SailPoint

Developer Days2023

</title>

Plugin Development Frontend Solution with Angular/Typescript

</title>

Speaker: Yu-Chih Chung (Mike);
title: Senior Solution Consultant, CISSP, SailPoint Certified IdentityIQ Architect;
company: KOGIT GmbH;

Agenda

- Introduction
- IDE Setup
- Integrate Angular/Typescript with IdentityIQ Plugin Development (KOGIT Org-Chart Plugin)
- Demo
- References
- FAQ (on developer community)

Who's KOGIT

- A Europe-wide recognized and leading consulting firm, we aim to develop secure and innovative solutions for customized Identity and Access Management (IAM).
- SailPoint 2022 Delivery Admiral.

KOGIT

KOGIT Consulting and Services

Analysis

Analysis of existing structures and processes (gap analysis / IAM assessment)

Consulting

Consulting in the area of technology, strategy and processes

Conception & Design

Product evaluation, feasibility studies, concepts, prototype development

Implementation

Implementation, customer-specific enhancements, roll-out, operation

Support

2nd & 3rd Level Solution Support Service

KOGIT Services

Identity Management
Structured and transparent management of identities of all types

Access Management
Which identity gets how, when & with which rights access to which IT systems & data

Compliance
Regulatory compliance through credential monitoring, analysis & periodic recertification.

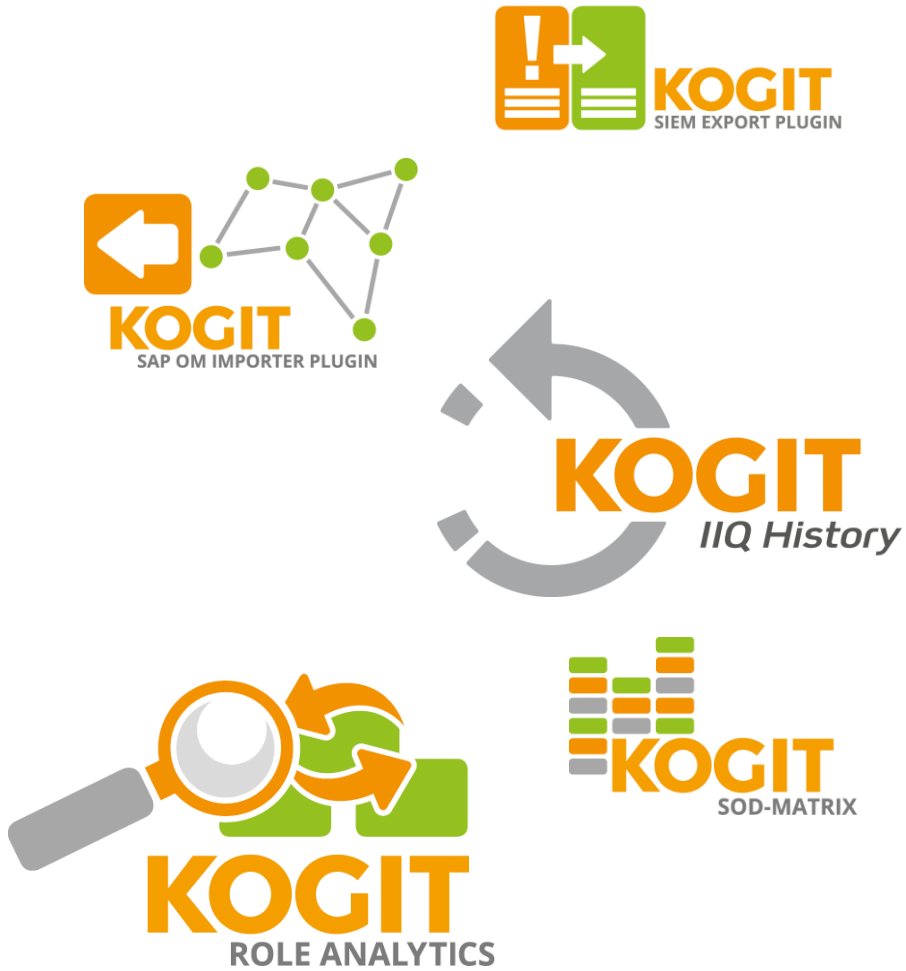
Privileged Access Management
Protect privileged user accounts & access rights from data breaches through efficient control

Security Information & Event Management
Collection & analysis of messages and log files, real-time detection of potential threats

Advanced Analytics
Identify risks of unauthorized data access/theft at an early stage & initiate defensive measures

KOGIT Plugin Library

- KOGIT History Plugin
- KOGIT Role Analytics Plugin
- KOGIT SIEM Plugin
- KOGIT SoD-Matrix Plugin
- KOGIT Monitoring Plugin
- KOGIT Assignment Calendar Plugin
- KOGIT Access Request Extension Plugin
- KOGIT Org Chart Plugin




if localVarHTTPResponse.StatusCode >= 300 {
newErr := &GenericOpenAPIError{
body: localVarBody,
error: localVarHTTPResponse.Status,
}
if localVarHTTPResponse.StatusCode == 400 {
var v ErrorResponseDto
err = a.client.decode(&v, localVarBody, localVarHTT
if err != nil {
newErr.error = err.Error()
return localVarReturnValue, localVarHTTPResp
}

IDE Setup

Suggested Knowledge

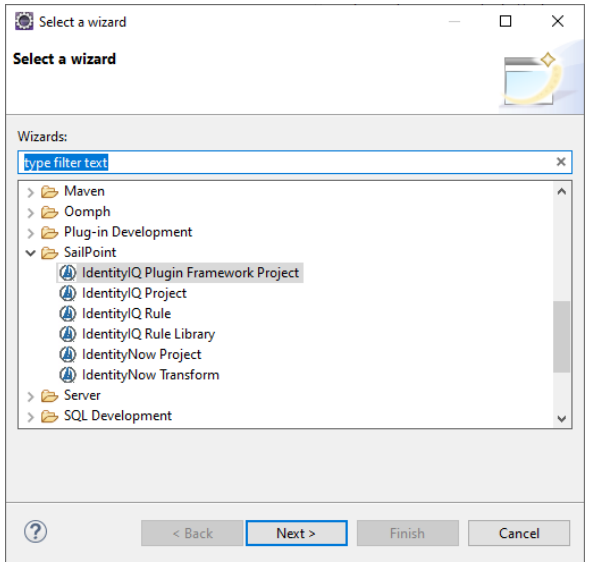
- Plugin Developer Guide – Overview
- SailPoint University: IdentityIQ Advanced Implementation: Plugins v8.3 - ELEARNING
- [YouTube] Angular Tutorial for Beginners: Learn Angular & TypeScript

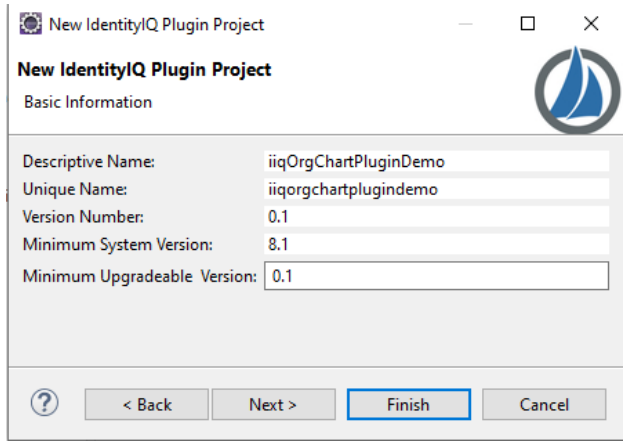
Environment Setup

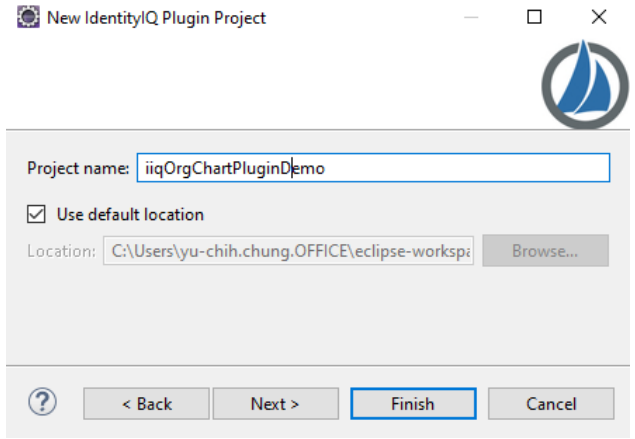
- NodeJS
- Eclipse + IIQDA: for Java development
- Visual Studio Code: for Angular/Typescript development
  - Angular Language Service
  - Auto Close Tag
  - Auto Import
  - Auto Import – ES6, TS, JSX, TSX
  - ES Lint
- IIQ 8.3 instance

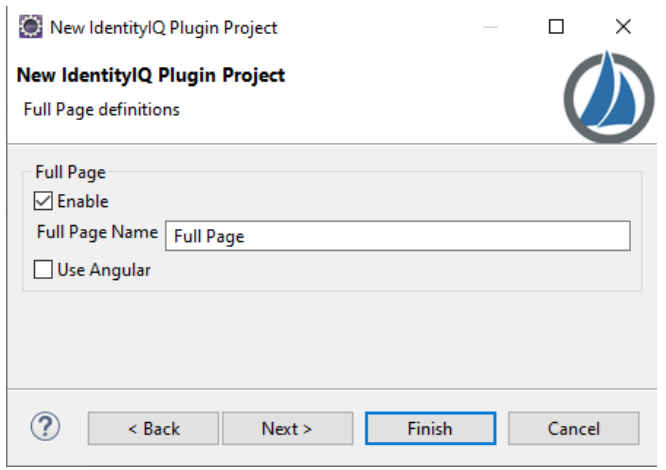
Initiate Plugin Framework Project

- File > Other > IdentityIQ Plugin Framework Project
- Delete files under import/ and db/









Create Java Classes

- Create packages under src/
  - demo.spdeveloper.plugin.orgchart.rest
  - demo.spdeveloper.plugin.orgchart.services
- Create Java Classes
  - demo.spdeveloper.plugin.orgchart.rest.IdentityResource.java
  - demo.spdeveloper.plugin.orgchart.service.IdentityService.java
  - demo.spdeveloper.plugin.orgchart.service.PluginSettingService.java

Angular Project Setup

- Install Angular Command Line
  - npm install -g @angular/cli
- Create new Angular Project
  - ng new iiqOrgChartPluginDemo

```
PS C:\Users\yu-chih.chung.OFFICE\eclipse-workspace\iiqOrgChartPluginDemo\src-ui> ng add @angular/material
i Using package manager: npm
√ Found compatible package version: @angular/material@15.1.1.
√ Package information loaded.

The package @angular/material@15.1.1 will be installed and executed.
Would you like to proceed? Yes
√ Packages successfully installed.
? Choose a prebuilt theme name, or "custom" for a custom theme: Indigo/Pink        [ Preview: https://material.angular.io?theme=indigo-pink ]
? Set up global Angular Material typography styles? Yes
? Include the Angular animations module? Include and enable animations
UPDATE package.json (1122 bytes)
√ Packages installed successfully.
UPDATE src/app/app.module.ts (525 bytes)
UPDATE angular.json (2913 bytes)
UPDATE src/index.html (589 bytes)
UPDATE src/styles.css (181 bytes)
PS C:\Users\yu-chih.chung.OFFICE\eclipse-workspace\iiqOrgChartPluginDemo\src-ui>
```

Angular Project Setup

- Rename folder as /src-ui
- Install Angular Material Module
  - ng add @angular/material
- Install D3-Org-Chart
  - npm i d3-org-chart
- Create sub-folder /src-ui/src/app/component/
- Under ~/component/ folder, create component with Angular CLI
  - ng generate component d3-org-chart
- Create sub-folder /src-ui/src/app/services/
- Under ~/services/ folder, create service with Angular CLI
  - ng generate service data

Angular Project Setup

- tsconfig.json
  - `"compilerOptions": {`

```
…
"experimentalDecorators": false
…
}
```

- angular.json
  - `"outputPath": "dist/"`
  - `"outputHashing": "none"`

Ant build.xml Customization

```
<target name="npm-build">
  <exec dir="./src-ui" executable="cmd.exe" output="npm-build.txt">
    <arg line="/c npm run build --prod"/>
  </exec>
</target>
```

Ant build.xml Customization

```
<target name="package" depends="compile,generateDTD,npm-build"
description="Output zip file overlay to build/dist dir">
  <copy todir="${overlay}/ui">
    <fileset dir="web">
    <include name="page.xhtml"/>
      </fileset>
    </copy>
  <copy todir="${overlay}/ui/js">
      <fileset dir="src-ui/dist">
        <include name="*.js"/>
      </fileset>
    <fileset dir="web/js">
    <include name="*.js"/>
      </fileset>
    </copy>
  <copy todir="${overlay}/ui/css">
      <fileset dir="src-ui/dist">
        <include name="*.css"/>
      </fileset>
    </copy>
</target>
```

Folder Structure

Java Implementation

Database scripts

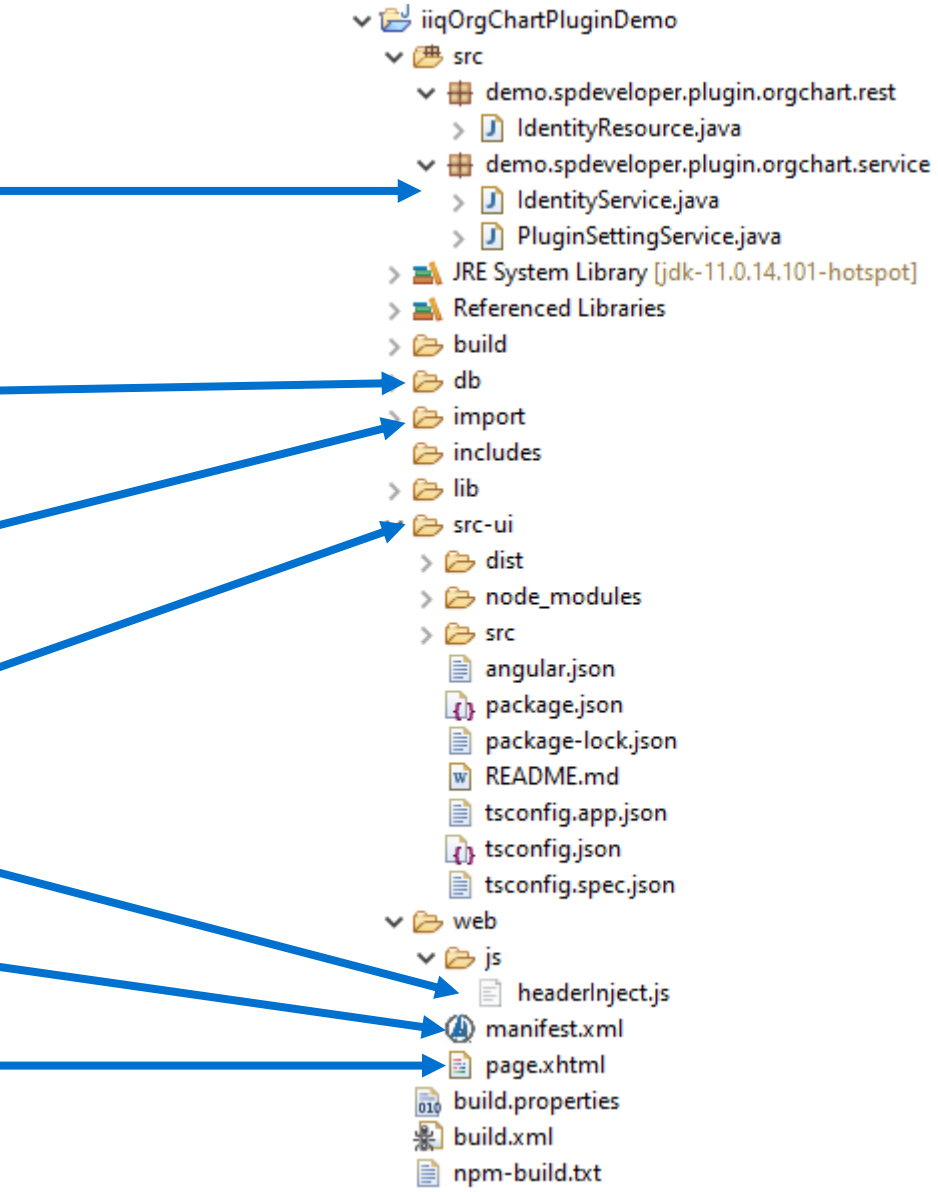
SailPoint objects to be imported

Angular project

JavaScript

Manifest file

Full page xhtml



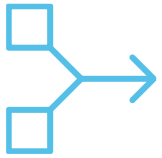
Angular/Typescript with Plugin Development

KOGIT Org-Chart Plugin

What will you achieve in this session

Single page web application

Able to build a single-page web application with Angular Framework and TypeScript



Utilize Angular Modules and 3rd libraries

Utilize Angular modules in frontend development: Angular Material, Angular HTTP and d3-org-chart

Angular Frontend in Plugin Framework

Integrate Angular frontend with PluginHelper and build with SailPoint IdentityIQ Plugin Framework

Angular & TypeScript

- Angular is **an open-source, JavaScript framework written in TypeScript,** its primary purpose is to develop single-page applications.

- TypeScript **extends JavaScript and improves the developer experience**. It is a strict syntactical superset of JavaScript and adds optional static typing to the language.

KOGIT Org-Chart Plugin

- Visualize Organization Hierarchy in Org-Chart graphical view
- Why use Org-Chart in IdentityIQ instead of HCM Tools
  - IdentityIQ has not only HR Data, but also non-HR data, for example, contractors, shared/technical identities, workgroups.
  - Better visualization in terms of reporting, analysis and monitoring instead of csv or excel format.

Org Chart Plugin Features

- The Org-Chart hierarchy is based on the Manager(Owner for workgroup) attribute on each Identity.
- Configurable color code for each identity type (e.g., employee, contractor, workgroup, partner).
- Change Org-Chart display: Compact, Swap Direction, Collapse/Expand.

D3-Org-Chart

https://github.com/bumbeishvili/org-chart

Highly customizable d3 org chart. Integrations available for Angular, React, Vue.

The idea is to reuse a 3rd party frontend library for Plugin development.

D3-Org-Chart: Data Model

- The D3-Org-Chart library can only support a single root tree.
- Every parentId value must exists in the library data set.
- Here we use IdentityIQ internal uuid as the unique identifier in the D3-Org-Chart.
- Additional attributes such as department, job title, email are configurable in the Plugin Setting.


```
{
    "identityIconImgAttribute":"iconImg",
    "jobtitle":"Technical Lead7",
    "displayName":"Jack Chung",
    "membershipCount":0,
    "type":"contractor",
    "parentId":"c0a80f817f021f9b817f022212420045",
    "inactive":false,
    "managesCount":0,
    "name":"W0001",
    "colorCode":"#8A03FA",
    "attributes":[
        "department",
        "jobtitle",
        "email"
    ],
    "id":"c0a80f817f021f9b817f02220c580037",
    "department":"IT",
    "email":"jack.chung@kogit.de"
}
```

REST Endpoint IdentityResource

- Path: <SAILPOINT_CONTEXT>/plugin/rest/iiqorgchartplugindemo/orgchart/{id}
- Method: GET
- SPRight: None


```java
@Path("iiqorgchartplugindemo")
@AllowAll
public class IdentityResource extends BasePluginResource {

    @GET
    @Path("orgchart/{id}")
    @Produces(MediaType.APPLICATION_JSON)
    public ResponseEntity<Map<String, Object>> getOrgChart(@PathParam("id") final String id) {
        SailPointContext context = this.getContext();
        HttpStatus responseStatus = HttpStatus.OK;
        String message = null;
        List<Map<String, Object>> nodes = null;
        try {
            nodes = IdentityService.getTreeNodes(context, id);
        } catch (GeneralException e) {
            responseStatus = HttpStatus.INTERNAL_SERVER_ERROR;
            message = e.getMessage();

        }
        Map<String, Object> responseBody = new HashMap<String, Object>();
        responseBody.put("nodes", nodes);
        responseBody.put("message", message);
        return ResponseEntity.status(responseStatus).body(responseBody);
    }

    @Override
    public String getPluginName() {
        return PluginSettingService.getPluginName();
    }

}
```

IdentityService Class: Outline

This class is mainly used to retrieve the identities which selected to be display in the Org Chart.

The public method getTreeNodes() is the main function within this class.

IdentityService Class: Outline

The below functions are used to support the public function getTreeNodes:

- getParents()
- getSiblings()
- getChilds()
- getNodeMap()

```
public static List<Map<String, Object>> getTreeNodes(SailPointContext context, String id) throws GeneralException {
    Identity identity = context.getObjectById(Identity.class, id);
    if (identity == null) {
        throw new GeneralException("Can not find identity object: " + id);
    }
    List<Map<String, Object>> nodes = new ArrayList<>();
    List<String> all = new ArrayList<>();
    List<String> childs = getChilds(context, id);
    logger.trace("childs: " + childs);
    all.addAll(childs);
    List<String> parents = getParents(context, identity, PluginSettingService.getSettingManagerLevels(), new ArrayList<>());
    String rootId = null;
    if (Util.nullSafeSize(parents) > 0) {
        rootId = parents.get(parents.size() - 1);
    } else {
        rootId = id;
    }
    logger.trace("parents: " + parents);
    all.addAll(parents);
    List<String> siblings = getSiblings(context, identity);
    logger.trace("siblings: " + siblings);
    all.addAll(siblings);
    Util.removeDuplicates(all);
    for (String _id : Util.safeIterable(all)) {
        Identity node;
        try {
            node = context.getObjectById(Identity.class, _id);
        } catch (GeneralException e) {
            logger.error("Failed to retrieve Identity object(" + _id + "): " + e);
            continue;
        }
        // Mandatory attributes: id, name, parentId, displayName
        try {
            nodes.add(getNodeMap(context, node, rootId));
        } catch (GeneralException e) {
            logger.error("Failed to build node map(" + _id + "): " + e);
            continue;
        }
    }
    if (logger.isTraceEnabled()) {
        logger.trace("Exit getTreeNodes, nodes: " + nodes);
    }
    return nodes;
}
```

IdentityService Class: getParents

This is a recursive function, it searches the manager of identity or owner of workgroup up to the pre-configured level of the provided identity.

```
private static List<String> getParents(SailPointContext context, Identity identity, int level, List<String> reportTos) {
    logger.trace("getParents, level: " + level);
    logger.trace("getParents, identityName: " + identity.getName());
    logger.trace("getParents, reportTos: " + reportTos);
    if (level == 0) {
        return reportTos;
    } else {
        if (identity.getManager() != null) {
            String managerId = identity.getManager().getId();
            if (reportTos.contains(managerId)) {
                // Close loop found
                return reportTos;
            } else {
                reportTos.add(managerId);
                return getParents(context, identity.getManager(), --level, reportTos);
            }
        } else if (identity.isWorkgroup() && identity.getOwner() != null)  // Workgroup
            String ownerId = identity.getOwner().getId();
            if (reportTos.contains(ownerId)) {
                // Close loop found
                return reportTos;
            } else {
                reportTos.add(ownerId);
                return getParents(context, identity.getOwner(), --level, reportTos);
            }
        } else {
            return reportTos;
        }
    }
}
```

IdentityService Class: getChilds

The below functions are used to support the public function getChilds:

- getManagedMembers(): retrieve all identities who's manager is the selected user.
- getOwnedWorkgroups(): retrieve all workgroups which it's owner is the selected user.

```
private static List<String> getChilds(SailPointContext context, String id)
        throws GeneralException {
    List<String> childs = new ArrayList<>();
    childs.addAll(getManagedMembers(context, id));
    childs.addAll(getOwnedWorkgroups(context, id));
    return childs;
}

private static List<String> getManagedMembers(SailPointContext context, String id)
        throws GeneralException {
    List<String> members = new ArrayList<>();
    QueryOptions qo = new QueryOptions();
    Filter filter = Filter.eq("manager.id", id);
    qo.add(filter);
    Iterator<Object[]> it = context.search(Identity.class, qo, "id");
    IdIterator idIt = new IdIterator(context, it);
    while (idIt.hasNext()) {
        String _id = idIt.next();
        members.add(_id);
    }
    return members;
}

private static List<String> getOwnedWorkgroups(SailPointContext context, String id)
        throws GeneralException {
    List<String> workgroups = new ArrayList<>();
    QueryOptions qo = new QueryOptions();
    Filter filter = Filter.and(Filter.eq("owner.id", id), Filter.eq("workgroup", true));
    qo.add(filter);
    Iterator<Object[]> it = context.search(Identity.class, qo, "id");
    IdIterator idIt = new IdIterator(context, it);
    while (idIt.hasNext()) {
        String _id = idIt.next();
        workgroups.add(_id);
    }
    return workgroups;
}
```

IdentityService Class: getSiblings

Re-use getChilds() function to get all child nodes (identity and workgroup) which managed/owned by the provided user's manager.

```
private static List<String> getSiblings(SailPointContext context, Identity identity)
        throws GeneralException {
    List<String> siblings = new ArrayList<>();
    Identity parent = null;
    if (identity.isWorkgroup()) {
        parent = identity.getOwner();
    } else {
        parent = identity.getManager();
    }
    if (parent != null) {
        siblings = getChilds(context, parent.getId());
    } else { // No manager or owner
        siblings.add(identity.getId());
    }
    return siblings;
}
```

IdentityService Class: getNodeMap

This function is similar to a data transfer object (DTO) class which translate IIQ object into the required frontend data format.

The below functions are used to support the public function getNodeMap:

- getWorkgroupNodeMap(): convert workgroup into required data model.
- getIdentityNodeMap(): convert identity into required data model.

As required by d3-org-chart, attribute (id) is required to be the unique identifier for the library.

```
private static Map<String, Object> getNodeMap(SailPointContext context, Identity node, String rootId)
        throws GeneralException {
    if (logger.isTraceEnabled()) {
        logger.trace("Enter getNodeMap...");
        logger.trace("Node: " + node.getName());
        logger.trace("Root Id: " + rootId);
    }
    Map<String, Object> nodeMap = new HashMap<>();
    nodeMap.put(ATTR_ID, node.getId());
    nodeMap.put(ATTR_NAME, node.getName());
    nodeMap.put(ATTR_DISPLAYNAME, node.getDisplayableName());
    if (node.isWorkgroup()) {
        nodeMap.putAll(getWorkgroupNodeMap(context, node, rootId));
    } else {
        nodeMap.putAll(getIdentityNodeMap(context, node, rootId));
    }

    if (logger.isTraceEnabled()) {
        logger.trace("Exit getNodeMap, nodeMap: " + nodeMap);
    }
    return nodeMap;
}
```

IdentityService Class: getNodeMap

Both functions map the owner/manager field as the required attribute (parentId) and enrich with additional attributes information.

```
private static Map<String, Object> getWorkgroupNodeMap(SailPointContext context, Identity node, String rootId)
        throws GeneralException {
    Map<String, Object> map = new HashMap<>();
    map.put(ATTR_TYPE, TYPE_WORKGROUP);
    map.put(ATTR_COLOR_CODE, getColorCode(TYPE_WORKGROUP));
    if (rootId != null && rootId.equals(node.getId())) {
        map.put(ATTR_PARENT, null);
    } else {
        map.put(ATTR_PARENT, getParentNodeId(node, node.getOwner()));
    }
    map.put(ATT_MANAGES_COUNT, getManagedCount(context, node.getId()));
    // Custom attributes
    List<String> customWorkgroupCardAttrs = PluginSettingService.getSettingWorkgroupCardAttrs();
    for (String attr : Util.safeIterable(customWorkgroupCardAttrs)) {
        if ("description".equals(attr)) {
            map.put(attr, node.getDescription());
        } else {
            map.put(attr, node.getAttribute(attr));
        }
    }
    map.put(ATTR_ATTRIBUTES, customWorkgroupCardAttrs);
    try {
        map.put(ATTR_MEMBER_COUNT, Util.nullSafeSize(getMembers(context, node, "id")));
    } catch (GeneralException e) {
        logger.error("Failed to get members for workgroup: " + node.getName());
    }
    return map;
}

private static Map<String, Object> getIdentityNodeMap(SailPointContext context, Identity node, String rootId)
        throws GeneralException {
    Map<String, Object> map = new HashMap<>();
    map.put(ATTR_TYPE, (node.getType() != null ? node.getType() : "none"));
    map.put(ATTR_COLOR_CODE, getColorCode(node.getType()));
    if (rootId != null && rootId.equals(node.getId())) {
        map.put(ATTR_PARENT, null);
    } else {
        map.put(ATTR_PARENT, getParentNodeId(node, node.getManager()));
    }
    map.put(ATT_MANAGES_COUNT, getManagedCount(context, node.getId()));
    map.put(ATTR_INACTIVE, node.isInactive());
    // Custom attributes
    List<String> customIdentityCardAttrs = PluginSettingService.getSettingIdentityCardAttrs();
    for (String attr : Util.safeIterable(customIdentityCardAttrs)) {
        map.put(attr, node.getAttribute(attr));
    }
    map.put(ATTR_ATTRIBUTES, customIdentityCardAttrs);
    map.putAll(getIconImg(node));
    // Workgroups
    List<Identity> workgroups = node.getWorkgroups();
    map.put(ATT_MEMBERSHIP_COUNT, Util.nullSafeSize(workgroups));
    return map;
}
```

Angular Http Module

The Angular Http module enables the frontend to retrieve data from backend RESTful endpoints.

Angular provides a client HTTP API for Angular applications, the HttpClient service class in @angular/common/http

Angular Http Module: Data Service

Here in the data.service.ts we will implement the Angular http module to send rest api call to the plugin backend.

The HttpClient, Observalbe... libraries are required.

As we will use PluginHelper and SailPoint object which provided by standard **SailPointBundleLibrary.js**, here we need to declare the used function within typescript. This tells the compiler that the source exists in another file.

Security: TLS!

```
import { HttpClient, HttpErrorResponse, HttpHeaders } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { Observable, Observer } from 'rxjs';
import { environment } from 'src/environments/environment';

declare const PluginHelper: {
  getCsrfToken: Function
  getPluginRestUrl: Function
};

declare const SailPoint: {
  CONTEXT_PATH: string
};
```

Angular Http Module: Data Service

Here we use PluginHelper to retrieve the plugin URL and CRSF token for authentication.

We use SailPoint object to get the SailPoint context URL.

Within Angular development, we can specify different environments (e.g. production, dev), and specify different parameters for each environment.

Reference: PluginHelper Methods

```
export class DataService {
  token: string = '';
  pluginUrl: string = '';
  iiqUrl: string;

  constructor(
    private http : HttpClient,
  ) {
    if (environment.production) {
      this.pluginUrl = PluginHelper.getPluginRestUrl('iiqorgchartplugindemo');
      this.iiqUrl = SailPoint.CONTEXT_PATH;
      this.token = PluginHelper.getCsrfToken();
    } else {
      this.pluginUrl = 'http://192.168.15.129:8080/identityiq/plugin/rest/iiqorgchartplugindemo';
      this.iiqUrl = 'http://192.168.15.129:8080/identityiq';
    }
  }

  private defaultHeaders = () => (
    new HttpHeaders({
      "X-XSRF-TOKEN": this.token,
      "Content-Type": "application/json"
    })
  )
```

Angular Http Module : Data Service

The *fetch* function use GET method to call the RESTful endpoint from the input path parameter.

In Angular Observable provides support for passing messages between parts of the application. Whenever a new value or a change in data is detected the logic described in .subscribe will be executed.

```
fetch = (path: string) => {
  return new Observable((observer: Observer<any>) => {
    const url = this.pluginUrl + path;
    let headers;
    if (environment.production) {
      headers = this.defaultHeaders();
    } else {
      headers = this.defaultHeadersDev();
    }

    this.http.get<Request>(url, { headers : headers }).subscribe((res: any) => {
      if (res === null) {
        console.error("Response is null.");
      } else if (res['statusCodeValue'] && (res.statusCodeValue.toString().startsWith('2') === false) ) {
        if (res.body && res.body.message && res.body.message.length > 0) {
          console.error(res.body.message);
        }
      }
      observer.next(res);
      observer.complete();
    })

  });
}
```

Angular Material

- Public Angular frontend components and APIs
- Easy usage with examples and documentation
- Offers re-usable, beautiful, attractive UI components like Cards, Date Picker, Data Table and more.

Angular Material: Auto-Complete

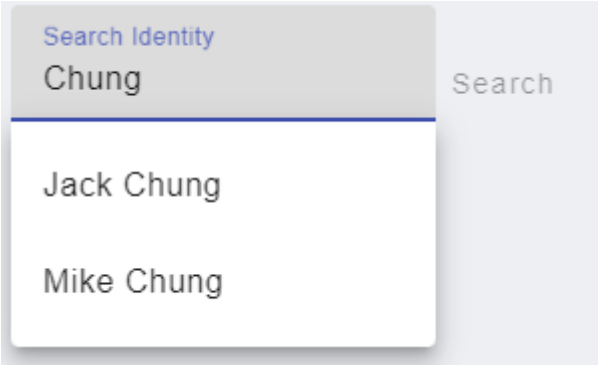
In the d3-org-chart.component.html, we define the **formControl** and **matAutocomplete**.

The mat-autocomplete section will execute the function defined in the formControl and reflect the data changes synchronously in the **mat-option** directives.

```
<div class="orgChartContainer">
  <!-- Org Chart -->
  <div class="chart-container"></div>
  <!-- Search Box -->
  <div class="search-box">
    <form>
      <mat-form-field>
        <mat-label>Search Identity</mat-label>
        <input
          type="text"
          placeholder="Identity Name or DisplayName"
          matInput
          [formControl]="myIdentityControl"
          [matAutocomplete]="idententiyAuto">
        <mat-autocomplete #idententiyAuto="matAutocomplete" [displayWith]="displayFn">
          <mat-option *ngFor="let option of identityOptions" [value]="option">
            {{ option.displayName }}
          </mat-option>
        </mat-autocomplete>
      </mat-form-field>
      <button
        mat-button
        color="primary"
        [disabled]="selected === undefined"
        (click)="getOrgChart()"
        type="button">
        Search
      </button><br/>
    </form>
  </div>
```

Angular Material: Auto-Complete

We defined a valueChanges function in the form control. It will call the out-of- the-box Identity Suggest endpoint every 0.5 sec. And refresh the selectable results under mat-options drop-down list.



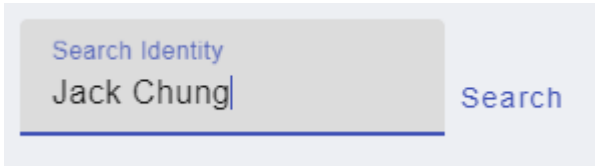

```
ngOnInit(): void {
  this.initIdentityOptions();
  this.getIdentityOptions('');
}

private initIdentityOptions = () => {
  this.myIdentityControl.valueChanges
    .pipe(debounceTime(500))
    .subscribe(input => {
      if (typeof input === "string" && input && input.length) {
        this.selected = undefined;
        this.getIdentityOptions(input);
      } else {
        if (this.myIdentityControl.value instanceof Object) {
          this.selected = this.myIdentityControl.value;
        }
      }
  });
}

private getIdentityOptions = (queryStr: string) => {
  console.log('Retrieving data for query: ' + queryStr);
  const filter = { query: queryStr, limit: 10, extraParams: { context: "Global", suggestId: "IncludeWorkGroups"}};
  const path = '/ui/rest/suggest/object/sailpoint.object.Identity';
  this.dataService.fetchSuggest(path, filter).subscribe(res => {
    if (res && res.objects) {
      const items = res.objects.map((obj:any) => {
        return { name: obj.name, displayName: obj.displayableName, id: obj.id }
      })
      this.identityOptions = items;
      console.log('getIdentityOptions:' + this.identityOptions.length);
    }
  });
}
```

Search Button: getOrgChart()

Once the identity is selected, we can then click on the search button.

The button click event will trigger the getOrgChart() function.

```
<button
  mat-button
  color="primary"
  [disabled]="selected === undefined"
  (click)="getOrgChart()"
  type="button">
  Search
</button><br/>
```



getOrgChart()

The getOrgChart() function will utilize data service to retrieve the data from the plugin endpoint. And call updateChart() function to initialize the Org-Chart.

```
getOrgChart = () => {
  if (this.data.length !== 0) {
    this.data = [];
  }
  this.nodePointer = (' ' + this.selected?.id).slice(1);
  const path = `/orgchart/${this.nodePointer}`;
  this.dataService.fetch(path).subscribe(res => {
    if (res && res.body && res.body.nodes) {
      this.data = res.body.nodes;
      console.log("getOrgChart, data: ");
      console.log(this.data);
      this.updateChart();
      console.log(this.nodePointer);
      if (this.nodePointer !== undefined) {
        this.mark(this.nodePointer);
        this.expandAll();
      }
    }
  });
}
```

D3-Org-Chart Initialize: updateChart()

A new OrgChart() object is initialized with parameters to define the node attributes. The actual node graphic display is constructed under .nodeContent() function.

```
updateChart = () => {
  if (!this.data) {
    return;
  }
  this.chart = new OrgChart()
    .container('.chart-container')
    .data(this.data)
    .nodeWidth((d) => 300)
    .initialZoom(0.7)
    .nodeHeight((d) => 240)
    .childrenMargin((d) => 40)
    .compactMarginBetween((d) => 15)
    .compactMarginPair((d) => 80)
    .nodeContent(function (d: any, i: any, arr: any, state: any) {
```

D3-Org-Chart: Customizable Card

Customizable HTML

```
return `
  <div style="padding-top:30px;background-color:none;margin-left:1px;height:${d.height}px;border-radius:2px;overflow:visible;">`+
    `<!-- image icon -->
    <div id="${d.data.id}" style="height:${d.height - 32}px;padding-top:0px;background-color:white;border:1px solid lightgray;">` +
      iconImgDiv
      + `<div style="margin-right:10px;margin-top:15px;float:left">` +
      status
      + `</div>
      <!-- ID -->
      <div style="margin-right:10px;margin-top:15px;float:right">
        ${d.data.name}
      </div>
      <div style="margin-top:-30px;background-color:${d.data.colorCode};height:10px;width:${d.width - 2}px;border-radius:1px">
      </div>
      <div style="padding:20px; padding-top:35px;text-align:center">
        <div style="color:#111672;font-size:16px;font-weight:bold">
          ${d.data.displayName}
        </div>` +
        customAttrsDiv
        + `<div style="display:flex;justify-content:space-between;padding-left:15px;padding-right:15px;">
          <div> Manages:  ${d.data.managesCount} 👤</div> ` +
        workgroupMemberCountDiv
        + `</div>
      </div>
    </div>
  </div>
`;
```

User Card

D3-Org-Chart: Display Features

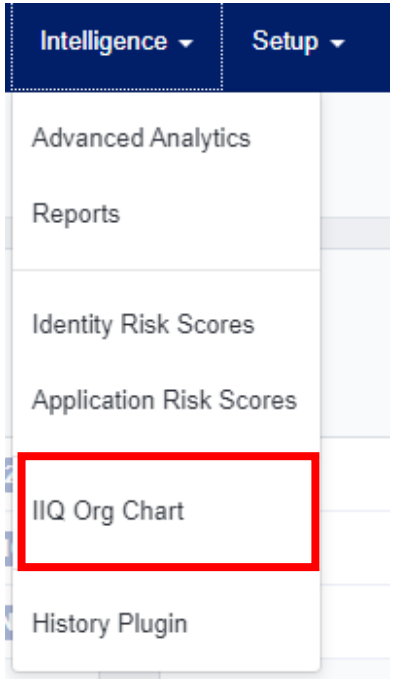
Library APIs

```
expandAll = () => {
  this.chart.expandAll();
}

collapseAll = () => {
  this.chart.collapseAll();
}

compactDisplay = () => {
  this.chart.compact(!!(this.compact++%2)).render().fit()
}

swapDirection = () => {
  this.chart.layout(["right","bottom","left","top"][this.index++%4]).render().fit()
}
```

Action Buttons

JavaScript Snippet

Use JQuery to insert the plugin page hyperlink under Menu > Intelligence > Application Risk Scores




```
jQuery(document).ready(function() {

    var orgChartPluginUrl = SailPoint.CONTEXT_PATH + '/plugins/pluginPage.jsf?
    pn=iiqorgchartplugindemo';
    var check = jQuery("a[href$='/manage/riskScores/appRiskScores.jsf']");

    if (check.length > 0) {

        jQuery("a[href$='/manage/riskScores/appRiskScores.jsf']").parent().after(
            '<li role="presentation" aria-hidden="true" class="divider"></li>' +
            '<li role="presentation"><a href="'+ orgChartPluginUrl +'" role="menuitem"
            class="menuitem" tabindex="0">IIQ Org Chart</a></li>'
        );

    } else {

        jQuery("a[href$='/analyze/reports/viewReports.jsf?resetTab=true&forceLoad=true']").
        parent().after(
            '<li role="presentation" aria-hidden="true" class="divider"></li>' +
            '<li role="presentation"><a href="'+ orgChartPluginUrl +'" role="menuitem"
            class="menuitem" tabindex="0">IIQ Org Chart</a></li>'
        );

    }

});
```

Manifest

Define the rest resources with the backend java class.

Define the JavaScript snippets file.

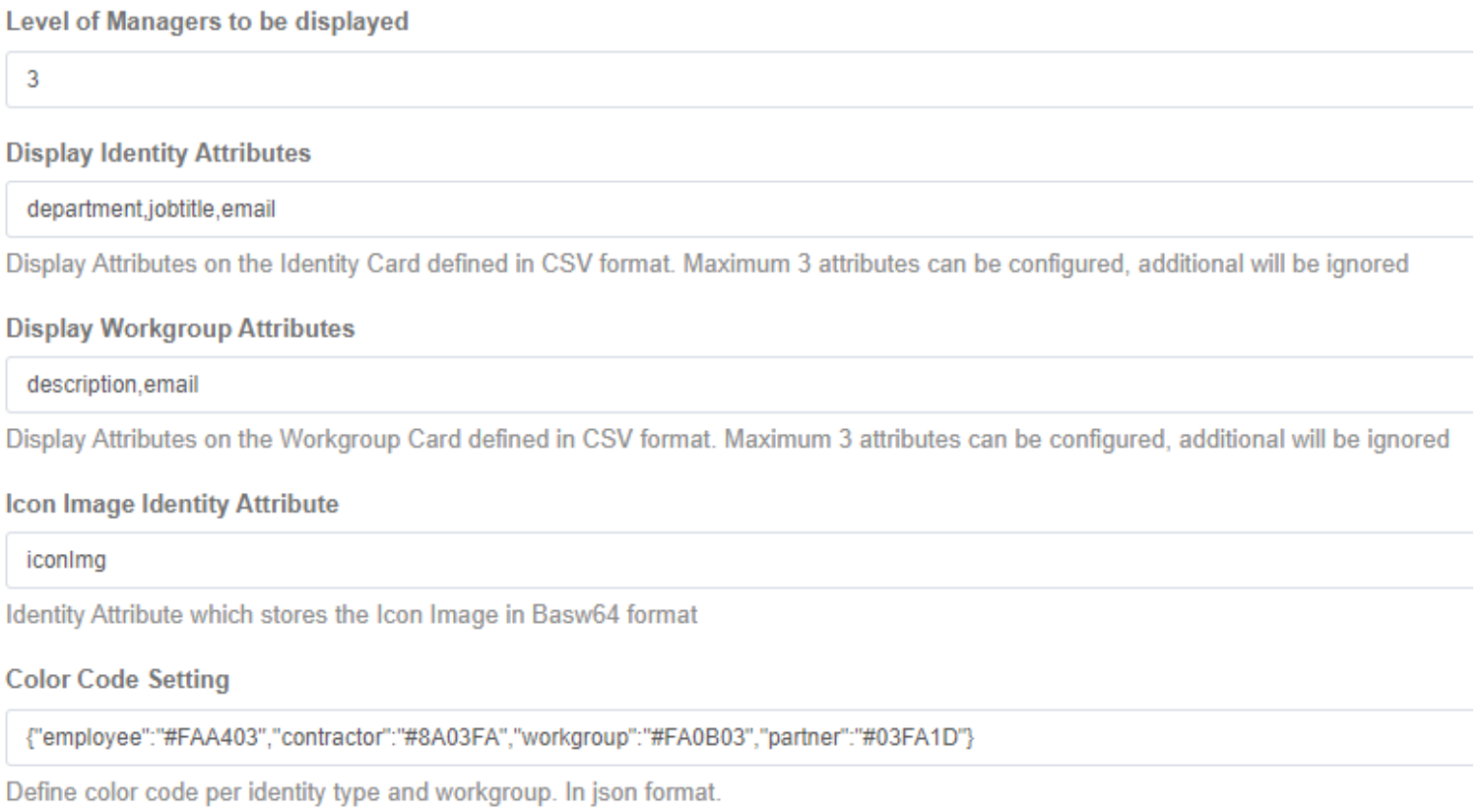
```
<entry key="restResources">
  <value>
    <List>
      <String>demo.spdeveloper.plugin.orgchart.rest.IdentityResource</String>
    </List>
  </value>
</entry>
<entry key="snippets">
  <value>
   <List>
    <Snippet regexPattern=".*">
     <Scripts>
      <String>ui/js/headerInject.js</String>
     </Scripts>
    </Snippet>
   </List>
  </value>
 </entry>
```

Plugin Settings

The plugin is designed with below configurable attributes:

- Level of Managers to be displayed
- Display Identity Attributes
- Display Workgroup Attribute
- Icon Image Identity Attribute
- Color Code Setting

A Java Class (PluginSettingService.java) is the helper class for retrieving the value from Plugin Setting.



Page.xhtml

In the full page (page.xhtml) we simply configure <app-root> directive, it will be recognized by the Angular framework.

During the build process, the command “npm run build --prod“ is executed. NodeJS will compile the Angular project into several JavaScript files. Here we simply need to include all the compiled JavaScript files in the xhtml.

```
<ui:composition>
    <app-root style="background:#ffffff; top: 100px !important; height: 100%;  bottom: 0 !important;"></app-root>
    <script src="#{plugins.requestContextPath}/plugin/#{plugins.pluginName}/ui/js/runtime.js"></script>
    <script src="#{plugins.requestContextPath}/plugin/#{plugins.pluginName}/ui/js/polyfills.js"></script>
    <script src="#{plugins.requestContextPath}/plugin/#{plugins.pluginName}/ui/js/main.js"></script>
    <link href="#{plugins.requestContextPath}/plugin/#{plugins.pluginName}/ui/css/styles.css" rel="stylesheet"></link>
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet"></link>
</ui:composition>
```

Ant Build: Plugin ZIP File Content

Use Eclipse to execute Ant build with target package.

The compiled package (.zip) is under build/ folder.

```
if localVarHTTPResponse.StatusCode >= 300 {
        newErr := &GenericOpenAPIError{
                body:  localVarBody,
                error: localVarHTTPResponse.Status,
        }
```

Demo

```
if localVarHTTPResponse.StatusCode == 400 {
                var v ErrorResponseDto
                err = a.client.decode(&v, localVarBody, localVarHTT
                if err != nil {
                        newErr.error = err.Error()
                        return localVarReturnValue, localVarHTTPResp
                }
```

References

- KOGIT Plugin Library
- Source code used in this session is available on Developer Community post.

FAQ

Please leave your questions on Developer Community post.

[Thank you!]

KOGIT GmbH | Rheinstr. 40-42 | 64283 Darmstadt | www.kogit.de

Tel: +49 6151 7869-0 | Fax: +49 6151 7869-370