Link Generation
Architecture Model
Element Extraction


<uml:Model xmi:version="20131001" xmlns:xmi="http://www.omg.org/spec/XMI/20131001" xmlns:uml="http://www.eclipse.org/uml2/5.0.0/UML" xmi:id="_VdKBIDVWEeqPG_FgW3bi6Q" name="aName">
<packagedElement xmi:type="uml:Interface" xmi:id="_Sb4SYDVfEeqPG_FgW3bi6Q" name="Recommender">
<ownedOperation xmi:id="_yF_BwDVhEeqPG_FgW3bi6Q" name="trainRecommender"/>
<ownedOperation xmi:id="_CtRPADVgEeqPG_FgW3bi6Q" name="recommendProducts"/>
</packagedElement>
<packagedElement xmi:type="uml:Interface" xmi:id="_uy1agDVXEeqPG_FgW3bi6Q" name="ProductActions">
<ownedOperation xmi:id="_v-0FcDVXEeqPG_FgW3bi6Q" name="getRecommendations"/>
</packagedElement>
<packagedElement xmi:type="uml:Interface" xmi:id="_gGczsDVZEeqPG_FgW3bi6Q" name="LoadBalancer">
<ownedOperation xmi:id="_h3laIDVZEeqPG_FgW3bi6Q" name="updateOrder"/>
<ownedOperation xmi:id="_J1GicDViEeqPG_FgW3bi6Q" name="trainRecommender"/>
<ownedOperation xmi:id="_XirUcDVcEeqPG_FgW3bi6Q" name="persistOrderItem"/>
<ownedOperation xmi:id="_hE4K0DVZEeqPG_FgW3bi6Q" name="placeOrder"/>
<ownedOperation xmi:id="_pKnpQDVhEeqPG_FgW3bi6Q" name="recommendProducts"/>
<ownedOperation xmi:id="_WvtmcDVcEeqPG_FgW3bi6Q" name="persistOrder"/>
</packagedElement>
<packagedElement xmi:type="uml:Interface" xmi:id="_g6IlUDVXEeqPG_FgW3bi6Q" name="CartActions">
<ownedOperation xmi:id="_i1PnIDVXEeqPG_FgW3bi6Q" name="confirmOrder"/>
<ownedOperation xmi:id="_ju390DVXEeqPG_FgW3bi6Q" name="updateOrder"/>
</packagedElement>
<packagedElement xmi:type="uml:Interface" xmi:id="_BYKdQDVgEeqPG_FgW3bi6Q" name="RecommenderStrategy">
<ownedOperation xmi:id="_URhv8DVgEeqPG_FgW3bi6Q" name="recommend"/>
<ownedOperation xmi:id="_OJ2AADVgEeqPG_FgW3bi6Q" name="train"/>
</packagedElement>
<packagedElement xmi:type="uml:Interface" xmi:id="_oP8cMKTJEeqkjI323B3R3w" name="ImageProvider">
<ownedOperation xmi:id="_wsW_cKTKEeqkjI323B3R3w" name="getImages"/>
</packagedElement>
<packagedElement xmi:type="uml:Interface" xmi:id="_Mh_1QDVcEeqPG_FgW3bi6Q" name="Persistence">
<ownedOperation xmi:id="_NiisQDVcEeqPG_FgW3bi6Q" name="persistOrder"/>
<ownedOperation xmi:id="_OShXsDVcEeqPG_FgW3bi6Q" name="persistOrderItem"/>
</packagedElement>
<packagedElement xmi:type="uml:Interface" xmi:id="_py9KsDVYEeqPG_FgW3bi6Q" name="AuthCart">
<ownedOperation xmi:id="_v-SrkDVYEeqPG_FgW3bi6Q" name="placeOrder"/>
<ownedOperation xmi:id="_XZ23ADVZEeqPG_FgW3bi6Q" name="updateOrder"/>
</packagedElement>
<packagedElement xmi:type="uml:Component" xmi:id="_ouzFYDVgEeqPG_FgW3bi6Q" name="DummyRecommender">
<interfaceRealization xmi:id="_gecFDB8CEe2st8EPFuwF6A" client="_ouzFYDVgEeqPG_FgW3bi6Q" supplier="_BYKdQDVgEeqPG_FgW3bi6Q" contract="_BYKdQDVgEeqPG_FgW3bi6Q"/>
</packagedElement>
<packagedElement xmi:type="uml:Component" xmi:id="_YkXeIDVgEeqPG_FgW3bi6Q" name="SlopeOneRecommender">
<interfaceRealization xmi:id="_gecFDh8CEe2st8EPFuwF6A" client="_YkXeIDVgEeqPG_FgW3bi6Q" supplier="_BYKdQDVgEeqPG_FgW3bi6Q" contract="_BYKdQDVgEeqPG_FgW3bi6Q"/>
</packagedElement>
<packagedElement xmi:type="uml:Component" xmi:id="_lnx1oDVWEeqPG_FgW3bi6Q" name="Persistence">
<interfaceRealization xmi:id="_gecFEB8CEe2st8EPFuwF6A" client="_lnx1oDVWEeqPG_FgW3bi6Q" supplier="_Mh_1QDVcEeqPG_FgW3bi6Q" contract="_Mh_1QDVcEeqPG_FgW3bi6Q"/>


PopularityBasedRecommender
RecommenderStrategy
DummyRecommender
RecommenderStrategy
OrderBasedRecommender
RecommenderStrategy
Recommender
Persistence
WebUI
LoadBalancer
Persistence
RecommenderStrategy
RecommenderStrategy
Recommender
ImageProvider
ImageProvider
SlopeOneRecommender
PreprocessedSlopeOneRecommender
Registry
LoadBalancer
Auth


ArDoCo UI
Upload A
Upload D
Upload Links
Arch
UI
Logic
Logic
Doku
UI ist toll
Ga´nz ganz toll
Links
Tabelle
Arch | Satz
UI 3,4,5,6


TS
TypeScript

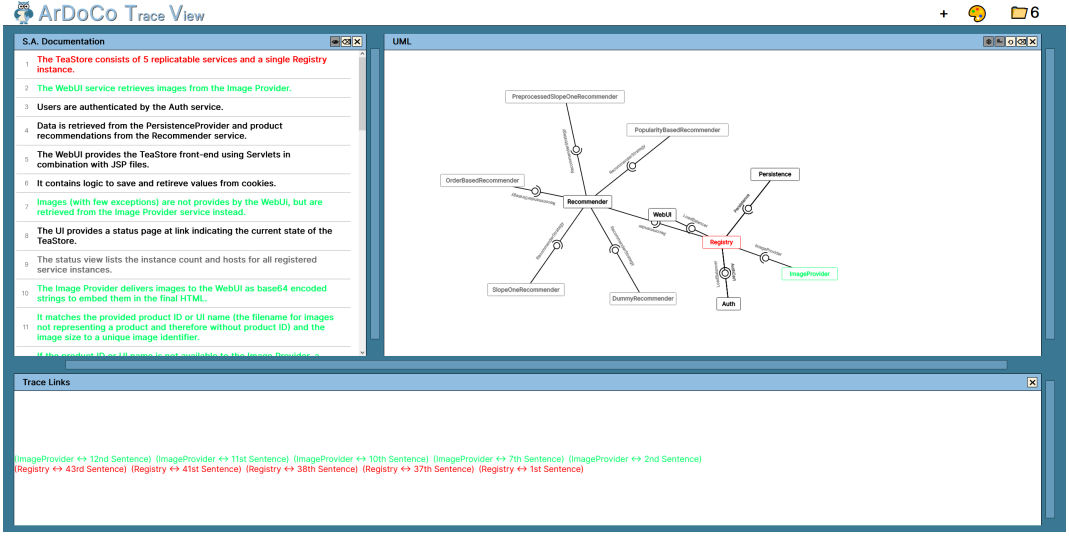

webpack


TSDoc


The JavaScript library for bespoke data visualization
Create custom dynamic visualizations with unparalleled flexibility
Get started
What is D3?
Examples

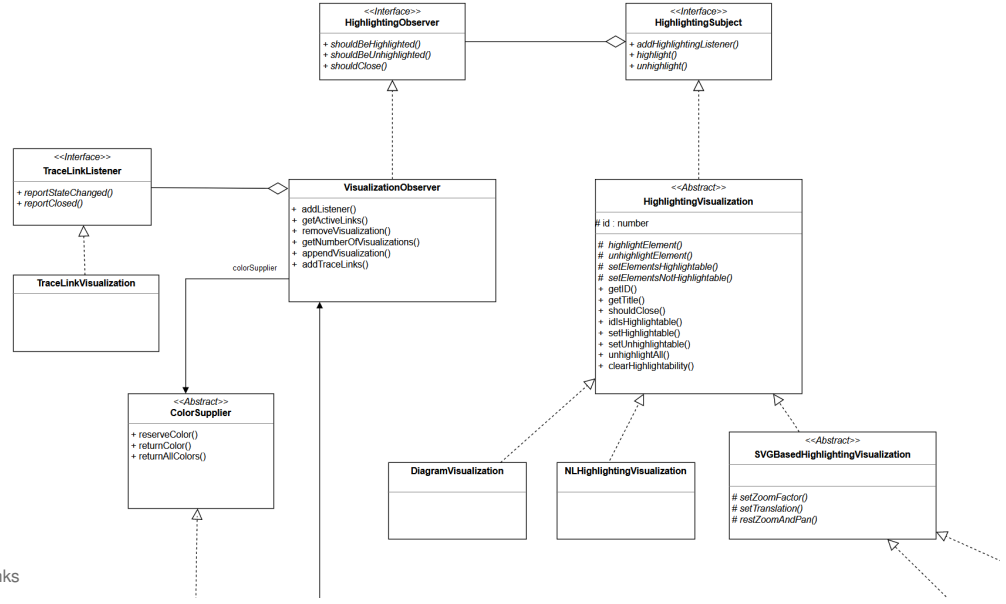
TraceView - Features

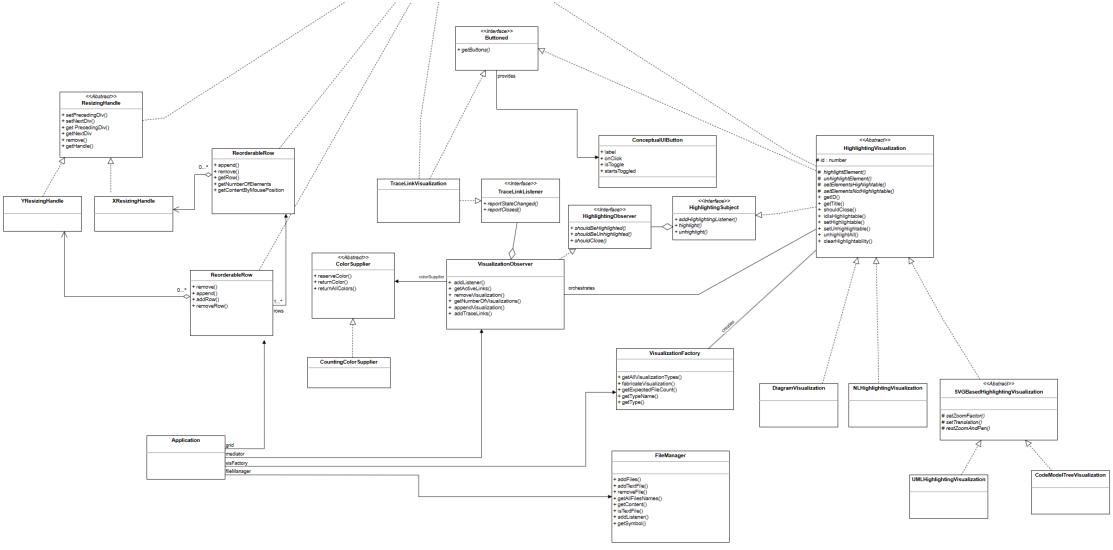
- Eigene Visualisierungen für
  - Dokumentation in natürlicher Sprache
  - Software Architecture Model (UML)
  - Code Model
  - Diagramme
- Implizite Darstellung von Traceability Links
- Erweiterbarkeit


ArDoCo Trace View
S.A. Documentation
1 The TeaStore consists of 5 replicatable services and a single Registry instance.
2 The WebUI service retrieves images from the Image Provider.
3 Users are authenticated by the Auth service.
4 Data is retrieved from the PersistenceProvider and product recommendations from the Recommender service.
5 The WebUI provides the TeaStore front-end using Servlets in combination with JSP files.
6 It contains logic to save and retireve values from cookies.
7 Images (with few exceptions) are not provides by the WebUI, but are retrieved from the Image Provider service instead.
8 The UI provides a status page at link indicating the current state of the TeaStore.
9 The status view lists the instance count and hosts for all registered service instances.
10 The Image Provider delivers images to the WebUI as base64 encoded strings to embed them in the final HTML.
11 It matches the provided product ID or UI name (the filename for images not representing a product and therefore without product ID) and the image size to a unique image identifier.
UML
PreprocessedSlopeOneRecommender
PopularityBasedRecommender
OrderBasedRecommender
Recommender
Persistence
WebUI
Registry
ImageProvider
SlopeOneRecommender
DummyRecommender
Auth
Trace Links
(ImageProvider ↔ 12nd Sentence) (ImageProvider ↔ 11st Sentence) (ImageProvider ↔ 10th Sentence) (ImageProvider ↔ 7th Sentence) (ImageProvider ↔ 2nd Sentence)
(Registry ↔ 43rd Sentence) (Registry ↔ 41st Sentence) (Registry ↔ 38th Sentence) (Registry ↔ 37th Sentence) (Registry ↔ 1st Sentence)

```
this.leftVisualization = new NLHighlightingVisualization(leftViewport, sentences, Array.from(highlightableSentences), sentenceColors);
this.rightVisualization = new UMLHighlightingVisualization(rightViewport, umlObjects, Array.from(highlightableUMLObjects), umlColors);
const lv = this.leftVisualization;
const rv = this.rightVisualization;
let leftToRightListener : HighlightingListener = new class implements HighlightingListener {
    wasHighlighted(id: string): void {
        traceLinks.filter((link) => link.source == id).forEach((link) => rv.setHighlighted(link.target,true,sentenceColors.get(id)!));
    }
    wasUnhighlighted(id: string): void {
        traceLinks.filter((link) => link.source == id).forEach((link) => rv.setHighlighted(link.target, false,sentenceColors.get(id)!));
    }
}
let rightToLeftListener : HighlightingListener = new class implements HighlightingListener {
    wasHighlighted(id: string): void {
        const leftIds : string[] = traceLinks.filter((link) => link.target == id).map((link) => link.source);
        leftIds.forEach((leftId) => lv.setHighlighted(leftId, true, umlColors.get(id)!));

    }
    wasUnhighlighted(id: string): void {
        const leftIds : string[] = traceLinks.filter((link) => link.target == id).map((link) => link.source);
        leftIds.forEach((leftId) => lv.setHighlighted(leftId, false, umlColors.get(id)!));
    }
}
this.leftVisualization.addHighlightingListener(leftToRightListener);
this.rightVisualization.addHighlightingListener(rightToLeftListener);
```


<<Interface>>
HighlightingObserver
+ shouldBeHighlighted()
+ shouldBeUnhighlighted()
+ shouldClose()
<<Interface>>
HighlightingSubject
+ addHighlightingListener()
+ highlight()
+ unhighlight()
<<Interface>>
TraceLinkListener
+ reportStateChanged()
+ reportClosed()
VisualizationObserver
+ addListener()
+ getActiveLinks()
+ removeVisualization()
+ getNumberOfVisualizations()
+ appendVisualization()
+ addTraceLinks()
<<Abstract>>
HighlightingVisualization
id : number
highlightElement()
unhighlightElement()
setElementsHighlightable()
setElementsNotHighlightable()
+ getID()
+ getTitle()
+ shouldClose()
+ idIsHighlightable()
+ setHighlightable()
+ setUnhighlightable()
+ unhighlightAll()
+ clearHighlightability()
TraceLinkVisualization
colorSupplier
<<Abstract>>
ColorSupplier
+ reserveColor()
+ returnColor()
+ returnAllColors()
DiagramVisualization
NLHighlightingVisualization
<<Abstract>>
SVGBasedHighlightingVisualization
setZoomFactor()
setTranslation()
restZoomAndPan()


<<Interface>>
Buttoned
+ getButtons()
provides
<<Abstract>>
ResizingHandle
+ setPrecedingDiv()
+ setNextDiv()
+ get PrecedingDiv()
+ getNextDiv
+ remove()
+ getHandle()
YResizingHandle
XResizingHandle
ReorderableRow
+ append()
+ remove()
+ getRow()
+ getNumberOfElements
+ getContentByVisualPosition
0..*
ReorderableRow
+ remove()
+ append()
+ addRow()
+ removeRow()
0..*
rows
TraceLinkVisualization
<<Interface>>
TraceLinkListener
+ reportStateChanged()
+ reportClosed()
ConceptualUIButton
+ label
+ onClick
+ isToggle
+ startsToggled
<<Interface>>
HighlightingObserver
+ shouldBeHighlighted()
+ shouldBeUnhighlighted()
+ shouldClose()
<<Interface>>
HighlightingSubject
+ addHighlightingListener()
+ highlight()
+ unhighlight()
<<Abstract>>
HighlightingVisualization
id : number
highlightElement()
unhighlightElement()
setElementsHighlightable()
setElementsNotHighlightable()
+ getID()
+ getTitle()
+ shouldClose()
+ setHighlightable()
+ setHighlightable()
+ setUnhighlightable()
+ unhighlightAll()
+ clearHighlightability()
<<Abstract>>
ColorSupplier
+ reserveColor()
+ returnColor()
+ returnAllColors()
colorSupplier
VisualizationObserver
+ addListener()
+ getAllActiveLinks()
+ removeVisualization()
+ getNumberOfVisualizations()
+ appendVisualization()
+ addTraceLinks()
orchestrates
CountingColorSupplier
VisualizationFactory
+ getAllVisualizationTypes()
+ fabricateVisualization()
+ getExpectedFileCount()
+ getTypeName()
+ getType()
creates
DiagramVisualization
NLHighlightingVisualization
<<Abstract>>
SVGBasedHighlightingVisualization
setZoomFactor()
setTranslation()
resZoomAndPan()
UMLHighlightingVisualization
CodeModelTreeVisualization
Application
grid
mediator
vizFactory
fileManager
FileManager
+ addFiles()
+ addTextFile()
+ removeFile()
+ getAllFileNames()
+ getContent()
+ isTextFile()
+ addListener()
+ getSymbols()

Erweiterungen

- Weitere Visualisierungstypen
- Weitere Datentypen
- UI
- Backend

Demo