# Designing a Web Application with Complex REST-generated GUIs Using JHipster and Angular.

# 1. Introduction.

This document describes the process of designing and developing a web application for the discovery, creation, removal, and modification of resources. By the term 'resource' we mean a basic entity of a complex Information System (which hereafter we will refer to as the IS), described in terms of metadata. Metadata for each resource, as well as hierarchies and relationships between resources, are returned in JSON format by Java APIs of the IS. A peculiarity of the web app we will describe lies in the fact that the resources' structures can vary dynamically over time, so it was necessary to build graphical interfaces based on descriptive metadata gathered in real time.

## 2. Working Environment.

The web app was developed using the JHipster platform (https://www.jhipster.tech/), which provides an integrated full-stack development environment web applications and microservices, based on Java and Node.js. For the backend side, JHipster leverages the Java reference frameworks Spring Boot and Spring Security (for the security and authentication layers), while for the frontend developers can choose among three of the most used frameworks based on Typescript, i.e., Angular, React and Vue. We decided to adopt Angular.

Since the IS APIs required Java 11, we could not use the latest version of JHipster, which required at least Java 17 and did not guarantee full retro-compatibility. The final configuration of the working environment resulted the following:

JAVA	openjdk version 11.0.20
JHIPSTER	v. 7.9.3
Angular CLI	14.2.1 (using rxjs 7.5.6 and typescript 4.8.2)
Node.js	14.20.1
Package Manager	npm 9.8.1
IDE	SpringToolSuite(STS) 4
	Mozilla Firefox Developer Tools

Angular natively supports Bootstrap for styling GUIs, and, in its most recent versions, also Material. We had decided to use Material, but the version of Angular shipped within Jhipster 7.9.3 still did not support it, so it was necessary to manually import and configure in Angular the Material widgets we wanted to use (e.g., Navigation Trees, Tabbed panes, Tables, etc.). The GUIs were tested and debugged on Mozilla Firefox and Google Chrome.

## 3. Archictecture.

The backend of a web app developed with JHipster is based on REST services. In our case, this meant that first of all it was necessary to develop a layer of Java code that exploited the APIs of the external environment (i.e., the Information System) to produce web services to be exposed by JHipster to precise endpoints. These endpoints are used by typescript services that expose data and functionalities to the

GUIs of the frontend, the exchange format between services being JSON. For both the backend and frontend, it is necessary to provide entities to hold the data to be exchanged. In the backend this is done through so-called DTOs (Data Transfer Objects). A DTO is essentially a java bean with no internal logic that is used to transfer data towards the frontend. Similarly, a Typescript interface is used in Angular to define the fields that will be returned by a service connected to a REST endpoint exposed by JHipster's backend. Figures 1 and 2 schematically show what we have just described.



#### Figure 1 A high-level description of the JHipster architecture.

Once the web app is deployed, the complete list of the backend web services can be found on a specific web page provided by JHipster. This is very useful for debugging purposes, as it is possible to query the services created by passing any required parameters, and to check the JSON response. Figure 3 shows what this page looks like in the case of our web app; we can distinguish between POST requests, which are related to CRUD methods, and GET requests, related to discovery operations performed on the Information System (i.e., operations to get all the resources, the JSON specification of each resource, etc.).

Figure 2: A high-level description of the architecture of our web application.



Figure 3: JHipster's web page to access to backend webservices.

information-system-resource	^
POST /api/is/updateresource	~
POST /api/is/deleteresource	$\checkmark$
POST /api/is/createresource	$\checkmark$
GET /api/is/resourcetypes	~
GET /api/is/resourcetypejson	~
GET /api/is/resourcetype	~
GET /api/is/resourcejson	$\checkmark$
GET /api/is/resourceinstances	~
GET /api/is/querytemplates	~
GET /api/is/facetspecifications	~
GET /api/is/facetfields	~
GET /api/is/allcontexts	~

## 4. Dynamic GUI Creation.

Angular allows GUIs to be created according to a modular mechanism, i.e. it is possible to compose a GUI using independent, self-consistent base elements, called 'components'. An Angular component is a graphical interface defined by the developer, and can have its own parameters, can be connected to

Angular services, has its own style, etc. A component can interact via parameters with other components, parents or children, and above all it can be used several times in the web app with different configurations and data. Each component is uniquely identified by a tag, by which it must be referred to within the html code (https://angular.io/guide/component-overview).

Figure 5 shows a web page of our application, in which some components are highlighted.

🖹 File 🛛 🖍 Edit 🛛 🖉 View 🖉 🖗 Holp	Contexts * /gcube/devsec/devVRE *	Context Name devVRE	Context UUID ded93b00-934c-11	ee-869e-bfa3de466a	2b 🔽 4	0			
Resource Types	VirtualService 0	0						<tab< th=""><th>le-so</th></tab<>	le-so
✓ Resource <sup>A</sup>									
✓ GCubeResource <sup>A</sup>	Search by Name	3 Reload + Add new							
✓ Actor A						Last			
LegalBody		Name		Group versi	on Id	Modified	Actic	ens	
Person	Resource Catalogue			Data Discovery	bd2a1151- da7c-424c-	2024-06-25			
✓ ConfigurationTemplate	Resource Calangue			and Access	8ce1- e510e8f928e9	12:59:43	• /	- C.	
Configuration					Items per page: 10	▼ 1-1 of 1	14 4	> >	
✓ Dataset								_	11.
ConcreteDataset	bd2a1f51-da7c-424c-8	×							
Schema	Raw JSON -> bd2a1f51-da	17c-424c-8ce1-e510e8f928e9							
✓ Service <sup>A</sup>								^	
✓ EService	"type": "VirtualService", "supertypes": [								
RunningPlugin	"Service", "GCubeResource",								
HostingNode	], 	*8ce1=e510e8f028e0".							
VirtualMachine	"consistsOf": [ {	, ,							
VirtualService	"type": "IsIdentifier	Юу",							

Figure 4: Angular components used to compose a GUI.

As an example, consider the creation of the resource-tree component for resource navigation. In the backend layer, we have provided a method to get the hyerarchy of the resources from the IS, and exposed it through a web service at the endpoint '/api/is/resourceinstances'. On the Angular frontend side, we created the following pieces of code:

- resource-tree.component.ts: the Typescript logic for the component
- resource-tree.component.html: the component 's HTML structure
- resource-tree.component.scss: for the component 's styling
- resource-tree.service.ts: the backend communication service injected into the component
- i-resource.ts: the Typescript interface which allows to translate the JSON objects retrieved by the service into proper Typescript objects.

In resource-tree.component.ts an Angular tree is instantiated. The component also contains the code to manipulate the resource implementations, so as to create objects suitable to be hierarchically displayed in the tree. Moreover, it defines the interactive behavior of the interface (e.g., how to respond to 'onClick' and 'onHover' events).

# 5. Forms for Creating Resources.

The structure of each IS resource is described in terms of elementary sub-resources, called "facets". The IS APIs provide methods to return the JSON description of a resource in terms of facets and, for each facet, the associated data structure. Combining this methods allowed us to discover in real time the composition of a resource and consequently set up its creation GUI. In Angular, it is possible to build forms dynamically through the "reactive forms" feature (https://angular.dev/guide/forms/reactive-forms). As the official Angular states, "reactive forms provide a model-driven approach to handling form inputs whose values change over time". We defined a number of form templates in JSON format, to account for data-entries required for the different facets; these templates could be dynamically composed in accordance with each resource specification. Using the proper methods of the IS API, we then created a method the backend REST, to retrieve the form specification for any kind of resource. Figure 5 shows part of the JSON response of the REST for a specific resource type. The JSON description of the data entry fields required for the creation of a facet is highlighted.



Figure 5: JSON specification of the fields required for the creation of a MemoryFacet, as part of the form for creating a HostingNode resource instance.

Since a resource may contain a variable number of facets of the same type, it was necessary to give users the possibility to dynamically add and remove the form potions related to facets. The overall GUI for the creation of a resource, therefore, is composed by the creation forms of the various facets, which can be added or removed depending on what is specified in the IS. The appendix to this document shows various screenshots related to the creation of a resource, namely an 'Eservice'.

6. Main Functionalities.

In this section, the main functionalities of the application are described. Screenshots of the application which display them are shown in the Appendix.

### 6.1 Switching Context.

The IS accounts for different 'contexts', i.e. different working environments, each containing different resources. The main GUI of the application provides a drop-down menu to switch between the available contexts and consequently retrieve the list of the available resource types and instances.

## 6.2 Navigating Through Resource Types.

Resource types are organized hierarchically, therefore they are shown by means of a tree view. Some resource types are 'abstract', i.e., they do not allow for instances; these resource types are highlighted using italics and labelled with the letter 'A'; no event is triggered when clicking on an abstract resource type.

### 6.3 Browsing and Searching Resources.

'onClick' events are captured on each non-abstract node of the resource types' tree and trigger the research of all the correspondent resource instances, which are shown in table form. Resource instances of a given type are shown in a table which displays the resources' main features. The number of rows in the table depends on the pagination parameter set by the user. Alphabetical and numerical (ascending/descending) ordering for each column of the table is supported; searching on the basis of the resource name is also supported.

### 6.4 Visualizing Descriptive Information and Metadata.

For each resource type, it is possible to access a high-level textual description and its complete JSON specification. The JSON metadata file of each resource implementation is also available for display.

### 6.5 CRUD Operations.

In each row of the table containing the resources two buttons are available, one for editing the resource and one for removing it. A button placed before the table allows to create a new resource instance for the current type.

## Appendix A – Screenshots of the Main Functionalities.

This appendix gathers screenshots of the most important GUIs of the web application. All of them have been developed in Angular using Material design components (<u>https://material.angular.io/</u>). This was challenging, since the JHipster version we used did not support Material for Angular natively, so we had to manually import and set up each Material component we wanted to use (tree view, table, tabbed view, etc.).

#### A.1 Browsing and Discovery

An expandable tree is used to show the relations between resource types. The component leverages the Material look and feel.

Resourc	e Types
✓ Reso	burce <sup>A</sup>
~ (	GCubeResource <sup>A</sup>
~	Actor <sup>A</sup>
	LegalBody
	Person
~	ConfigurationTemplate
	Configuration
~	Dataset
	ConcreteDataset
	Schema
~	Service <sup>A</sup>
	✓ EService
	RunningPlugin
	HostingNode
	VirtualMachine
	VirtualService
	Site
~	Software
	Plugin

# A.2 Providing Descriptions and Metadata.

For each resource type it is possible to view a general descriptive information and its JSON specification. The following screenshots have been taken for the 'Hosting Node' type.

			_	TE Home	- Adm	inistration *	. A ma	ateres	a, par
🔄 File 🧨 Edit 🕖 View 🕥 Help	Contexts * Context Name /gcube - 10 gcube	Context UUID 9a7752f6-ca01		^					
esource Types	HostingNode 0 O	Hosting	Noc	de	0		0		
<ul> <li>✓ Resource <sup>A</sup></li> <li>✓ GCubeResource <sup>A</sup></li> </ul>	Search by Name SReload + Add new		/				Ŭ		
✓ Actor <sup>A</sup>	Name	м 🥢	Status	Last Modified	Available Memory	HD Space	3	Actions	
LegalBody									
Pera     The HostingNode represent a con conceived to enable any services Config in terms of facets reflects the one enabling the HostingNode is not in	ntainer capable of managing the lifecycle of an electronic service, i.e., bein respecting the container rules to be operative. The container does not typ presented for (@link \irtualMachine). In particular, facets representing m represented, referred systems can provide \irtual machines as resource	g capable to host and operate an (@link EService). Examples are i ically provide any functionality rather than allowing the hosted see emory, CPU and networking interface are used to describe the Ho or containers as resources. In some cases. the description of a co	docker, tome rvice to oper ostingNode w	cat. A containe rate. The Hosti when the (@link udes (virtual) h	r is a servic ngNode ch c VirtualMac ardware inf	ce which is aracterisation chine} formation. It is		1	and the
Control     Contro     Control     Control     Control     Control     Control     Co	tainer capable of managing the lifecycle of an electronic service, i.e., bein respecting the container rules to be operative. The container does not by spreamed to (Gilkin Vitualiaching), in previously, foots representing represented. Federated systems can provide vitual machines as resource phode is not subclass of (Gilkin VitualiAccine).	g capable to host and operate an (@ink EService). Examples are scally provide any functionality rather than allowing the hosted service, CPU and networking interface are used to describe the ise or containers as resources. In some cases, the description of a co	docker, tom rvice to oper astingNode w ontainer inclu	cat. A containe rate. The Hosti when the (@link udes (virtual) h	r is a servic ngNode ch c VirtualMac ardware inf	ce which is laracterisation chine} formation. It is MB	, , , ,	1 1 1	-
Legatedy     Per     The HostingNade represent a con     Contextual to enable any services     Contextual to enable     Contextual     Contextual     Contextual	trainer capable of managing the lifecycle of an electronic service, i.e., bein respecting the container rules to be operation. The container does not by prosented for (life) Windhafching, in provide virtual machines as resource ghode is not a subclass of (life) VirtualNachine). Inservice storagehot. 1 doub devide/one.org	g capable to host and operate an (@rink EService). Examples are scalar provide any functionality rether than allowing the hosted set encourses of the set of the set or containers as resources. In some cases, the description of a co 20ex181-te 7beil-tope 4963-23001530882	docker, tome rvice to oper ostingNode w ontainer inclu down	cat. A containe rate. The Hosti when the (@link udes (virtual) h 14.07.22 2024.03-22 10.41.32	r is a servic ngNode ch c VirtualMac ardware inf 0 MB	ce which is aracterisatior chine} formation, it is MB 3944 MB	, , , , , , ,	1 1 1 1	1
Contrabutest     Contrabutest     Contrabutest     Contrabutest	tative capable of managing the lifecycle of an electronic service, i.e., bein respecting the container rules to be operative. The container does not by presented to ((life): Withaulichnich), presenting an errorsented. Federated systems can provide virtual machines as resource globels in of a subclass of ((life): Virtual/Acchine).	g capable to host and operate an (@rink EService), Examples are scally provide any functionality retrief than allowing the hosted set ency, CPU and networking interface are used a describe the to or containers as resources, in some cases, the description of a co 20018146 / Roll 4266 9155 253001538802 20018146 / Roll 4266 9155 253001538802	docker, tomo rvice to oper sstingNode w ontainer inclu down	cat. A containe rate. The Hosti when the (@link udes (virtual) h 14.07.22 2024-03-22 10.41.32 2024-01-18 15.29:00	r is a servic ngNode ch c VirtualMac ardware inf 0 MB 0 MB	ce which is laracterisation chine) formation. It is MB 3944 MB	· · · ·	1111	1
Legalitary Para Convertigent and the main any services Convertigent and the main any services concrete the service services and the services concrete the services and the services convertigent and the hydrogen that the service convertigent and the services Services 4 Convertigent and the services Convertigent and the services Co	trainer capable of managing the lifecycle of an electronic service, i.e., bein respecting the container rules to be operative. The container obes not by service the (life) kindhalching in generational, hosts respectivity in represented to (life) kindhalching in generational, hosts resource pilode a not a subclass of (life) kindhalching). Best strangenter 1.doud dev.discores.org gest 1.doud dev.discores.org gest 1.doud dev.discores.org	g capable to host and operate an (@rink EService). Examples are scally provide any functionality system than allowing the hosted se monty. (CPU and networking instring are are used to describe the to or containers as resources, in some cases, the description of a co 20x1814er 7bb8-k606-953-203001030882 described-ext2-497e-6458-203001030882 735a4bca.door 1414-9827-63778584058	docker, tomi rvice to oper stingNode w ontainer inclu down down down	Cat. A containe rate. The Hosti when the (@link udes (virtual) h 14:07:22 2024-03-22 10:41:32 2024-01-18 15:29:00 2024-01-19 14:19:22	r is a servic ngNode ch VirtualMac ardware inf 0 MB 0 MB	ce which is aracterisation chine} formation. It is 3944 MB 3944 MB 3944 MB		1 1 1 1 1 1	11111
Legatility         Perror         The Hostingfolds represent a concretived to enable any services in terms of facts reflects the concretived to enable any services in terms of facts reflects the concretive data of the terms of the of terms of the terms of terms o	tainer capable of managing the lifecycle of an electronic service, i.e., bein respecting the container rules to be operative. The container does not by presented for (life): Virualization (in presented as a resource pixed is not a subclassical (life): Virualization): bited is not a subclassical (life): Virualization); defined is not a subclassical (life): Virualization); gest-1.dbud-developmen.org gest-1.dbud-developmen.org gest-1.dbud-developmen.org	g capable to host and operate an (@rink EService). Examples are scally provide any functionality strate than allowing the hosted se service of the service of the service of the service of the service or containers as resources. In some cases, the description of a co doubter root service of the service of the service of the service doubter root service of the service of the service of the service doubter of the service of the service of the service of the service doubter of the service of the serv	docker, tom rvice to oper stringNode w ontainer inclu down down down down	cat. A containe rate. The Hosti when the (@Vini 44.07.22 2004-00-22 10:41:32 2024-01-18 15:28:00 2024-01-19 15:28:58	r is a servic ngNode ch c VirtualMac ardware inf 0 MB 0 MB 0 MB	ze which is taracterisation chine) formation. It is 3944 MB 3944 MB 3944 MB	· · · · · · · · · · · · · · · · · · ·	1111111	1
Legalitady Perr Table hashing lack aspected as con- analying the Hosting/Index aspected as con- analying the Hosting/Index in or a marking the Hosting the Hosting/Index in or a marking the Hosting/Index in or a	tainer capable of managing the lifecycle of an electronic service, i.e., bein respecting the container rules to be operative. The container does not by servershot for (life)-Winabaching), inspressional, foots representing pixed is not a subclass of (life)-Vinabhaching), and provide virtual machines as resource gixed is not a subclass of (life)-Vinabhaching), and the operative gest-1.cloud-devideomes.org	g capable to host and operate an (@rink EService), Examples are scally provide any functionality rather than allowing the hosted se encoder the second second second second second second second second second or containers as resources, in some cases, the description of a co 20x1814e-7x88-4265-9153-22X007530882 98x1068e-6072-4076-6853-40260470421e 732446a-004-1418-9073-607364004584 au6x304-7x72-4x05-8033-4026040584	docker, tome rvice to open stringNode w ontainer inclu down down down down	cat. A containe rate. The Hosts when the (@Inia des (withual) h 14.07.22 2024-03-22 10.41.32 2024-01-18 15.28:00 2024-01-18 15.28:00 2024-05-10 15.38:56 2024-05-03 15.38:56 2024-05-10 09.44:54	r is a servic ngNode ch c VirtualMac ardware inf 0 MB 0 MB 0 MB 0 MB	ee which is aracterisation chine) formation. It is 3944 MB 3944 MB 3944 MB 3944 MB 3944 MB 2917 MB		1 1 1 1 1 1 1	1



For each resource implementation, it is also possible to view its JSON metadata. All the resources of a certain type are shown in a table. Each row of the table contains the resource's main properties and three buttons, namely, 'view', edit, and 'delete'. When the 'view' button is clicked, a new tab is shown in a tabbed pane placed below the table, containing the JSON description retrieved from the IS.

whnmanager		7c5fc06c-ef7b- 4375-b383- ab027603bb69	down	2024-05-06 13:41:42	http://grsf-publish	er-2.cloud	•		i
					ver page: 10 👻	1 – 10 of 117	< <	>	>1
1b7a08fb-9f15-4a4f-b X 6295598e-	3187-4c8d-a ×								
Raw JSON -> 6295598e-3f87-4c8d-ab26-536d46	8d87b2								
<pre>{     "type": "EService",     "supertypes": [         "supertypes": [             "Service",             "GoUbeResource",             "Resource" ],             "consistsof": [             {                   type": "IsIdentifiedBy",                   "supertypes": [                         "Consistsof"             ],                  "id": "fb01ced6-281c-4c78-9840-86300c4c0b3             "propagationConstraint",                  "add": "propagate",             "add": "propragate",             "add": "propagate",             "add":</pre>	ia",								
"delete": "cascade", "remove": "cascade", "supertypes": [ "Property" ] }.									Ŧ

## A.3 CRUD Operations

CRUD operations on the resources are allowed by means of the buttons highligted in the picture below. The 'Add new' button opens a form to create a new instance for a given resource type, while the 'edit' and 'remove' buttons placed on the row of each resource allow its modification or removal.

HostingNode 🟮 💿	Create the typ	ite a new resource of type HostingNode						
Search by Name Search by Name	Add new					Update or de	elete this	
Name	Id	Status	Last Modified	Available Memory	HD Space	resource Actions	1	
gcat-2.cloud-dev.d4science.org	1fe9775e- 8d36-41f0- 97f9- dbe257ec6cbf	down	2024-01-18 15:19:07	0 MB	3944 MB	0 🖍	Î	
accounting-service-1.cloud-dev.d4science.org	21657bd0- f14f-4f23- 99dc-	certified	2024-08-04 06:39:57	9740 MB	3944 MB	• /	ĩ	

## A.4 Creation Forms

The following screenshots show parts of the form for the creation of a new 'EService' instance. The form is composed of a number of facet-related forms, one for each facet that makes up the resource. Since the number of facets of the same type may be variable, facet-related forms can be added or removed

dynamically by the user. The screenshots here shown represent the portions of the GUI for the creation of a new EService, related to two of its facets, namely, an AccessPointFacet and an EventFacet.

<i>,</i> ,					
Resource <sup>A</sup>	FacetComposer: New EService (/gcube/devsec)				
✓ GCubeResource <sup>A</sup>	EService				
✓ Actor <sup>A</sup> LegalBody	AccessPointFacet 1 Boour max	d Endpoint		Actions	
Person	AccessPointRoet captures information on an 'access point' of a resource. Le. any web-based endpoint to programmatically	3-04 http://accounting-service-1.cl	o	1	1
Configuration	relation ConsistsOf +				
✓ Dataset	Description	1-08 https://accounting-service.dev	O	1	I
Schema	EntryName	7-23 http://access- 6 d.d4science.org	o	1	
✓ Service <sup>A</sup>	Authorization				
RunningPlugin	Choose: - mariateresa paratore	3-04 2 http://accounting-service-2.cl	0	1	1
HostingNode VirtualMachine	Land V	5-28 http://data1.dev.int.d4science	o	1	
VistualSarvica	Submit Cancel				
	967/0565-				
purce Types	Econico O				
ource Types Resource *	FacetComposer: New EService (/gcube/devsec)				
ource Types Resource <sup>A</sup> GoubeResource <sup>A</sup>	FacetComposer: New EService (/gcube/devsec)				
OUITCE Types Resource <sup>A</sup> COubeResource <sup>A</sup> Actor <sup>A</sup> LegalBody	FacetComposer: New EService (/gcube/devsec)	d Endpoint		Actions	
Ource Types Resource <sup>A</sup> CoubeResource <sup>A</sup> Actor <sup>A</sup> LegalBody Person ConfigurationTemplate	FacetComposer: New EService (/gcube/devsec)  Authorization  Choose:  mariateresia paratore  Protocol	d Endpoint	0	Actions	
OUICE Types Resource <sup>A</sup> CoubeResource <sup>A</sup> Actor <sup>A</sup> LegalBody Person Configuration Femplate Configuration	FacetComposer: New EService (/gcube/devsec)  Authorization  Choose maniateresa paratore  Protocol  Endpoint *  This field is required	d Endpoint d Endpoint d http://accounting-service-1.cl	•	Actions	
Ource Types Resource <sup>A</sup> GoubeResource <sup>A</sup> Actor <sup>A</sup> LegalBody Person Configuration Template Configuration Dataset ConcreteDataset	FacetComposer: New EService (/gcube/devsec)  Authorization  Choose:  mariateresa paratore  Protocol  Endpoint *  This field is required  Custom Properties  boolean  ame *  Remove custom property	d Endpoint 1-04 http://accounting-service-1.cl 1-06 https://accounting-service dev	0	Actions	
Ource Types Resource <sup>A</sup> GOUBRESource <sup>A</sup> GOUBRESource <sup>A</sup> GOUBRESource <sup>A</sup> LegatBody Person  Configuration Template Configuration  Dataset ConcreteDataset Schema  Schema	FacetComposer: New EService (/gcube/devsec)	d Endpoint 1-04 http://accounting-service-1.cl 1-05 http://accounting-service.dev 1-23 http://access- d.d6aclence.org	•	Actions	
Ource Types Resource <sup>A</sup> CourbeResource <sup>A</sup> CourbeResource <sup>A</sup> CourbeResource <sup>A</sup> CourbeResource <sup>A</sup> LegalBody Person Configuration Template Configuration Config	FacetComposer: New EService (/gcube/devsec)  Authorization  Choose:  mariateresa paratore  Protocol  Custom Properties  boolean  ame *  Remove custom property  date  datetime	d Endpoint -04 http://accounting-service-1.cl -05 https://accounting-service-2.cl -04 http://accounting-service-2.cl	0	Actions	
Curce Types Resource <sup>A</sup> GOubeResource <sup>A</sup> GOubeResource <sup>A</sup> LegatBody Person Configuration Template Configuration Template ConcreteDataset Schema Schema Schema Schema Schema Schema HostingNagin HostingNagin	FacetComposer: New EService (/gcube/devsec)	d Endpoint Http://accounting-service-1.cl http://accounting-service-1.cl http://access- d.dsfailence.org	0	Actions	
Ource Types Resource <sup>A</sup> GCubeResource <sup>A</sup> Actor <sup>A</sup> LegalBody Person Configuration Template Configuration Configuration Dataset ConcreteDataset Schema Schema Schema ConcreteDataset ConcreteDataset Schema HostingNade VitualMachine	Econolica C FacetComposer: New EService (/gcube/devsec) Authorization Choose mariateresa paratore Protocol Indpoint * This field is required Indpoint * This field is required Indpoint * This field is required and * Remove custom property date	d     Endpoint       d     Endpoint       4     Intro Jilaccounting service-1 ct       108     http://accounting-service-1 ct       109     http://accounting-service-2 ct       109     http://accounting-service-2 ct       109     http://accounting-service-2 ct       109     http://accounting-service-2 ct	0 0 0 0	Actions	
OUICE Types Resource <sup>A</sup> GOubeResource <sup>A</sup> GOUbeResource <sup>A</sup> LegalBody Person Configuration Template Configuration Configuratio	FacetComposer: New EService (/gcube/devsec)	d Endpoint d Endpoint d http://accounting-service-1.cl. d dscience org	0 0 0 0	Actions	

Resource Types  C Resource <sup>4</sup> C GludeResource <sup>4</sup> C Actor <sup>4</sup> Legislitody	EService  FacetComposer: New Se EventFacet	EService (/gcube/dev	rsec)			Т					
Resource 4     Ocubantesource 4     Actor 4     LegetBody	FacetComposer: New	EService (/gcube/dev	occurr, max								
<ul> <li>GCubeResource <sup>4</sup></li> <li>Actor <sup>4</sup></li> <li>LegalBody</li> </ul>	EventFacet	occurr. min 1	occurr, max								
✓ Actor A LegalBody	EventFacet	occurr. min 1	occurr, max								
LegalBody			many			tatus	Last	Endpoint		Actions	
							Modified				
Person	e EventFacet captures inf event are the start time	ormation on a certain event/ of a virtual machine or the a	/happening characterising the life cycle of th activation time of an electronic service.	e resource. Examples of	an 14	sady	2024-08-04 06:50.25	http://accounting-service-1.cl	o	1	
Computation remplate	ConsistsOf	-									
Configuration	e Event					own	2024-01-08 11:56:35	https://accounting-service.dev	o	1	Î
ConverteDataset											
Schema	d type	▼ hame *	Remove custom property			own	2024-07-23	http://access- d.d4science.org	o	1	Î
✓ Service <sup>A</sup>	· · · · · · · · · · · · · · · · · · ·	•									
✓ EService	Add custom property					hadiv.	2024-08-04	http://accounting.service-2.cl	0		
RunningPlugin	Ū					,	06:48:12		Ŭ	1	
HostingNode											
VirtualMachine	d + Add new					ady	2024-08-28 12:46:34	http://data1.dev.int.d4science	Θ	1	Û
VirtualService						•					
Site	L.			Submit	Cancel	заду	2024-06-28	http://data1.dev.int.d4science	o	1	î
✓ Software						_					
Plugin					1e2da22c- 6330-4776-		2024-05-06	hite lines to should be affecte	-		

## Appendix B – Code Samples for the Creation of the "Resource Tree" Component.

This appendix shows the code used to create the resource-tree navigation component.

### B.1 rsc-tree.component.ts

This is the Typescript code responsible for creating the Angular component <jhi-rsc-tree> in the JHipster environment (the 'jhi-' prefix is required by JHipster). Besides the selector name, the html template and css style filenames are declared. The 'providers' keyword identifies the list of services which will be used to fill the component with data.

```
/* eslint-disable no-console */
import { Component , OnInit, Output, EventEmitter } from '@angular/core';
import { MatTreeNestedDataSource } from '@angular/material/tree';
import { NestedTreeControl } from '@angular/cdk/tree';
import { RestypesService } from 'app/services/restypes.service';
import { IResource } from 'app/services/i-resource';
@Component({
 selector: 'jhi-rsc-tree',
 templateUrl: './rsc-tree.component.html',
 styleUrls: ['./rsc-tree.component.scss'],
 providers: [RestypesService],
})
export class RscTreeComponent implements OnInit{
 nestedTreeControl = new NestedTreeControl<IResource>(node => node.children);
 nestedDataSource = new MatTreeNestedDataSource<IResource>();
 statusClass = 'not-active';
@Output() public resourceTypeEm = new EventEmitter<string>();
 constructor(private rtService:RestypesService) {
 }
 ngOnInit(): void {
      this.rtService.fetchAll().subscribe(res => {
```

```
this.nestedDataSource.data = res;
      this.nestedTreeControl.dataNodes = this.nestedDataSource.data;
      this.nestedTreeControl.expandAll();
       });
        }
hasNestedChild( : number, node: IResource): boolean {
   if (node.children == null) {
     return false;
    } else {
     return node.children.length > 0;
    }
  }
  //TODO: InformationSystemResourceClient shoud pass a code, not a name!
    onClickNodeTree(node:IResource):void{
        //this.setActiveClass();
        this.resourceTypeEm.emit(node.name);
    }
 setActiveClass():void{
   this.statusClass = 'active';
  }
}
```

#### B.2 rsc-tree.module.ts

In short, a 'module' is a file which declares the dependencies of our component.

```
import { NgModule } from '@angular/core';
import { SharedModule } from 'app/shared/shared.module';
import { MatIconModule } from '@angular/material/icon';
import { MatTreeModule } from '@angular/material/tree';
import { RscTreeComponent } from './rsc-tree.component';
@NgModule({
    imports: [SharedModule, MatIconModule, MatTreeModule],
    declarations: [
        RscTreeComponent
    ],
    entryComponents: [RscTreeComponent],
    exports: [RscTreeComponent],
    exports: [RscTreeComponent]
})
export class RscTreeModule { }
```

### B.3 i-resource.ts

This is the simple interface we built to contain data retrieved from the service.

```
export interface IResource {
  name: string;
  id: string;
  children?: IResource[];
}
```

#### B.4 restypes.service.ts

The Typescript service that connects to the proper JHipster REST endpoint and retrieves data to fill our component. Data are parsed form the JSON format into objects according with the defined interface.

```
import { Injectable } from '@angular/core';
import { HttpClient, HttpHeaders } from '@angular/common/http';
import { Observable } from 'rxjs';
import { IResource} from './i-resource';
import { ApplicationConfigService } from 'app/core/config/application-config.service';
@Injectable({
 providedIn: 'root',
})
export class RestypesService {
  httpOptions = {
   headers: new HttpHeaders({ 'Content-Type': 'application/json' }),
  };
 constructor(private http: HttpClient, private applicationConfigService:
ApplicationConfigService) { }
 fetchAll(): Observable<IResource[]> {
  const resourceUrl = this.applicationConfigService.getEndpointFor('api/is/resourcetypes');
   return this.http.get<IResource[]>(resourceUrl);
  }
}
```

#### B.5 rsc-tree.component.scss

This file defines the style to be applied to the component.

```
.example-tree-invisible {
 display: none;
}
.example-tree ul,
.example-tree li {
margin-top: 0;
 margin-bottom: 0;
 list-style-type: none;
}
.example-tree .mat-tree-node {
   width: 300px;
.example-tree .mat-tree-node :hover {
   background-color: bisque;
}
* This padding sets alignment of the nested nodes.
* /
```

```
.example-tree .mat-nested-tree-node div[role='group'] {
 padding-left: 25px;
}
/*
* Padding for leaf nodes.
 * Leaf nodes need to have padding so as to align with other non-leaf nodes
* under the same parent.
 * /
.example-tree div[role='group'] > .mat-tree-node {
 padding-left: 25px;
}
/*
.active{
 background-color: bisque;
* /
/*
.not-active{
    background-color: transparent;
*/
```

B.6 rsc-tree.component.html

This is the code which defines the html template for our component. It leverages Angular Material, in particular the Angular Material Tree (mat-tree) component.

```
<mat-tree [dataSource]="nestedDataSource" [treeControl]="nestedTreeControl" class="example-
tree">
   <mat-tree-node *matTreeNodeDef="let node" matTreeNodeToggle>
        <button mat-button (click)="onClickNodeTree(node)">
            {{ node.name }}
        </button>
    </mat-tree-node>
    <mat-nested-tree-node *matTreeNodeDef="let node; when: hasNestedChild">
      <div class="mat-tree-node">
        <button mat-icon-button matTreeNodeToggle>
          <mat-icon class="mat-icon-rtl-mirror">
            {{ nestedTreeControl.isExpanded(node) ? 'expand more' : 'chevron right' }}
         </mat-icon>
        </button>
        <button mat-button (click)="onClickNodeTree(node)">
           {{ node.name }}
       </button>
      </div>
      <div [class.example-tree-invisible]="!nestedTreeControl.isExpanded(node)" role="group">
       <ng-container matTreeNodeOutlet></ng-container>
      </div>
   </mat-nested-tree-node>
  </mat-tree>
```

Note: This report is based on work carried out as part of the FOSSR project (https://www.fossr.eu/) at the InfraScience laboratory of the CNR Institute of Information Science and Technology 'A. Faedo' in Pisa, Italy (https://infrascience.isti.cnr.it/).