SIEMENS
Ingenuity for life

1st Jazz Community Hackathon

Learn how to build Jazz Extensions

jazz-community.org | github.com/jazz-community

Martin Benninger / Lukas Steiger - November 2018

Goals and Agenda

SIEMENS
Ingenuity for life

Goal

- Help you to get started with extending Jazz
- Share our experience we had
- Create something useful – not just talk ☺

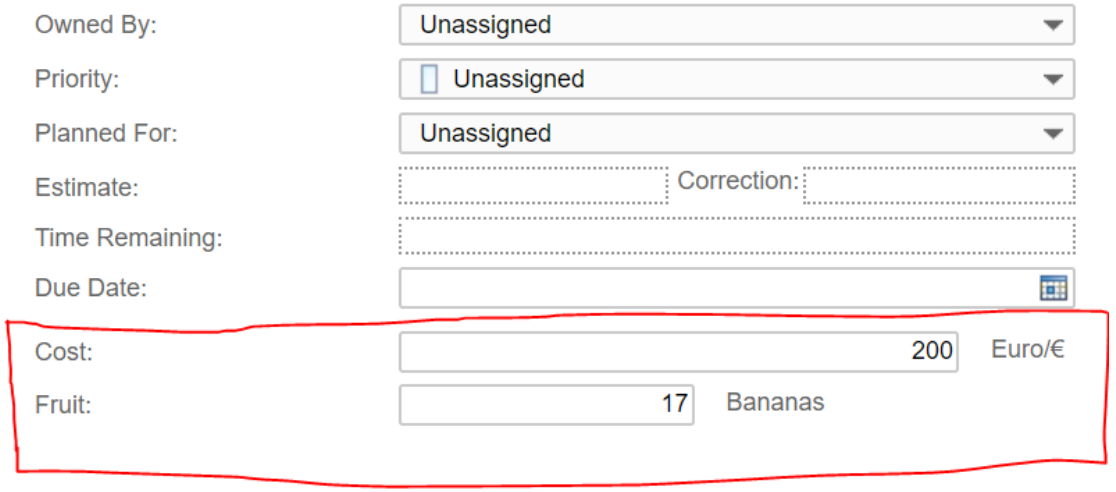
Agenda

09:00	Introduction
09:20	Dev. Environment Setup and Coffee Break
10:10	Client Side Extensions
10:45	Coding Session – Part 1
12:30	Lunch Break
13:30	Coding Session – Part 2
15:30	Presentation of Results
16:15	Input on Server Side Extensions
16:50	Bye bye

What We Will Create Today

SIEMENS
Ingenuity for life

Label Integer Presentation

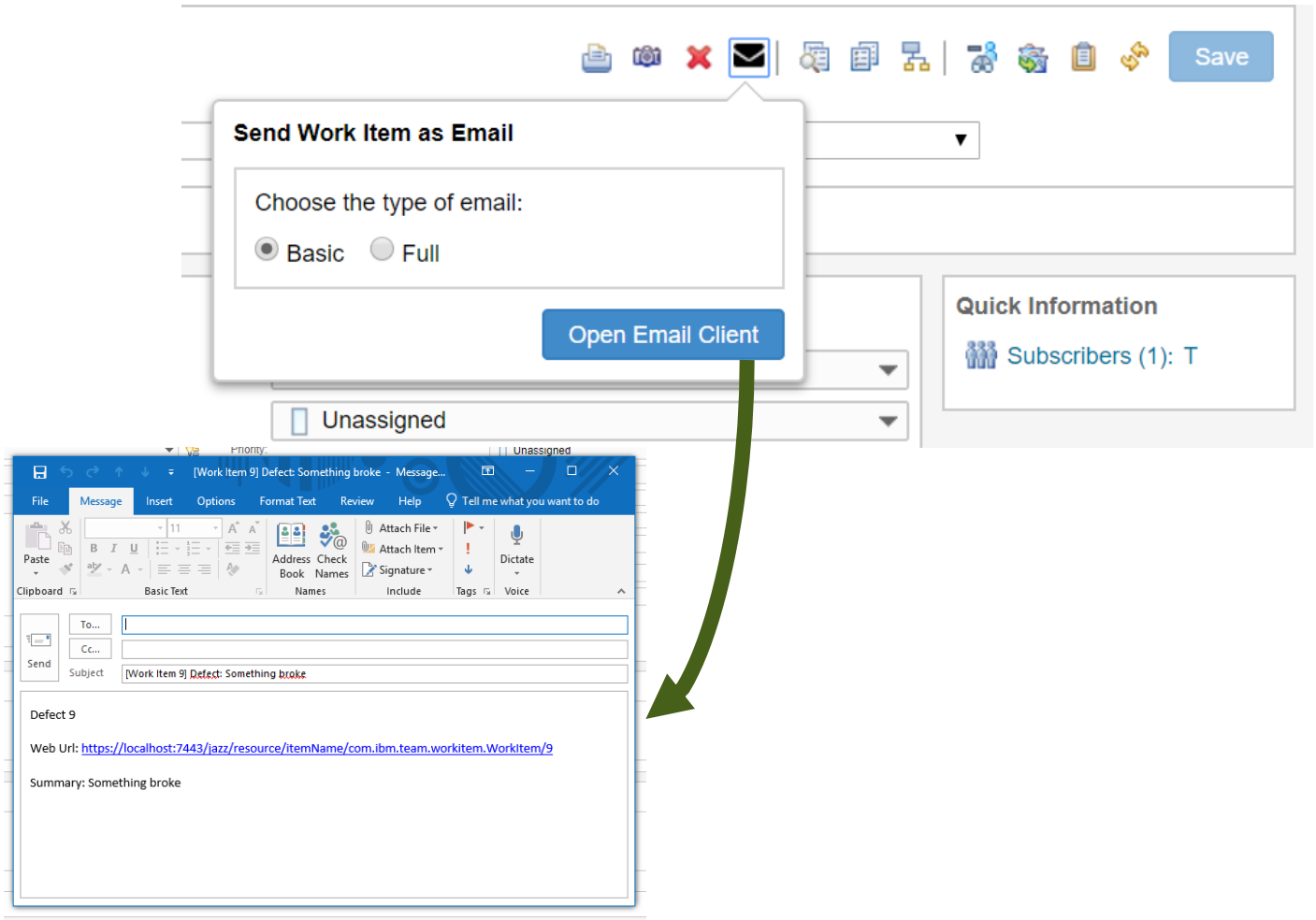


- Attribute based presentation
- Works with built in or custom attributes
- The integer value is right aligned
- The label on the right can be specified using presentation properties
- The label width can also be set in the presentation properties
- Displaying, saving, and validating the integer works like other integers in work items

- Custom attribute based presentations can only be configured with the Eclipse Client
- The presentation properties also can only be configured with the Eclipse Client

What We Will Create Today

Email Work Item Toolbar Action



- Work Item Editor Toolbar Action
- Available for all Work Item types
- Opens the local mail client (Outlook)
- Fills the subject and body of the email
- Has two modes: Basic and Full

- Basic just uses the summary and description
- Full takes all work item attributes that have a value

- The button will automatically appear when the plugin is included
- No configuration needed

Jazz Development Environment

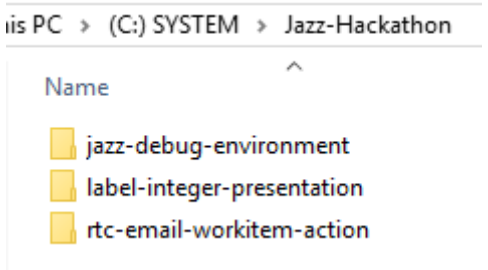
SIEMENS
Ingenuity for life

GitHub

- jazz-community/jazz-debug-environment
- Detailed Wiki on GitHub
- Compatible with v6.0.6 and v6.0.3
- Works with Windows and Linux
- Easy Setup
  - Get started with just three commands
  - Run in PowerShell
    - ./gradlew bootstrap
    - Copy SDK and Server files manually
    - ./gradlew setup
    - ./gradlew run
  - Access local Webserver:
    - https://localhost:7443/jazz

Hackathon Zip File

- Already setup to save time
- Needs an absolute path
- Extract to “C:\Jazz-Hackathon”



- Just run with ./gradlew run from within the “jazz-debug-environment” folder using PowerShell

- For Windows 10: Open PowerShell as Admin and set the execution policy
  - `Set-ExecutionPolicy -ExecutionPolicy RemoteSigned`

Get it Running on Your PC

SIEMENS
Ingenuity for life

Required

- Copy the Jazz-Hackathon zip file
- Extract to C:\Jazz-Hackathon
- Navigate to the jazz-debug-environment folder
- Hold Shift + Right Click
  - Choose: Open PowerShell window here
- Type in PowerShell: ./gradlew run
- Use a browser to go to: https://localhost:7443/jazz
- Login with TestJazzAdmin1 as User and Password
- Note: It's normal for there to be many errors in the PowerShell window

Optional

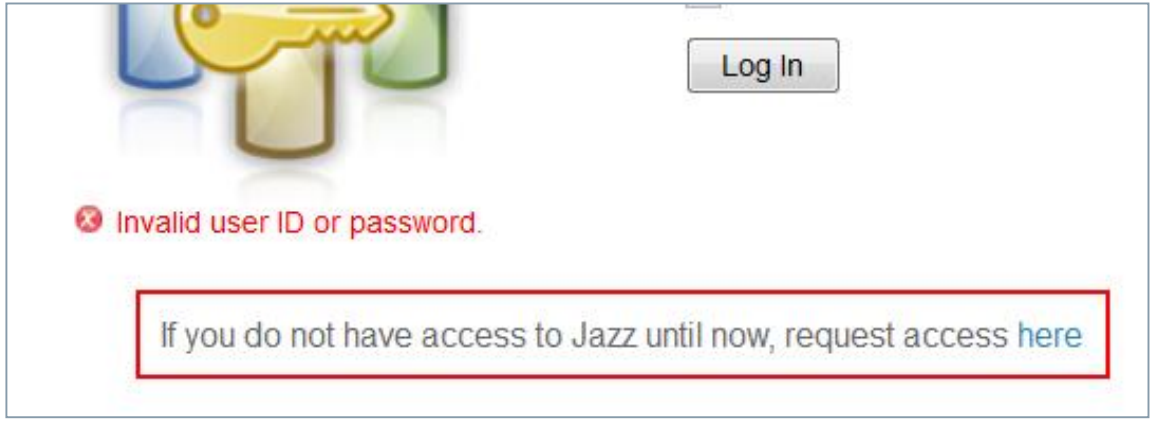
- Copy additional files if needed
  - Java-JDK – Required if you don't already have it. (Make sure to set the Java_Home environment variable)
  - VS Code – Open Source Code Editor
  - Chrome – Better Browser for Web Development
  - 7-Zip – The windows file manager has problems with some of the zip files
  - RTC-Eclipse-Client – In case you don't already have it

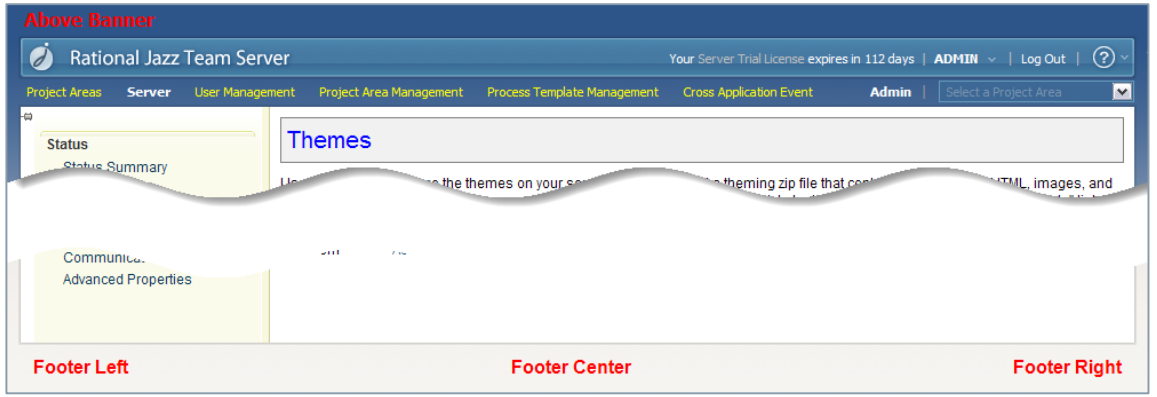
Focus Topic:
Client Side Extensions

Themes

SIEMENS
Ingenuity for life

- Goal: Extend Login Screen or page Header/Footer
- Good choice for:
  - Branding
  - User Guidance / Help
  - News and Announcements
- Bad choice for:
  - UI Hacks ☺
  - Any kind of expensive calls / operations
- Resources:
  https://jazz.net/wiki/bin/view/Main/WebUITheming
  https://jazz.net/forum/questions/245125/theme-with-custom-javascript

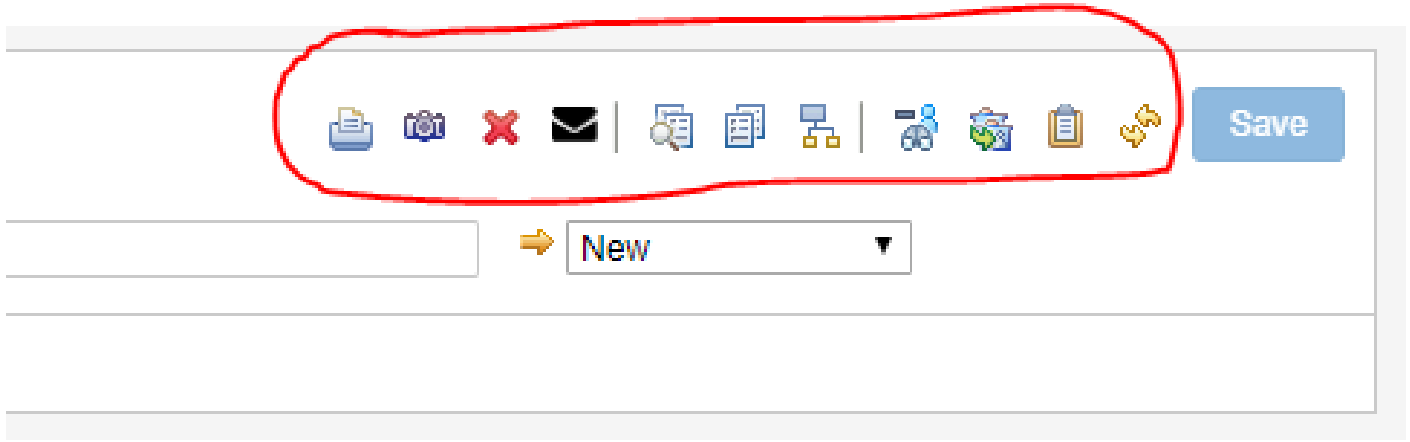




Extension Possibilities

SIEMENS
Ingenuity for life

Work Item Editor Toolbar Actions

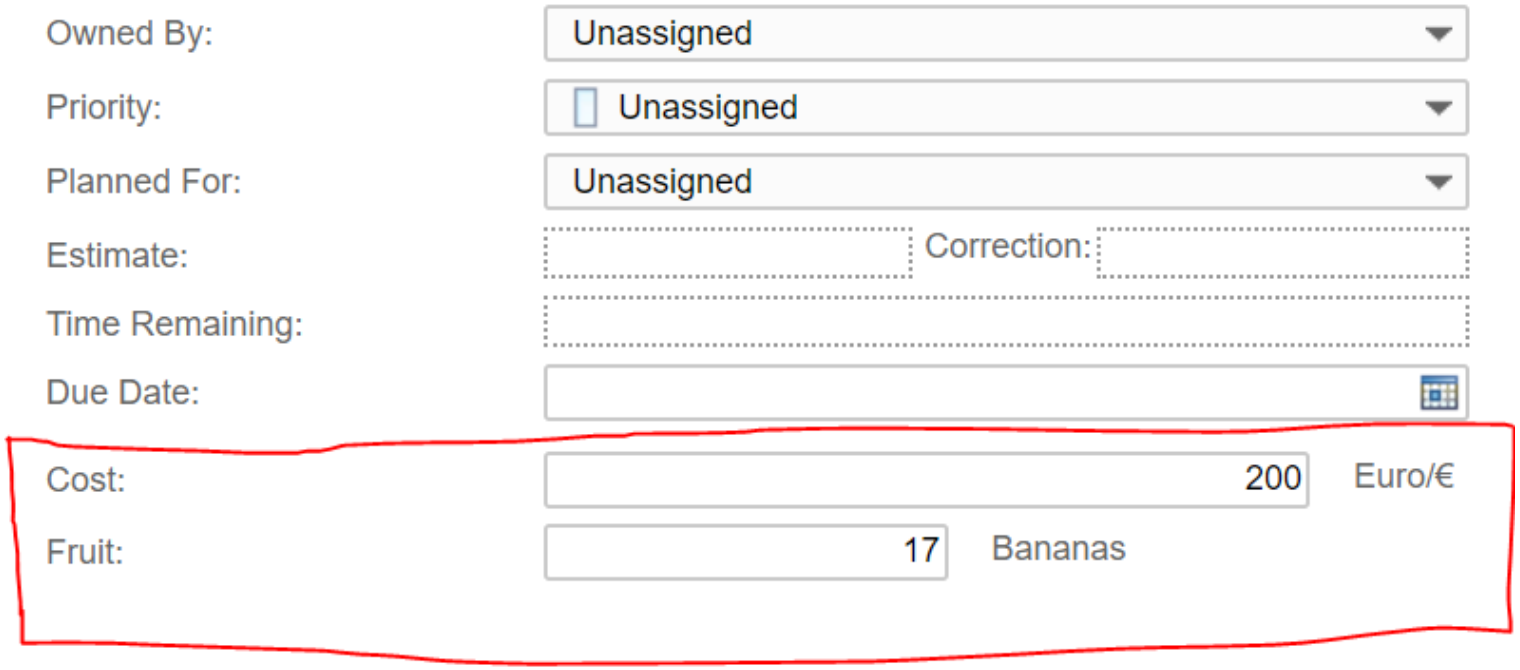


- Buttons that provide actions for all types of work items
- Small non intrusive button where the user expects it
- Can perform an action immediately or open a popup
- Has full access to the work item object
- Easy to extend with more buttons

Extension Possibilities

SIEMENS
Ingenuity for life

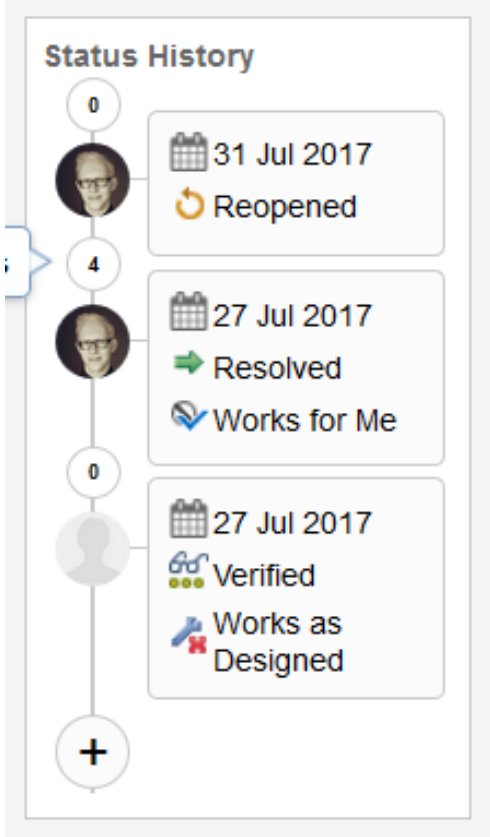
Attribute Based Presentations

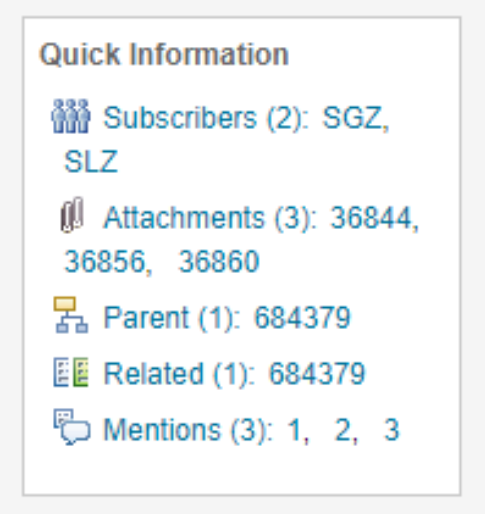


- Responsible for displaying and editing an attribute value
- Used for a specific type of attribute (e.g. integer attribute)
- Add functionality to an existing presentation
- Create a completely new presentation

Extension Possibilities

Non Attribute Based Presentations



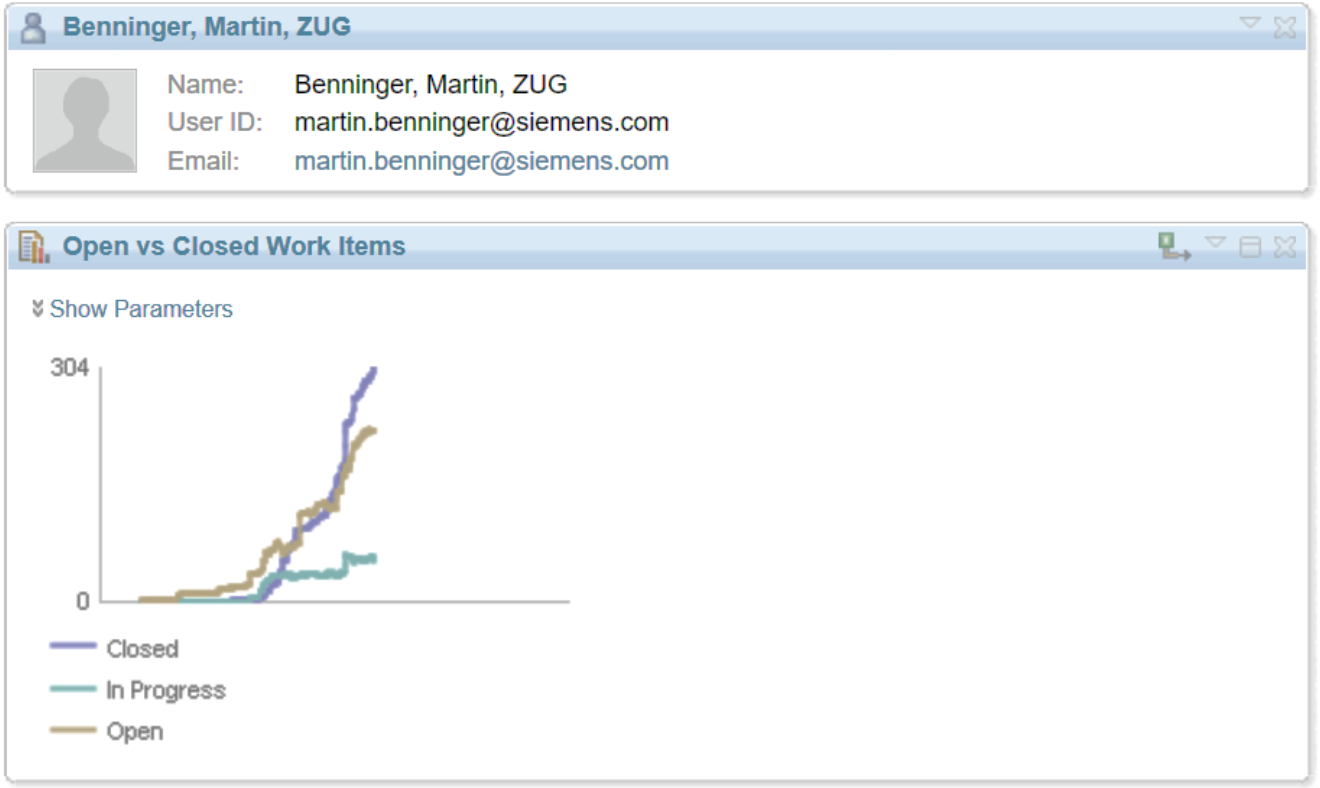


- Visible in the work item editor
- Present some information related to the work item but not to a specific attribute
- Present data in a clear way
- Provide additional functionality

Extension Possibilities

SIEMENS
Ingenuity for life

Dashboard Widgets



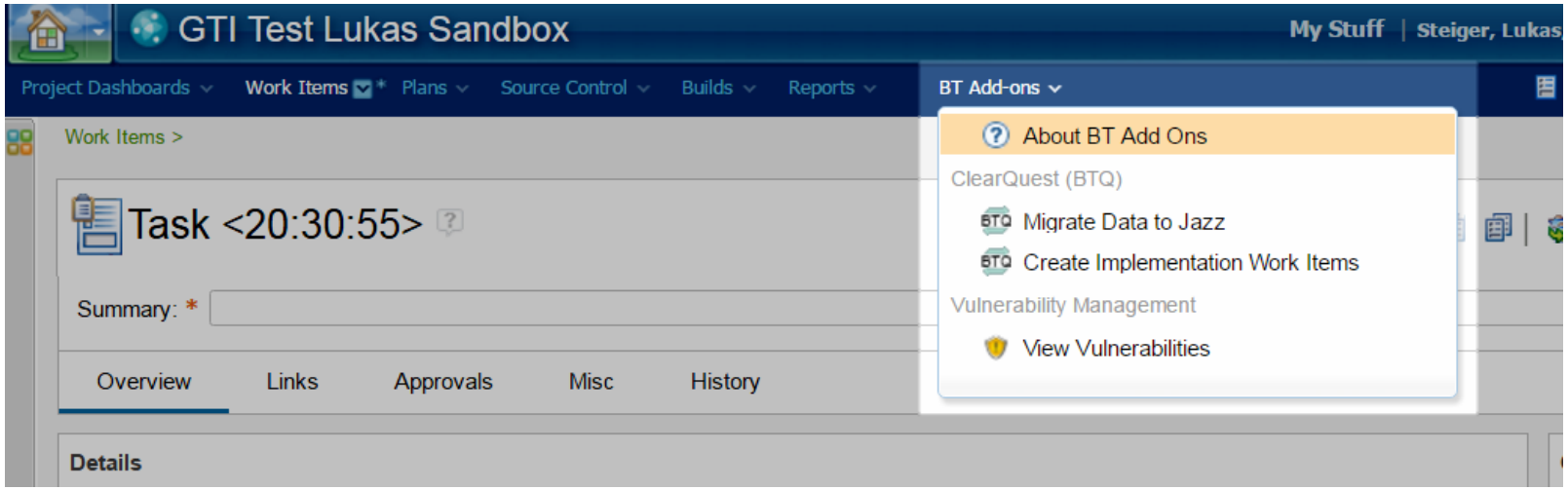
- Add to any dashboard
- Display all kinds of data
- Provide additional functionality
- Setting configurable from the dashboard

- More coming soon to the Jazz Community on GitHub

Extension Possibilities

SIEMENS
Ingenuity for life

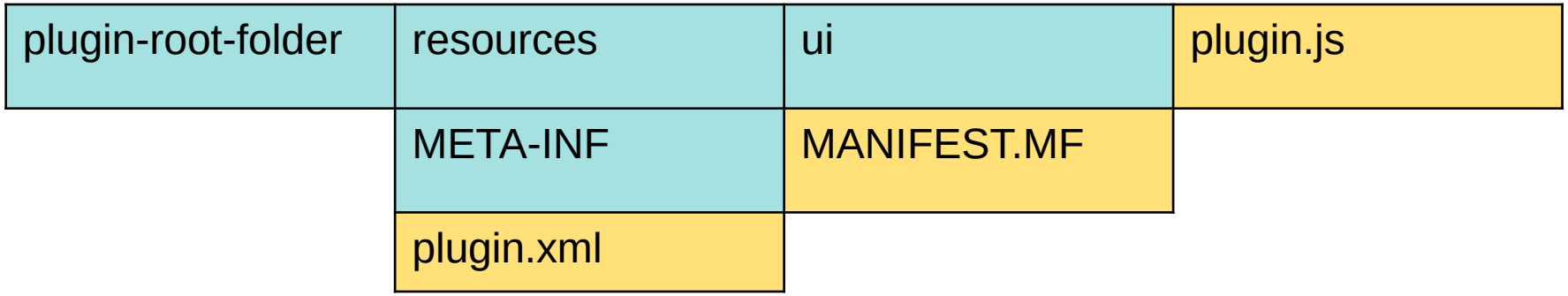
Page Extensions



- Add to the menu
- Full control of the page
- Good for larger full page actions

Plugin Structure

SIEMENS
Ingenuity for life



- plugin.xml → Contains the extension point and plugin id
- MANIFEST.MF → Contains describes the plugin (name, version, vendor, etc.)
- plugin.js → JavaScript file containing the dojo widget

- This basic structure is the same for all web plugins

Plugin Development Tips

SIEMENS
Ingenuity for life

Know what you're working with

- JavaScript, HTML, CSS
- Dojo version 1.8.4
- Jazz Ajax Framework
  - Minifying and bundling
- Jazz Source Code
  - Understanding it helps you write better code

Understand the plugin development process

- Starts with:
  - Researching, inspecting, reading source code, reverse engineering, looking at examples, understanding how it works

- Later becomes:
  - Using the provided tools (Dojo, internal Jazz methods, …)
  - Developing your functionality
  - Integrating functionality & look and feel with Jazz

- In the end:
  - Once you have it setup and are familiar with the quirks → It's front-end development

Extend Existing

SIEMENS
Ingenuity for life

Inheritance

- Dojo simulates class-based inheritance
- Inherit from IBM dojo widgets
- Possibilities with inheritance:
  - Provide additional functionality
  - Override existing functionality
  - Use the built in functionality
- Create a customized version of something that already exists

Pros

- Quickly add new features to existing functionality
- Don’t have to reimplement existing functionality
- Works the way IBM intended

Cons

- You don’t have full control
- Not ideal for larger changes

Debug Mode

SIEMENS
Ingenuity for life

https://localhost:7443/jazz/web/projects/Dev%20Project%20Area?debug=true#action=com.ibm.team.workitem.viewWorkItem&id=7

How

- Add ?debug=true to the URL
- At the end of the URL but be for the #action part

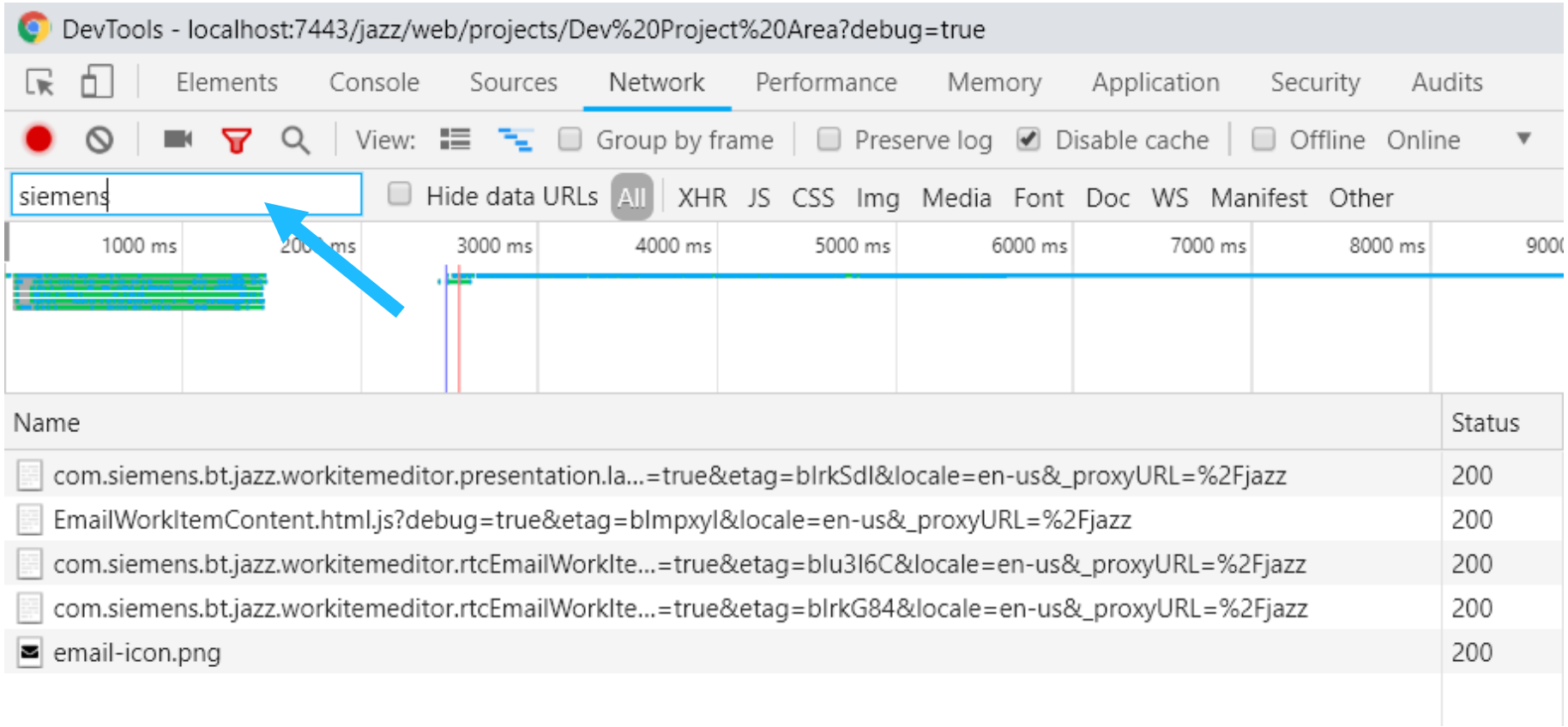
Pros

- Better JavaScript error logging
- JavaScript not bundled → Easier to inspect

Cons

- Loads very slowly
- Code is not run through the dojo parser on the server → Using this mode some things will work that will not work when you switch to the normal mode again (such as ES6 features). Be sure to test without debug mode.

Browser Developer Tools (F12)



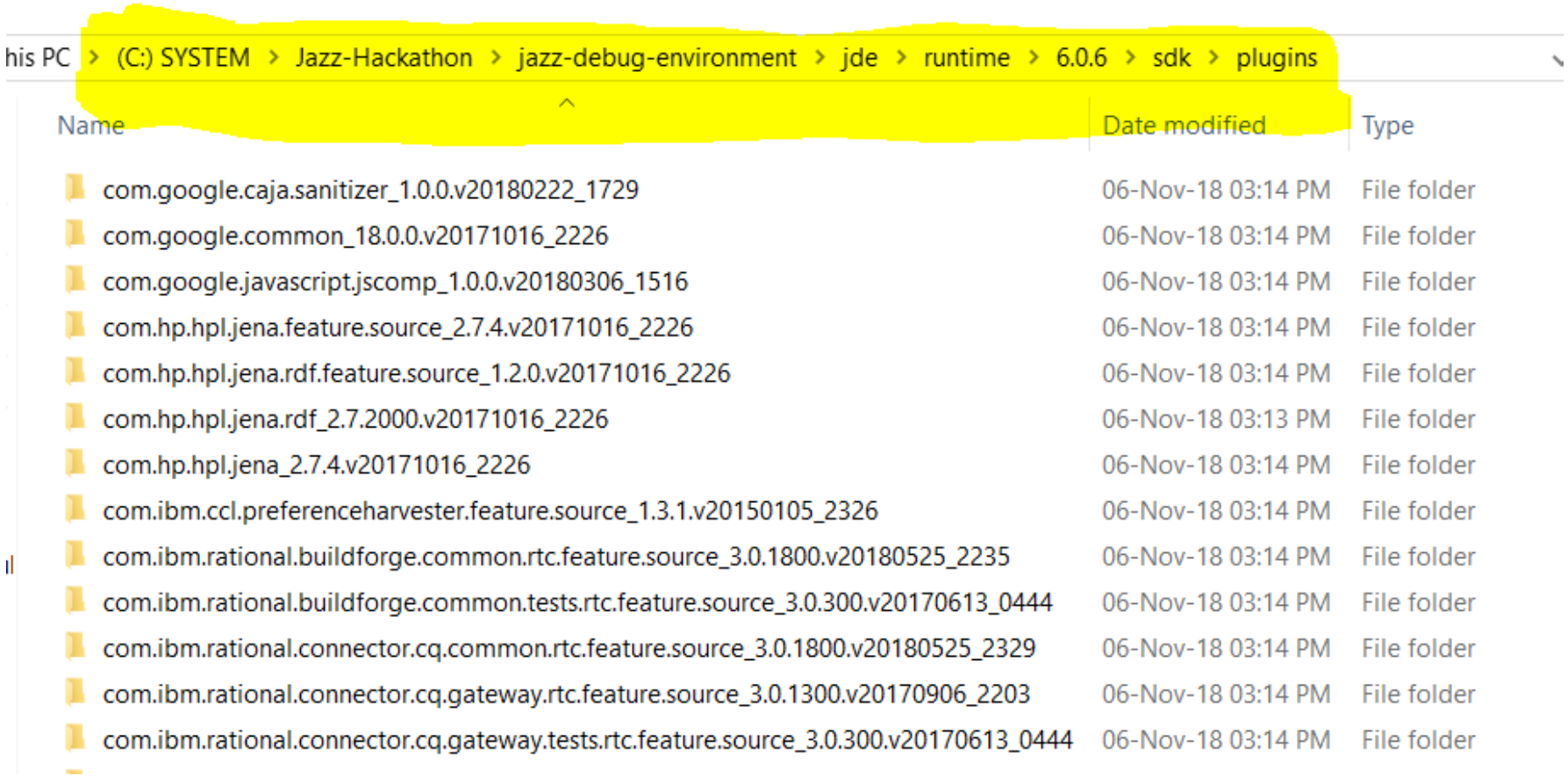
- Open with F12
- Use together with debug mode
- Filter for specific files
- Can view and download source files

JavaScript Source in SDK

SIEMENS
Ingenuity for life

How to find the files

- Some plugins are folders some are jar files (use 7-zip to extract)



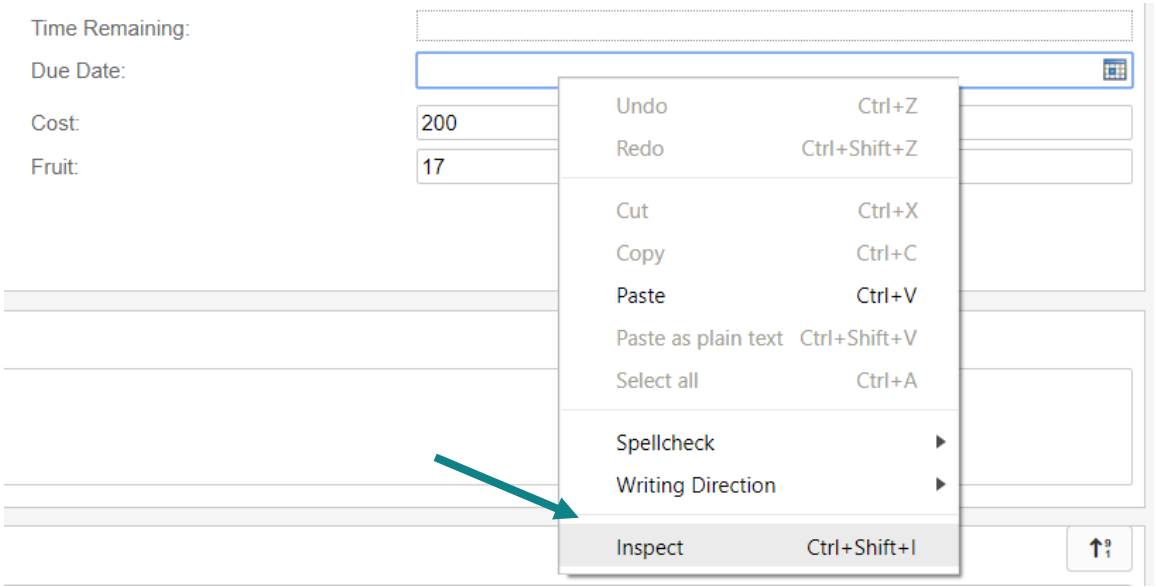
- Look through the plugins and find ones with web in the name
- Check if it contains JavaScript files
- Copy away for using as a reference

Inspect Elements

SIEMENS
Ingenuity for life

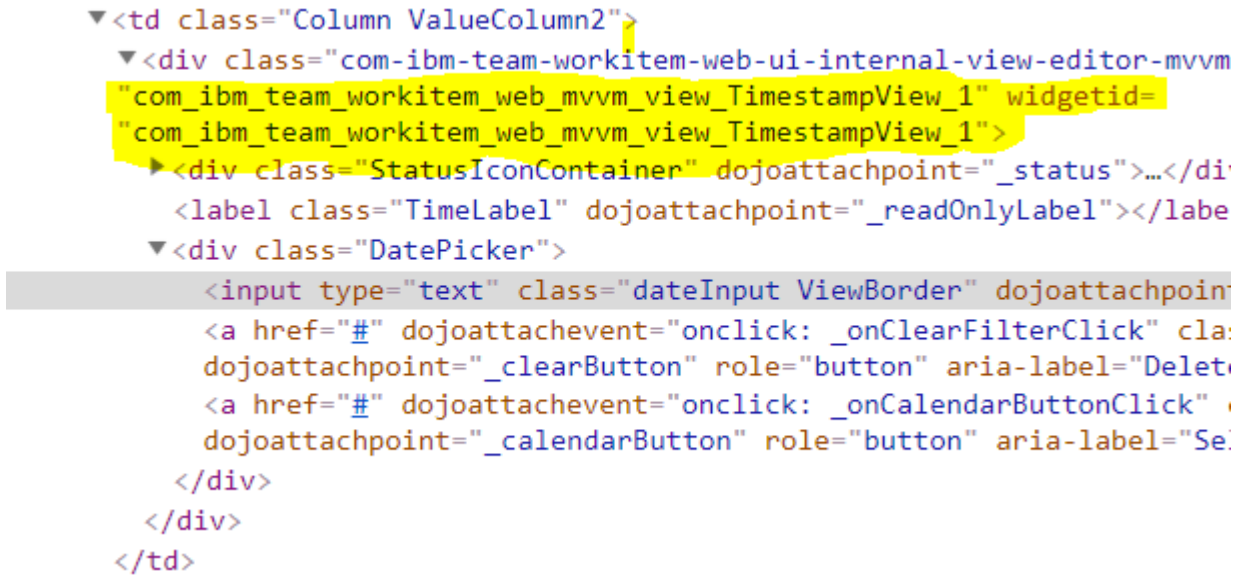
How to access

- Right click on an element in the browser
- Select the last item inspect

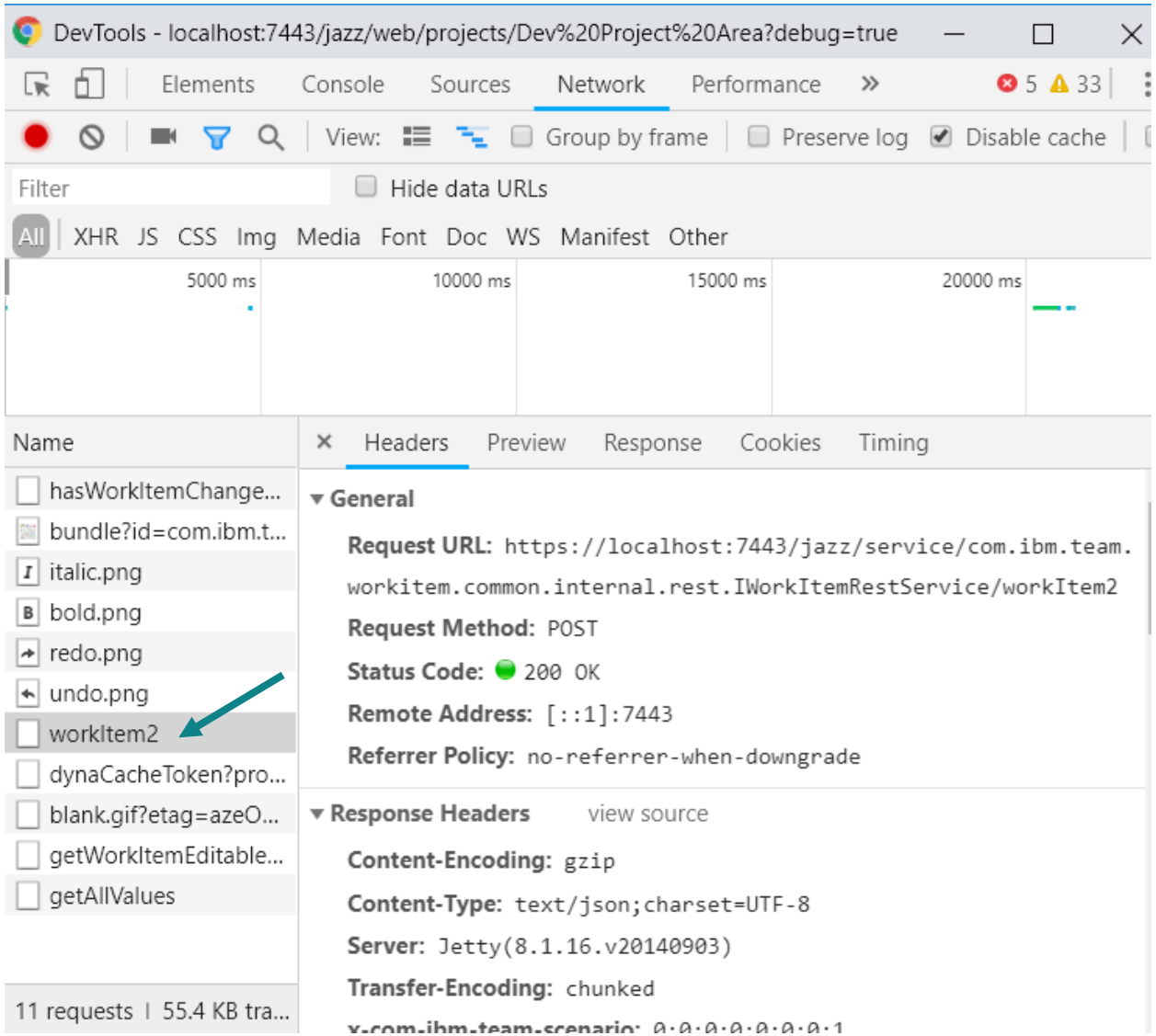


Find widget id

- Use to search in source



Find API Endpoints



- Open the DevTools (F12) and go to the Network tab
- Perform an action on the page (e.g. Save Work Item)
- Look for the request that sent the action to the server (workItem2)
- Click on the request to view the details
  - The Request URL is the endpoint that was used
  - View request headers and parameters to find out how to format the request
  - View the response headers and body to know what to expect as the result of the request

Debugging Own Code

SIEMENS
Ingenuity for life

Logging

- Using the console.log function

Pros

- Easy way to see the value of a variable
- Fine to do a lot of logging during development
- Doesn’t interrupt the program flow
- Not seen by the users (if you forget to remove it)

Cons

- The logged value is evaluated when you look at it, not when it was logged (for reference types). It could have changed since then.

Breakpoints

- Using the debugger; statement

Pros

- Works like a breakpoint
- Possible to view and change the value of all variables in the current scope
- Can continue with step by step debugging

Cons

- Breaks the program flow
- Gets annoying if there are too many
- Don’t use in a loop → Might crash your browser

Build with Webpack

SIEMENS
Ingenuity for life

NPM Package

- Package name: jazz-update-site-webpack-plugin
- Creates a zip file with the plugin in Update Site format
- Made by Lukas

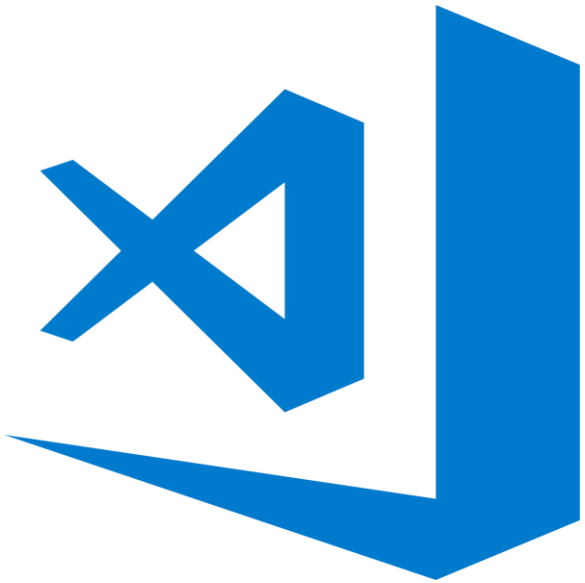
WebPack Config

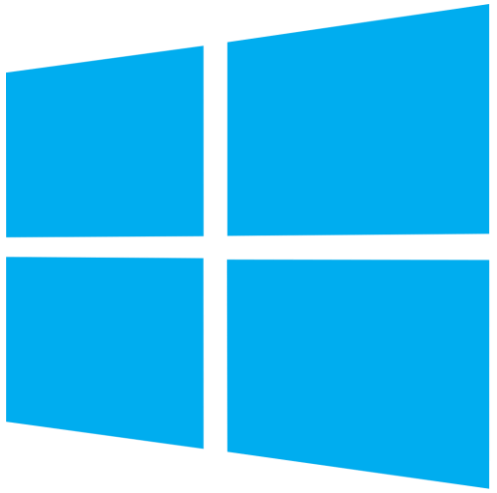
- Add to the plugins section of the webpack config

```
new JazzUpdateSitePlugin({
    appType: 'ccm',
    projectId: 'com.siemens.bt.jazz.workitemeditor.rtcGitConnector',
    acceptGlobPattern: [
        'resources/**',
        'META-INF/**',
        'plugin.xml',
    ],
    projectInfo: {
        author: packageJson.author,
        copyright: packageJson.author,
        description: packageJson.description,
        license: packageJson.license,
        version: version,
    },
})
```

My Preferred Tools

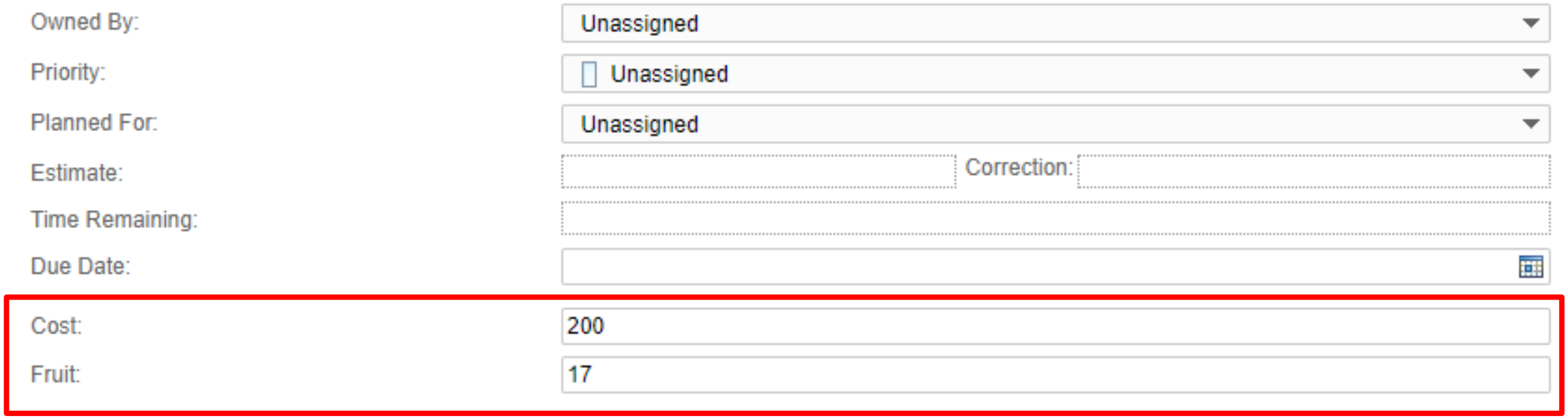
SIEMENS
Ingenuity for life



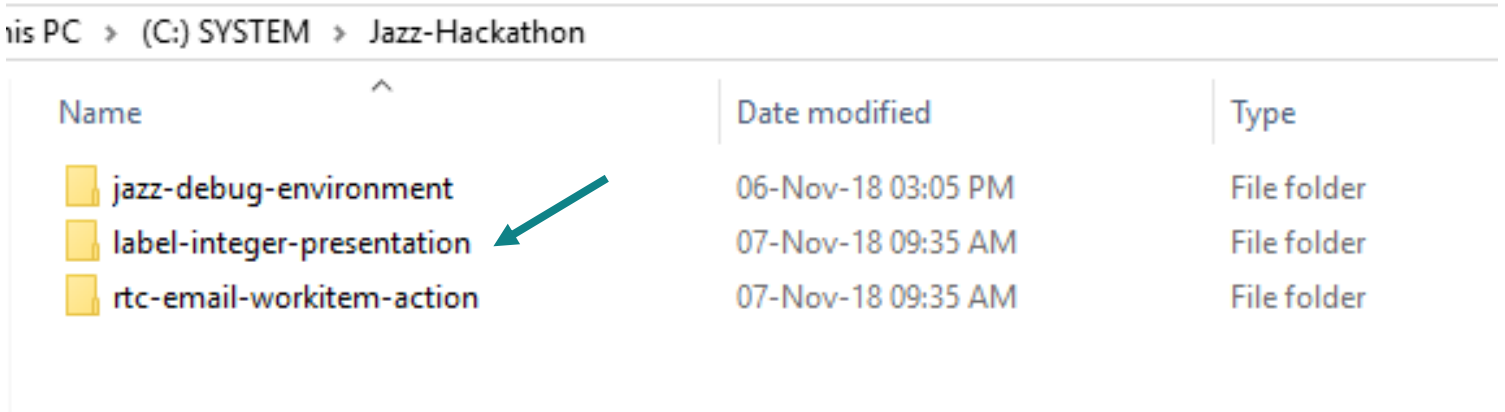


Label Integer Presentation – What We Have Now

SIEMENS
Ingenuity for life

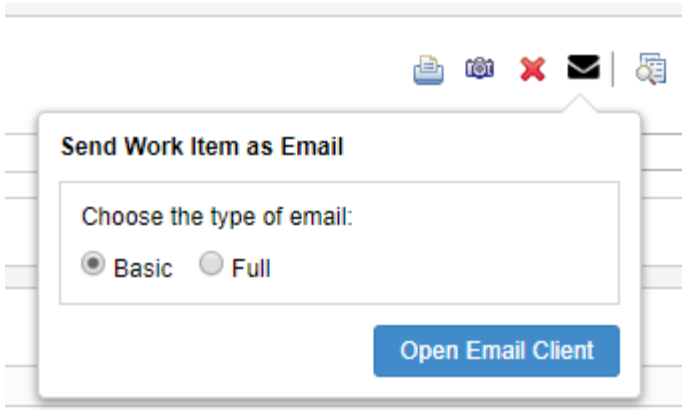


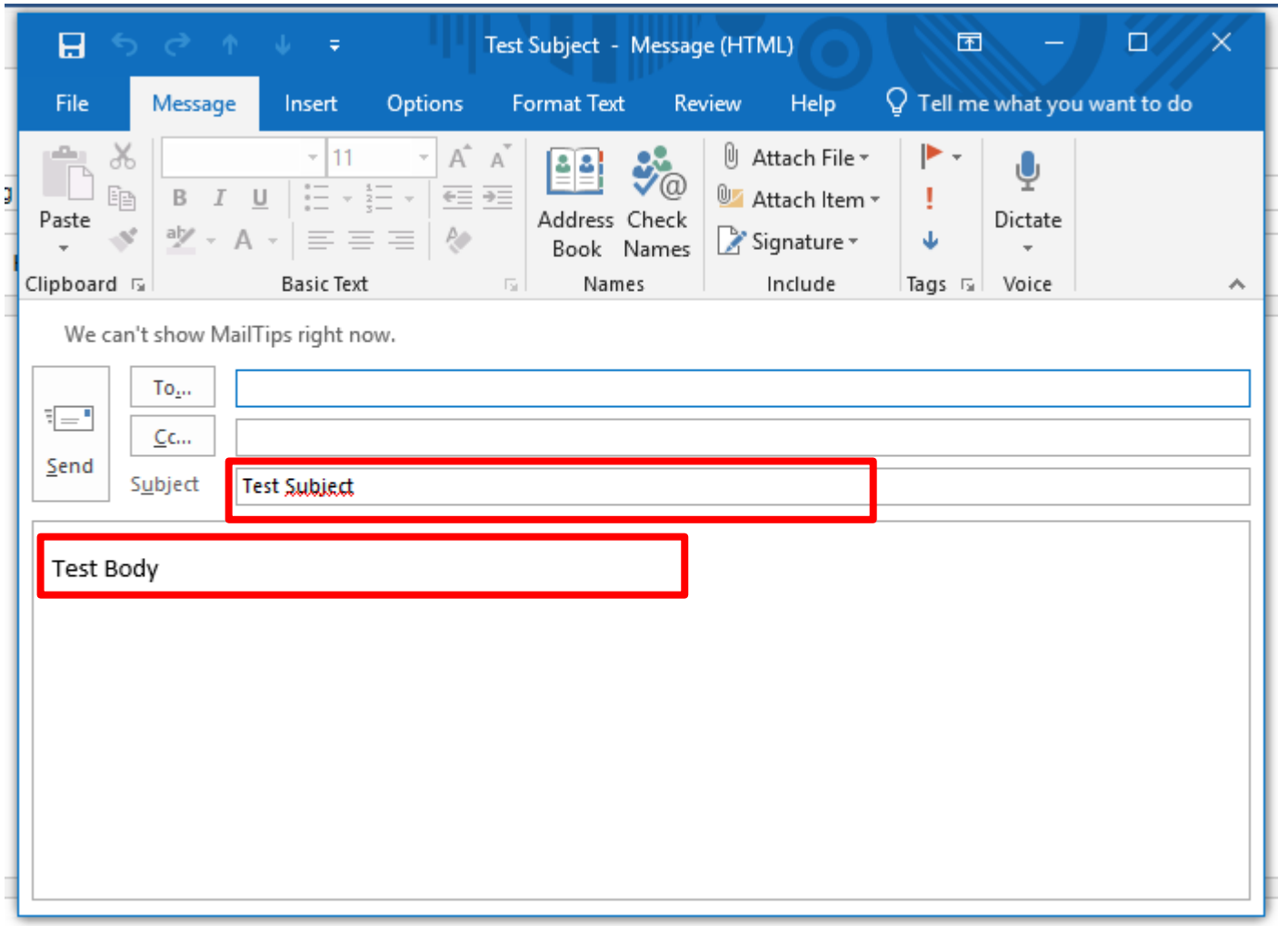
Label Integer Presentation - Code



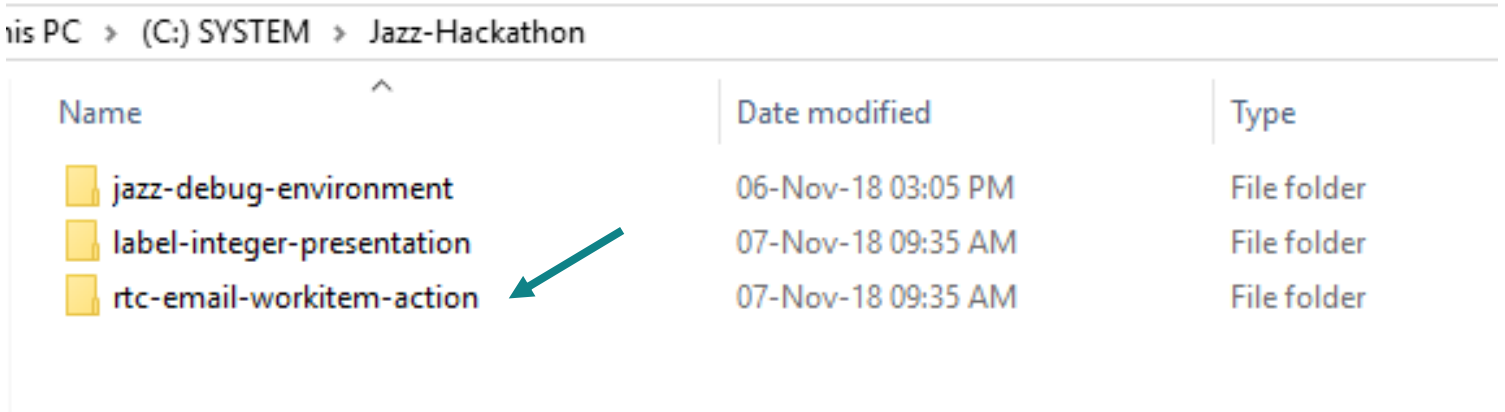
- Open with VS Code

Email Work Item Toolbar Action – What We Have Now



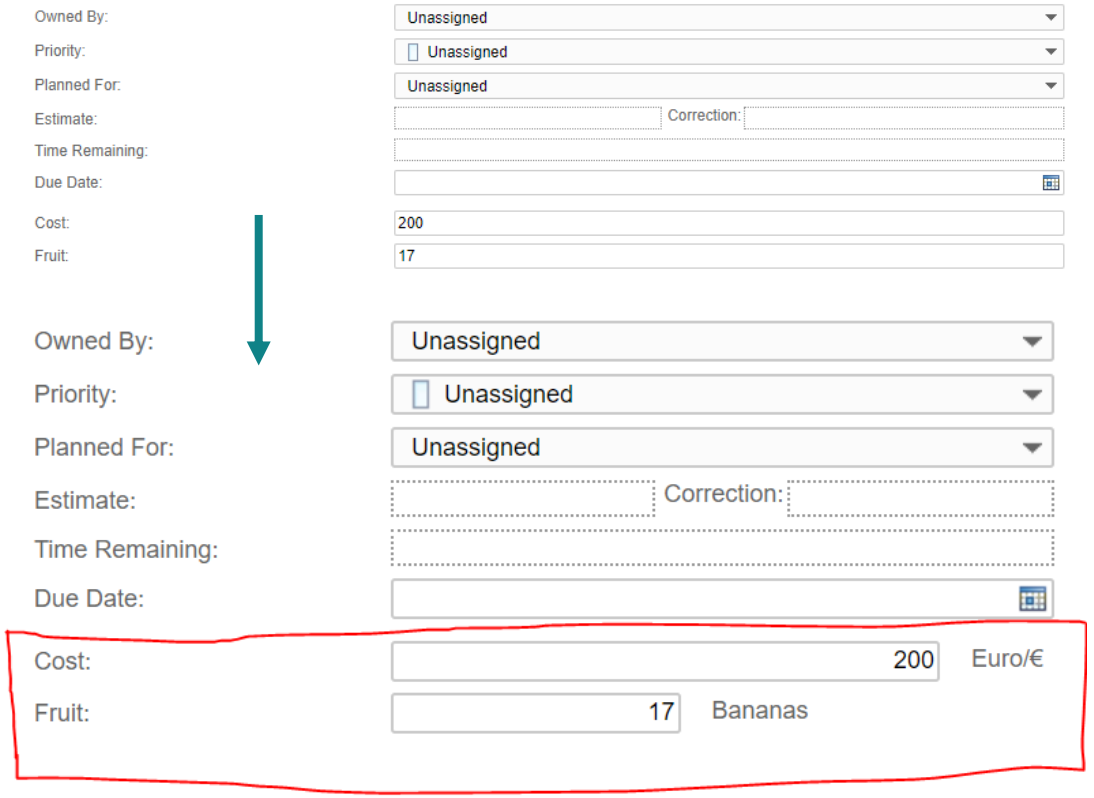


Email Work Item Toolbar Action - Code

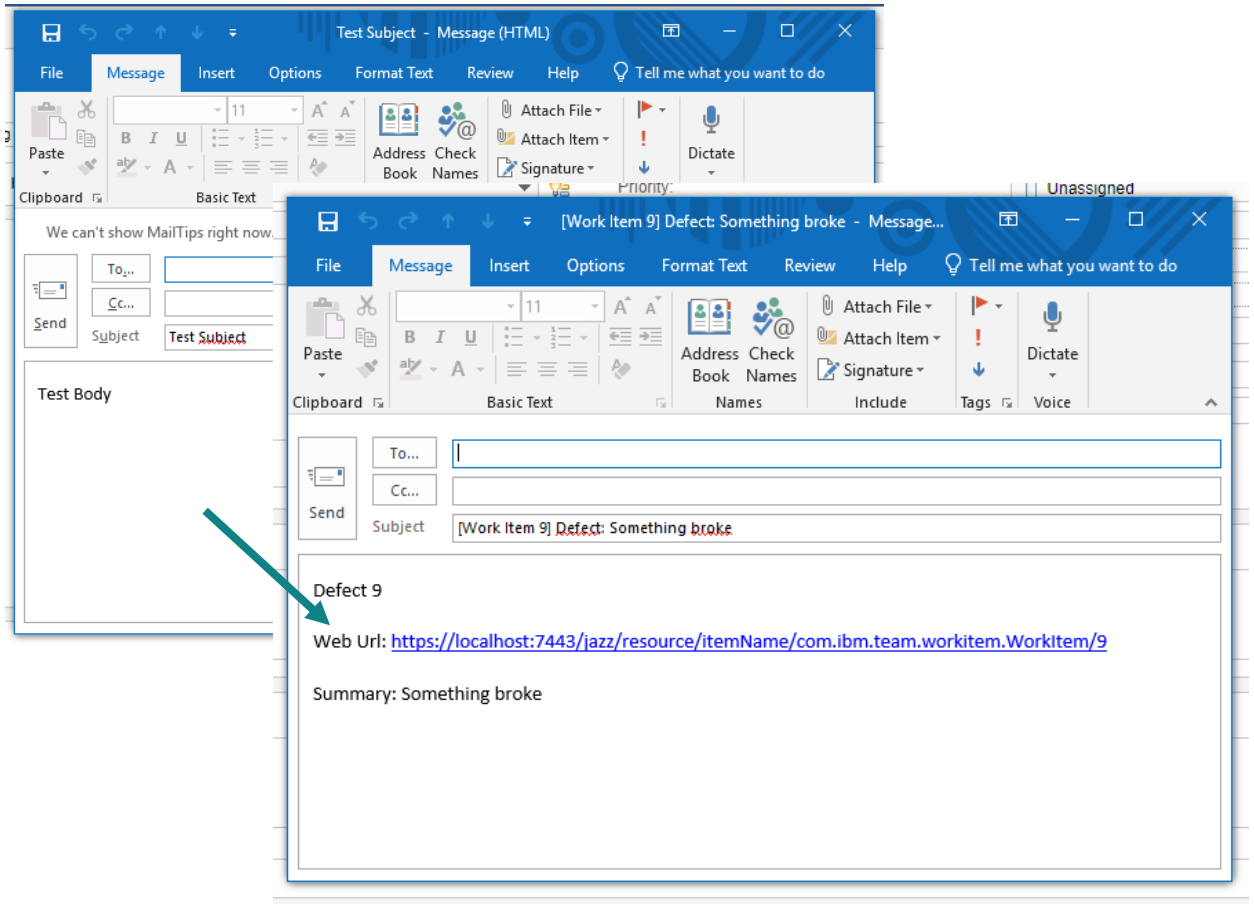


- Open with VS Code

Label Integer Presentation



Email Work Item Action



Deploy on Server

SIEMENS
Ingenuity for life

- Instructions on GitHub
- Can also find instructions from IBM and other sources

- Create zip file with npm and webpack

Installation

Deploy just like any other update site:

1. Extract the `com.siemens.bt.jazz.workitemeditor.rtcGitConnector_updatesite.ini` **file** from the zip file to the `server/conf/ccm/provision_profiles` directory
2. Extract the `com.siemens.bt.jazz.workitemeditor.rtcGitConnector_updatesite` **folder** to the `server/conf/ccm/sites` directory
3. Restart the server

Updating an existing installation

1. Request a server reset in **one** of the following ways:
   - If the server is currently running, call `https://server-address/ccm/admin/cmd/requestReset`
   - Navigate to `https://server-address/ccm/admin?internaltools=true` so you can see the internal tools (on the left in the side-pane). Click on `Server Reset` and press the `Request Server Reset` button
   - If your server is down, you can delete the ccm `built-on.txt` file. Liberty packed with 6.0.3 puts this file in a subfolder of `server/liberty/servers/clm/workarea/org.eclipse.osgi/**/ccm`. The easiest way to locate the file is by using your operating system's search capabilities.
2. Delete previously deployed updatesite folder
3. Follow the file extraction steps from the section above
4. Restart the server

package.json

SIEMENS
Ingenuity for life

- npm init
- npm i jazz-update-site-webpack-plugin --save-dev
- Run for all devDependencies
- npm run build

```
{
    "name": "com.siemens.bt.jazz.workitemeditor.rtcEmailWorkItemAction",
    "version": "1.0.0",
    "description": "Email Work Item with the Local Client",
    "main": "./resources/ui/EmailWorkItem.js",
    "devDependencies": {
        "jazz-update-site-webpack-plugin": "^0.4.1",
        "moment": "^2.22.2",
        "webpack": "^4.16.2",
        "webpack-cli": "^3.1.0"
    },
    "scripts": {
        "build": "node ./node_modules/webpack/bin/webpack.js"
    },
    "author": "Martin Benninger",
    "license": "MIT"
}
```

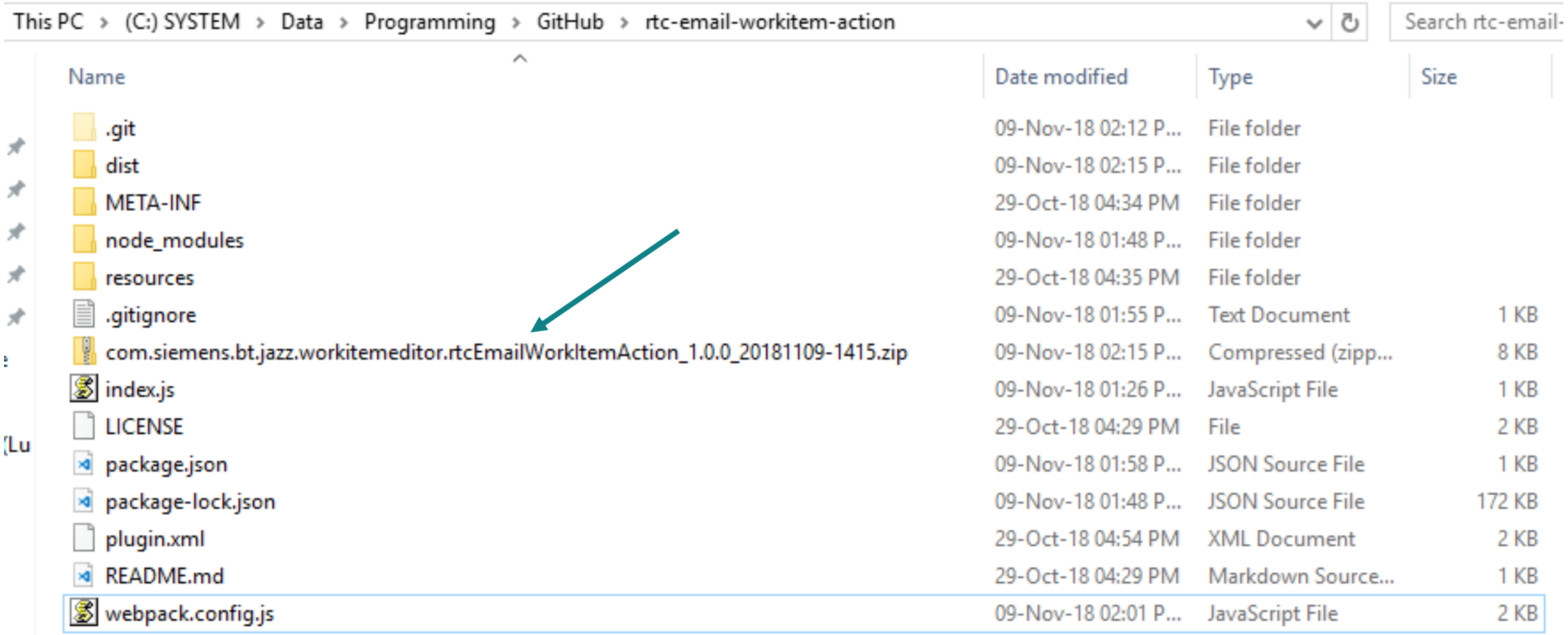
webpack.config.js

SIEMENS
Ingenuity for life

```
const JazzUpdateSitePlugin = require('jazz-update-site-webpack-plugin');
const moment = require('moment');
const packageJson = require('./package.json');

module.exports = (env) => {
    const timestamp = moment().format('[_]YYYYMMDD[-]HHmm');
    const version = (typeof env !== 'undefined' && (packageJson.version + "_" + env.buildUUID)) || packageJson.version + timestamp;
    const config = {
        entry: {
            app: './index.js',
        },

        plugins: [
            new JazzUpdateSitePlugin({
                appType: 'ccm',
                projectId: 'com.siemens.bt.jazz.workitemeditor.rtcEmailWorkItemAction',
                acceptGlobPattern: [
                    'resources/**',
                    'META-INF/**',
                    'plugin.xml',
                ],
                projectInfo: {
                    author: packageJson.author,
                    copyright: packageJson.author,
                    description: packageJson.description,
                    license: packageJson.license,
                    version: version,
                },
            })
        ]
    };

    return config;
};
```

Result zip file

SIEMENS
Ingenuity for life



Input:
Server Side Extensions

Extension Types and Points

SIEMENS
Ingenuity for life

Preconditions / Advisors

- Validate and Check information before the operation is executed
- e.g. check that the summary of a work item does not start with a lowercase letter and does not contain rich text.

Follow-up Actions / Participants

- Modify certain things after an operation was successful
- e.g. add the current owner of WI to the list of subscribers

Services

- Server side service that can be used by a client side plugin
- e.g. a custom REST service that can be called by a 3rd party app

Async Tasks

- Scheduled, async operations
- e.g. a Due Date Notifier (example see rsjazz.wordpress.com)

Resources:

- Leaning resources for RTC SDK based development: https://rsjazz.wordpress.com/2015/09/30/learning-to-fly-getting-started-with-the-rtc-java-apis/
- List of all Extension Points: https://jazz.net/wiki/bin/view/Main/CustomPreconditionsTable

Extending RTC – What and How

SIEMENS
Ingenuity for life

- Jazz Server
  - Client extensions:
    Tech.: JS / Dojo → OSLC and REST API
    → This is what you will learn coding today ☺
  - Server extensions:
    Tech.: Java → RTC SDK
    → learn here: https://github.com/jazz-community
- Eclipse client
  → As RTC is built on Eclipse, things are fairly similar here (see rsjazz for examples)
- External Apps
  - If they are written in Java, you can benefit from using the RTC SDK. Else, use the OSLC interface or the internal REST API's

Example 1: Role Cardinality Advisor – The goal

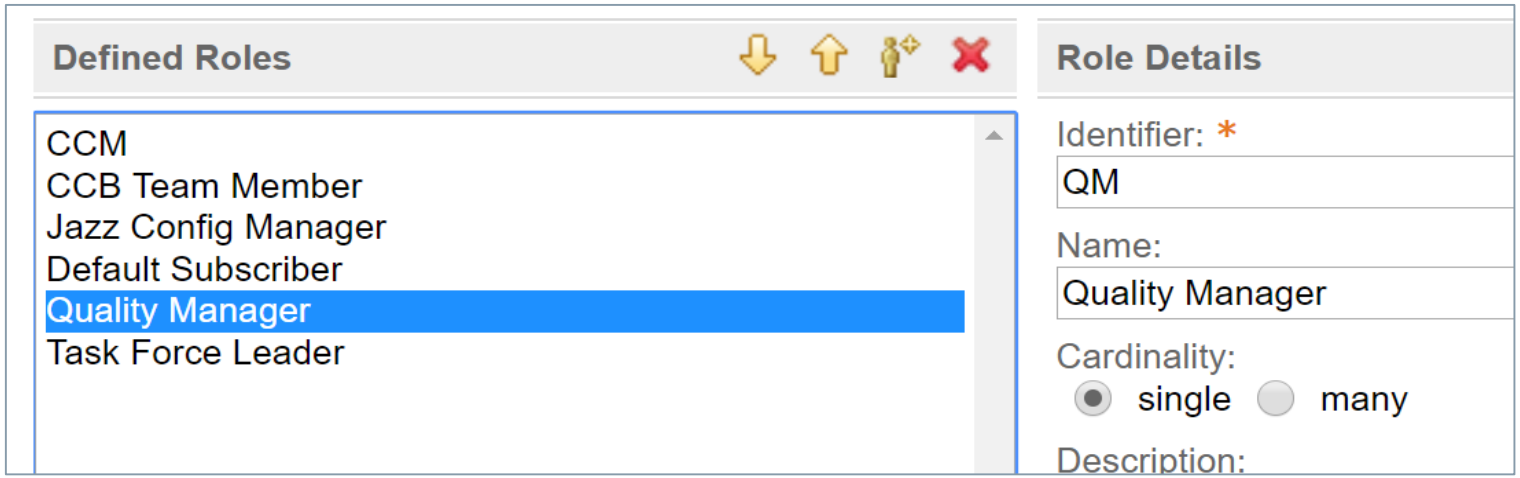
SIEMENS
Ingenuity for life

Main goal

- Simple example of how to write an Advisor / Precondition
- Show case that it's really not difficult to write a Jazz Advisor ☺

What we build

- Roles in Jazz have a cardinality: single or many
- From our understanding, this defines whether a role can be assigned at most once or to as many people as possible within a process area
- Problem: Jazz doesn't care about the cardinality



Example 1: Role Cardinality Advisor – The Advisor boilerplate

SIEMENS
Ingenuity for life

```
public class TeamAreaAdvisor extends AbstractService implements IOperationAdvisor {
    @Override
    public void run(AdvisableOperation operation,
                    IProcessConfigurationElement participantConfig,
                    IAdvisorInfoCollector collector, IProgressMonitor monitor) {
        if (!(operation instanceof TeamAreaSaveOperation)) return;
        TeamAreaSaveOperation data = (TeamAreaSaveOperation) operation;
        List<String> actions = Arrays.asList(data.getActions());
        if (!actions.contains(TeamAreaSaveOperation.ACTION_TEAM_AREA_MODIFY_MEMBERS)) return;
        IProcessItem[] newItems = data.getItems();
        for (IProcessItem newItem : newItems) {
            if (newItem instanceof ITeamArea) {
                IProcessServerService processService = getService(IProcessServerService.class);
                RoleCardinalityChecker.preventRoleCardinalityViolation((IProcessArea) newItem, processService);
            }
        }
    }
}
```

Example 1: Role Cardinality Advisor – Cardinality Check

SIEMENS
Ingenuity for life

```
public final class RoleCardinalityChecker {
    public static void preventRoleCardinalityViolation(IProcessArea pa, IProcessServerService processService) throws
TeamRepositoryException, InfoCollectorException {
        HashMap<IRole, List<IContributorHandle>> roleMap = new HashMap<>();
        // get all available roles from server
        IRole2[] roles = RoleHelpers.getRoleDefinitions(pa, processService);
        for(IRole2 role : roles) {
            roleMap.put(role, new ArrayList<IContributorHandle>());
        } // get list of users that will get this role after save
        for(IContributorHandle member : pa.getMembers()) {
            // member roles try to save
            IRole[] userRoles = pa.getRoleAssignments(member, roles);
            for(IRole role : userRoles) {
                List<IContributorHandle> list = roleMap.get(role);
                list.add(member);    }
        } // Check each role to ensure single cardinality
        for(IRole role : roleMap.keySet()) {
            String roleName = ((IRole2) role).getRoleName();
            if(role.getCardinality() == IRole2.CARDINALITY_SINGLE) {
                List<IContributorHandle> contributors = roleMap.get(role);
                if(contributors.size() > 1) {
                    throw new InfoCollectorException("Role cardinality violated.", "The role '" + roleName + "' cannot
be assigned to more than one user in team '" + pa.getName() + "'", InfoCollectorSeverity.ERROR);
                }  }  }  }  }
```

Example 2: WIBM (Work Item Bulk Mover) Service – The goal

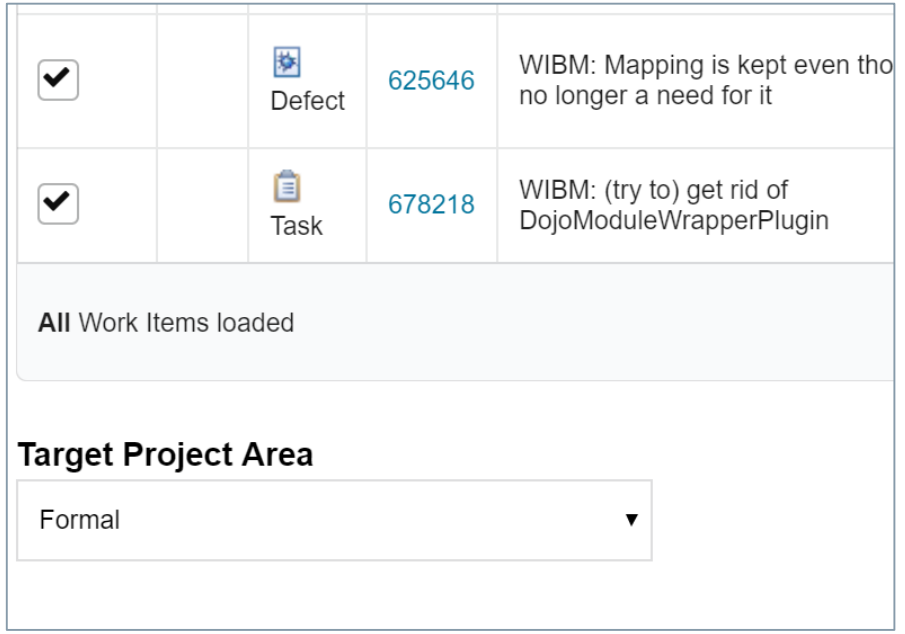
SIEMENS
Ingenuity for life

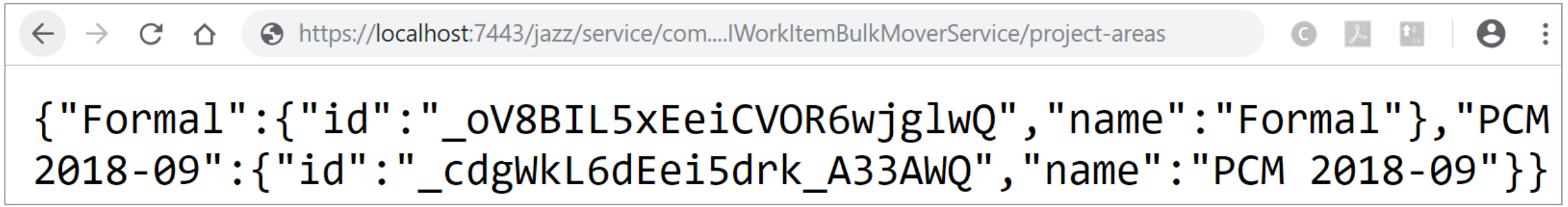
Main goal

- Use the single work item move from one pa to another for multiple work items – Expose it as a service!
- Show case that it's really not difficult to write a Jazz Service ☺

What we build

- Expose a service within a jazz application
- Different service endpoints for different cases





Example 2: WIBM – The Service Endpoints

SIEMENS
Ingenuity for life

```
public class WorkItemBulkMoverService extends BaseService implements IWorkItemBulkMoverService {
   public WorkItemBulkMoverService() {
      super();
      router.addService(HttpMethod.POST, "info", new RestFactory(InfoService.class));
      router.addService(HttpMethod.POST, "move", new RestFactory(MoveService.class));
      router.addService(HttpMethod.GET, "project-areas", new RestFactory(ProjectAreaService.class));
      router.addService(HttpMethod.GET, "types", new RestFactory(ProjectAreaTypeService.class));
   }
}
```

Example 2: WIBM – Get Users Project Areas

SIEMENS
Ingenuity for life

```
public class ProjectAreaService extends AbstractRestService {
    public ProjectAreaService(Log log, HttpServletRequest request, HttpServletResponse response, RestRequest
restRequest, TeamRawService parentService) {
        super(log, request, response, restRequest, parentService);
    }
    public void execute() {
        Gson googleJson = new Gson();
        try {
            IProcessServerService processServerService = parentService.getService(IProcessServerService.class);
            IContributorHandle contribHandle = processServerService.getAuthenticatedContributor();
            IRepositoryItemService itemService = parentService.getService(IRepositoryItemService.class);
            IContributor contributor = (IContributor) itemService.fetchItem(contribHandle, null);
            Map<String, ProjectArea> projectAreas = new TreeMap<String, ProjectArea>();
            IProcessArea[] areas = processServerService.findProcessAreas(contributor, null, null);
            for(IProcessArea a : areas) {
                IProjectArea pa = (IProjectArea) itemService.fetchItem(a.getProjectArea(), null);
                String paId = pa.getItemId().getUuidValue();
                projectAreas.put(pa.getName(), new ProjectArea(paId, pa.getName()));
            }
            String projectAreasJson = googleJson.toJson(projectAreas);
            response.getWriter().write(projectAreasJson);
        } catch (Exception e) {
            response.setStatus(500);
        }  }  }
```

Maven Archetype – Bootstrap Jazz Services in Minutes

https://github.com/jazz-community/jazz-plugin-maven-archetype

SIEMENS
Ingenuity for life

What are Jazz Services?

- Server side Java extension that do have full access to the RTC SDK capabilities
- Serves on https://*jazz-tld*/app/service/your-service
  your-service is the identifying name of the service
- Allows you to write more complex business logic

The challenge

Creation of our first service was quite difficult, due to the following reasons:

- Lacking documentation on how to do it
- Extremely challenging Build automation

Maven Archetype – What's that?

Maven is a build automation tool used primarily for Java projects. [.. It] addresses two aspects of building software: first, it describes how software is built, and second, it describes its dependencies.

Archetype: In short, Archetype is a Maven project templating toolkit. An archetype is defined as an original pattern or model from which all other things of the same kind are made.

Our solution

Our Maven archetype is a bootstrapper / generator to easily create Jazz based services

The plug-in can be used as well to bootstrap Participants and Advisors

p2 Repository Converter

https://github.com/jazz-community/jazz-p2-repository-converter

SIEMENS
Ingenuity for life

The Problem

- RTC SDK required to Build a Jazz Service
- Downloadable RTC SDK is a simple Zip, but Maven requires it to be in a specific format (called **p2**)

Our Workaround

- Conversion script to create a p2 package out of the RTC SDK
- Upload this package to our company internal artifact storage (Artifactory)
- Allows us to build Jazz Services within our company, but still prevents doing so online (e.g. using Travis), as we can't release it publicly without violating the IBM license

This is still valid! - IBM should provide the RTC SDK as a p2 repository on Maven Central

Final Thoughts

555 days of Jazz Community - Personal Highs and Lows

SIEMENS
Ingenuity for life

The Highs

- Articles written about us
- Mentioned on the official web site
- Very positive feedback of our users
- IBM Award for Guido & Markus
- Create Child extension part of core product (6.0.6)
- We have been copied (JazzApps)
- Increased satisfaction for us (work is more visible)
- Has lead to better design/architectural decisions
- Growing number of contributors

The lows

- Very few contributions
- Others not allowed or willing to open source

→ YOU can help us to get rid of the above lows ☺

Get in Touch

SIEMENS
Ingenuity for life

Martin Benninger
Junior Software Engineer

Siemens Schweiz AG, Building Technologies Division, International Headquarters
Building Technologies Division
Control Products & Systems
BT CPS R&D PI TI
Gubelstrasse 22
6300 Zug, Switzerland

E-Mail: martin.benninger@siemens.com
GitHub: https://github.com/MartinBenninger

Lukas Steiger
Junior Software Engineer

Siemens Schweiz AG, Building Technologies Division, International Headquarters
Building Technologies Division
Control Products & Systems
BT CPS R&D PI TI
Gubelstrasse 22
6300 Zug, Switzerland

E-Mail: lukas.steiger@siemens.com
GitHub: https://github.com/innerjoin
LinkedIn: https://linkedin.com/in/lukas-steiger/