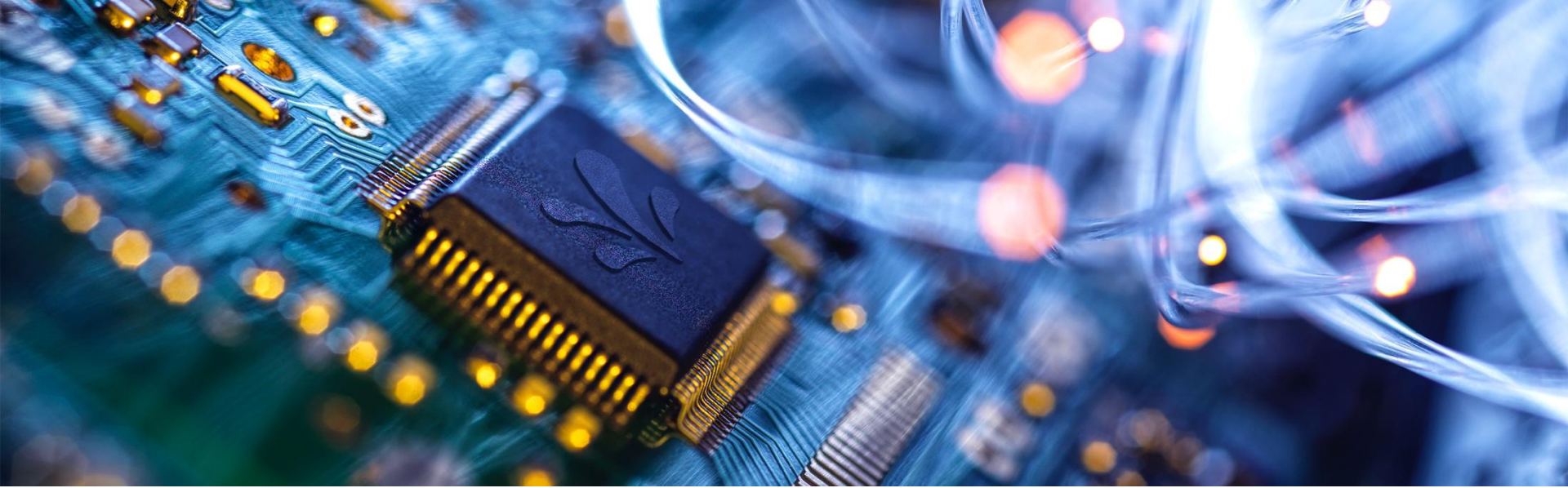
”Fed up with the tedious task of parsing through Gradle dependency logs?”

”Wish you could visualize Gradle dependencies instead of scrolling endlessly ?”

```
runtimeClasspath - Runtime classpath of source set 'main'.
+--- project :connector:standard-dam
|    \--- project :connector:base
|         +--- ch.qos.logback:logback-classic:1.3.5
|         |    +--- ch.qos.logback:logback-core:1.3.5
|         |    \--- org.slf4j:slf4j-api:2.0.4 -> 2.0.6
|         +--- ch.qos.logback:logback-core:1.3.5
|         +--- commons-io:commons-io:2.6 -> 2.11.0
|         +--- com.google.code.gson:gson:2.9.1
|         +--- javax.ws.rs:javax.ws.rs-api:2.0
|         +--- javax.xml.ws:jaxws-api:2.3.1
|         |    +--- javax.xml.bind:jaxb-api:2.3.1
|         |    |    \--- javax.activation:javax.activation-api:1.2.0
|         |    +--- javax.xml.soap:javax.xml.soap-api:1.4.0
|         |    \--- javax.annotation:javax.annotation-api:1.3.2
|         +--- javax.servlet:javax.servlet-api:4.0.1
|         +--- org.apache.commons:commons-collections4:4.3 -> 4.4
|         +--- commons-collections:commons-collections:3.2.2
|         +--- org.apache.commons:commons-lang3:3.5
|         +--- org.apache.poi:poi:5.2.3
|         |    +--- commons-codec:commons-codec:1.15
|         |    +--- org.apache.commons:commons-collections4:4.4
|         |    +--- org.apache.commons:commons-math3:3.6.1
|         |    +--- commons-io:commons-io:2.11.0
|         |    +--- com.zaxxer:SparseBitSet:1.2
|         |    \--- org.apache.logging.log4j:log4j-api:2.18.0
|         +--- org.apache.poi:poi-ooxml:5.2.3
|         |    +--- org.apache.poi:poi:5.2.3 (*)
|         |    +--- org.apache.poi:poi-ooxml-lite:5.2.3
|         |    |    \--- org.apache.xmlbeans:xmlbeans:5.1.1
|         |    |         \--- org.apache.logging.log4j:log4j-api:2.18.0
|         |    +--- org.apache.xmlbeans:xmlbeans:5.1.1 (*)
|         |    +--- org.apache.commons:commons-compress:1.21
|         |    +--- commons-io:commons-io:2.11.0
|         |    +--- com.github.virtuald:curvesapi:1.07
|         |    +--- org.apache.logging.log4j:log4j-api:2.18.0
|         |    \--- org.apache.commons:commons-collections4:4.4
|         +--- org.glassfish.jersey.containers:jersey-container-servlet:2.40
|         |    +--- org.glassfish.jersey.containers:jersey-container-servlet-core:2.40
|         |    |    +--- org.glassfish.hk2.external:jakarta.inject:2.6.1
|         |    |    +--- org.glassfish.jersey.core:jersey-common:2.40
|         |    |    |    +--- jakarta.ws.rs:jakarta.ws.rs-api:2.1.6
|         |    |    |    +--- jakarta.annotation:jakarta.annotation-api:1.3.5
|         |    |    |    +--- org.glassfish.hk2.external:jakarta.inject:2.6.1
|         |    |    |    \--- org.glassfish.hk2:osgi-resource-locator:1.0.3
|         |    |    +--- org.glassfish.jersey.core:jersey-server:2.40
|         |    |    |    +--- org.glassfish.jersey.core:jersey-common:2.40 (*)
|         |    |    |    +--- org.glassfish.jersey.core:jersey-client:2.40
|         |    |    |    |    +--- jakarta.ws.rs:jakarta.ws.rs-api:2.1.6
|         |    |    |    |    +--- org.glassfish.jersey.core:jersey-common:2.40 (*)
|         |    |    |    |    \--- org.glassfish.hk2.external:jakarta.inject:2.6.1
|         |    |    |    +--- jakarta.ws.rs:jakarta.ws.rs-api:2.1.6
|         |    |    |    +--- jakarta.annotation:jakarta.annotation-api:1.3.5
|         |    |    |    +--- org.glassfish.hk2.external:jakarta.inject:2.6.1
|         |    |    |    \--- jakarta.validation:jakarta.validation-api:2.0.2
|         |    |    \--- jakarta.ws.rs:jakarta.ws.rs-api:2.1.6
|         |    +--- org.glassfish.jersey.core:jersey-common:2.40 (*)
|         |    +--- org.glassfish.jersey.core:jersey-server:2.40 (*)
|         |    \--- jakarta.ws.rs:jakarta.ws.rs-api:2.1.6
|         +--- org.glassfish.jersey.core:jersey-server:2.40 (*)
|         +--- org.glassfish.jersey.ext:jersey-spring4:2.40
|         |    +--- org.glassfish.jersey.core:jersey-server:2.40 (*)
|         |    +--- org.glassfish.jersey.inject:jersey-hk2:2.40
|         |    |    +--- org.glassfish.jersey.core:jersey-common:2.40 (*)
|         |    |    +--- org.glassfish.hk2:hk2-locator:2.6.1
|         |    |    |    +--- org.glassfish.hk2.external:jakarta.inject:2.6.1
|         |    |    |    +--- org.glassfish.hk2.external:aopalliance-repackaged:2.6.1
|         |    |    |    +--- org.glassfish.hk2:hk2-api:2.6.1
|         |    |    |    |    +--- org.glassfish.hk2.external:jakarta.inject:2.6.1
|         |    |    |    |    +--- org.glassfish.hk2:hk2-utils:2.6.1
|         |    |    |    |    |    +--- jakarta.annotation:jakarta.annotation-api:1.3.4 -> 1.3.5
|         |    |    |    |    |    \--- org.glassfish.hk2.external:jakarta.inject:2.6.1
|         |    |    |    |    \--- org.glassfish.hk2.external:aopalliance-repackaged:2.6.1
|         |    |    |    +--- org.glassfish.hk2:hk2-utils:2.6.1 (*)
|         |    |    |    +--- jakarta.annotation:jakarta.annotation-api:1.3.4 -> 1.3.5
|         |    |    |    \--- org.javassist:javassist:3.22.0-CR2 -> 3.29.2-GA
|         |    |    \--- org.javassist:javassist:3.29.2-GA
|         |    +--- org.glassfish.jersey.containers:jersey-container-servlet-core:2.40 (*)
|         |    +--- org.glassfish.hk2:hk2:2.6.1
|         |    |    +--- org.glassfish.hk2:hk2-utils:2.6.1 (*)
|         |    |    +--- org.glassfish.hk2:hk2-api:2.6.1 (*)
|         |    |    +--- org.glassfish.hk2:hk2-core:2.6.1
```



Gradle dependency Visualizer

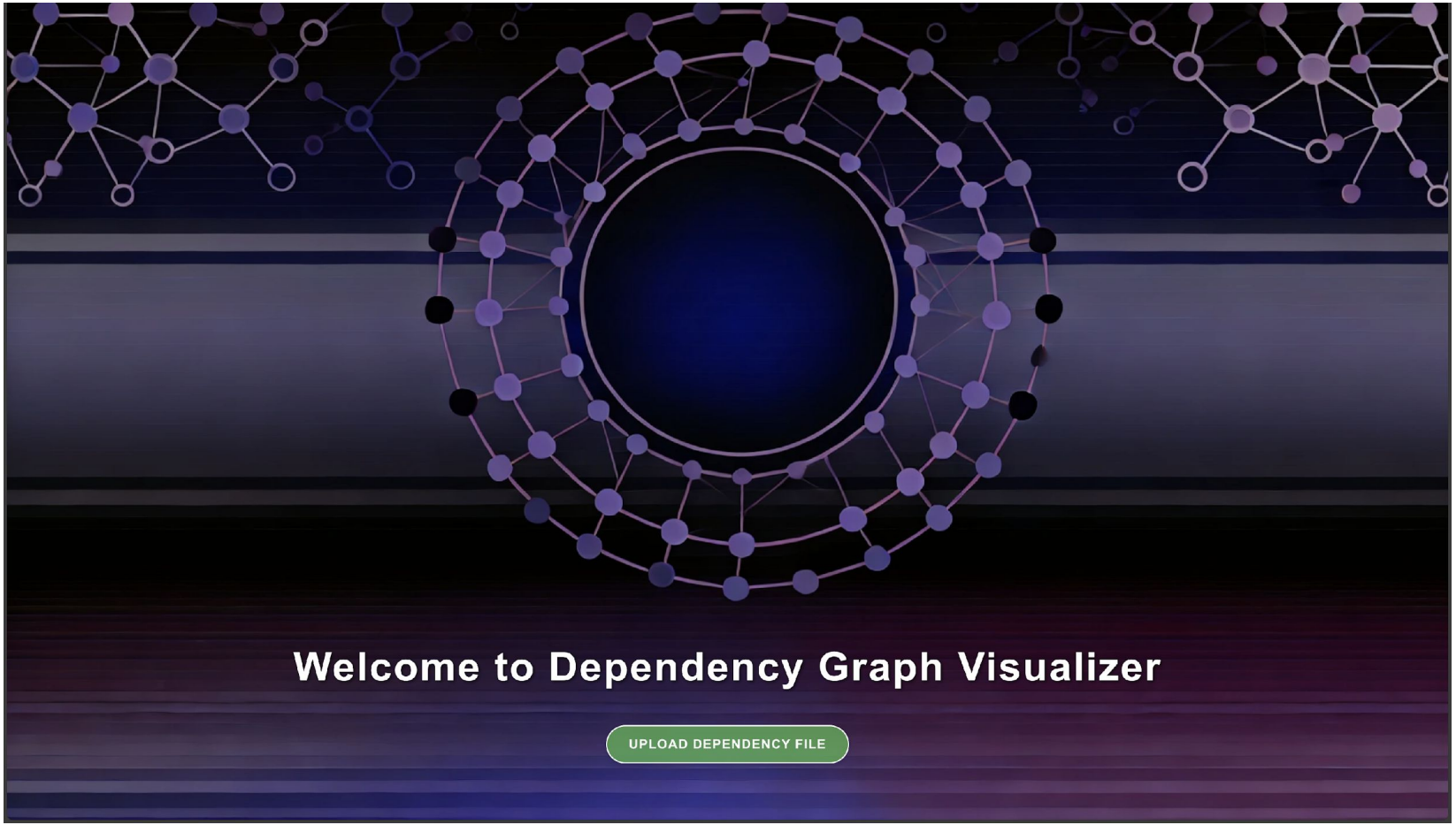
Arman Makhani | *Mentor* : Ashutosh Moudgil | *Manager* : Prateek Agrawal

sprinklr

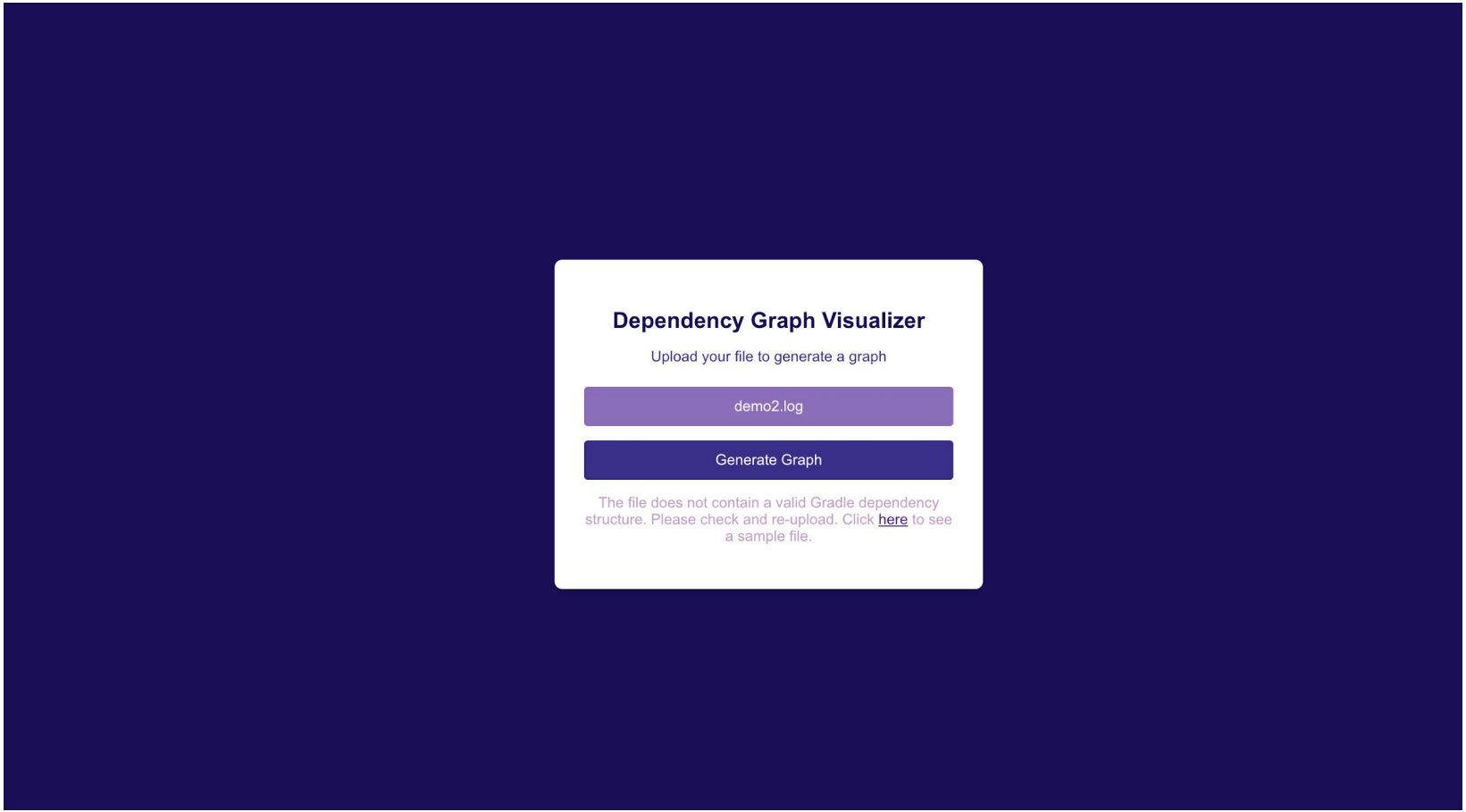
Objectives

1. Visualize the dependency log
2. Detects the wrong format
3. Shows circular dependencies
4. Search a particular dependency
5. Know the direct transitive dependency
6. Manage multimodular projects

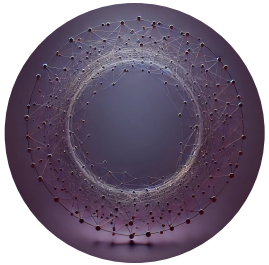
Welcome Page



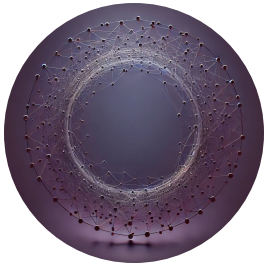
Upload page



Upload page features



Features



Only accepts .log or .txt files

Only accepts file with correct structure

Dependency Graph Visualizer

Upload your file to generate a graph

fault2.log

Generate Graph

The file structure is not in the correct Gradle dependency format or is incomplete. Please check and re-upload.

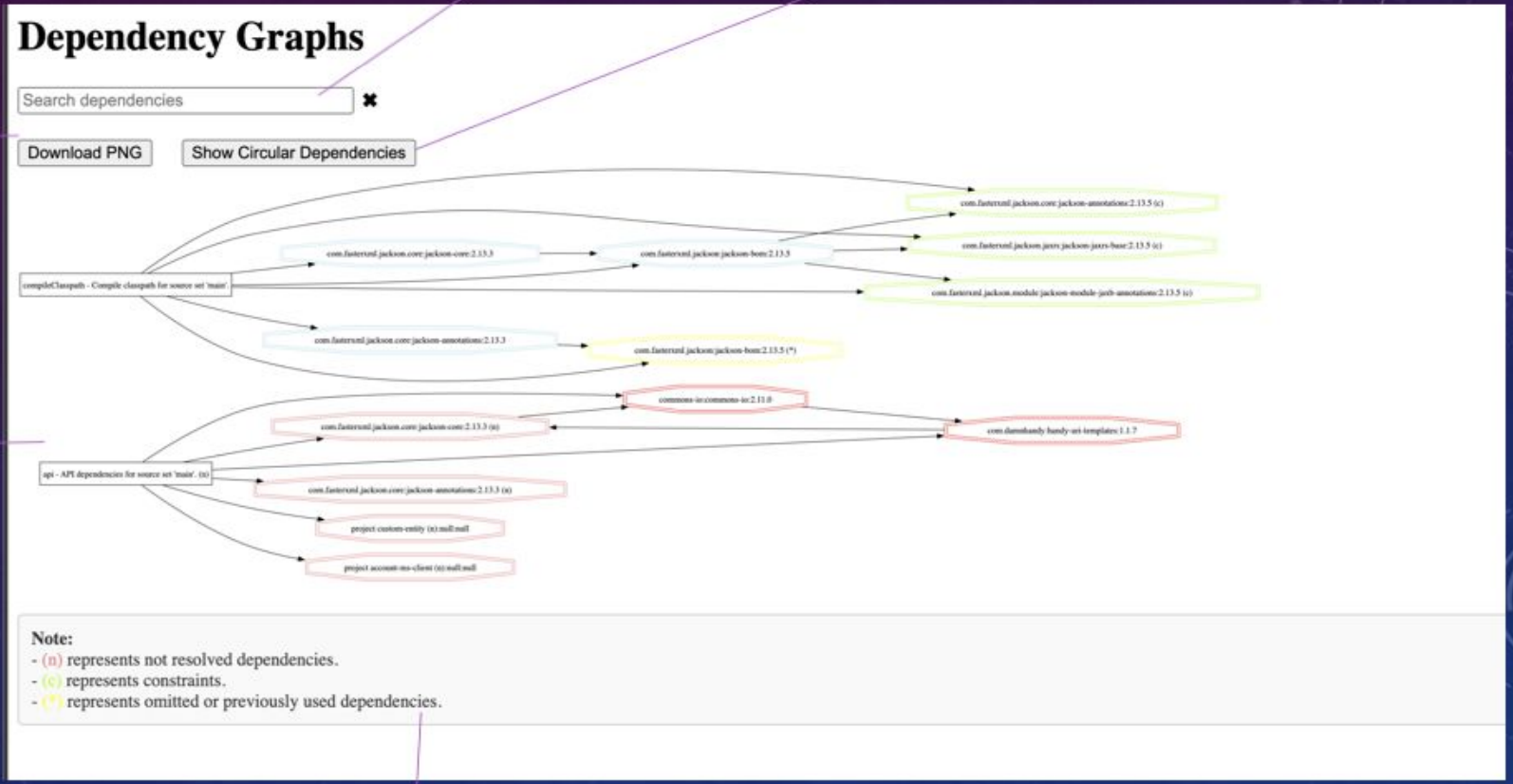
THE VIEW PAGE

Search bar to search
For something specific

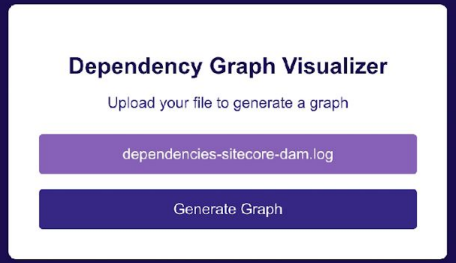
Makes the edges
involved in cycle blink

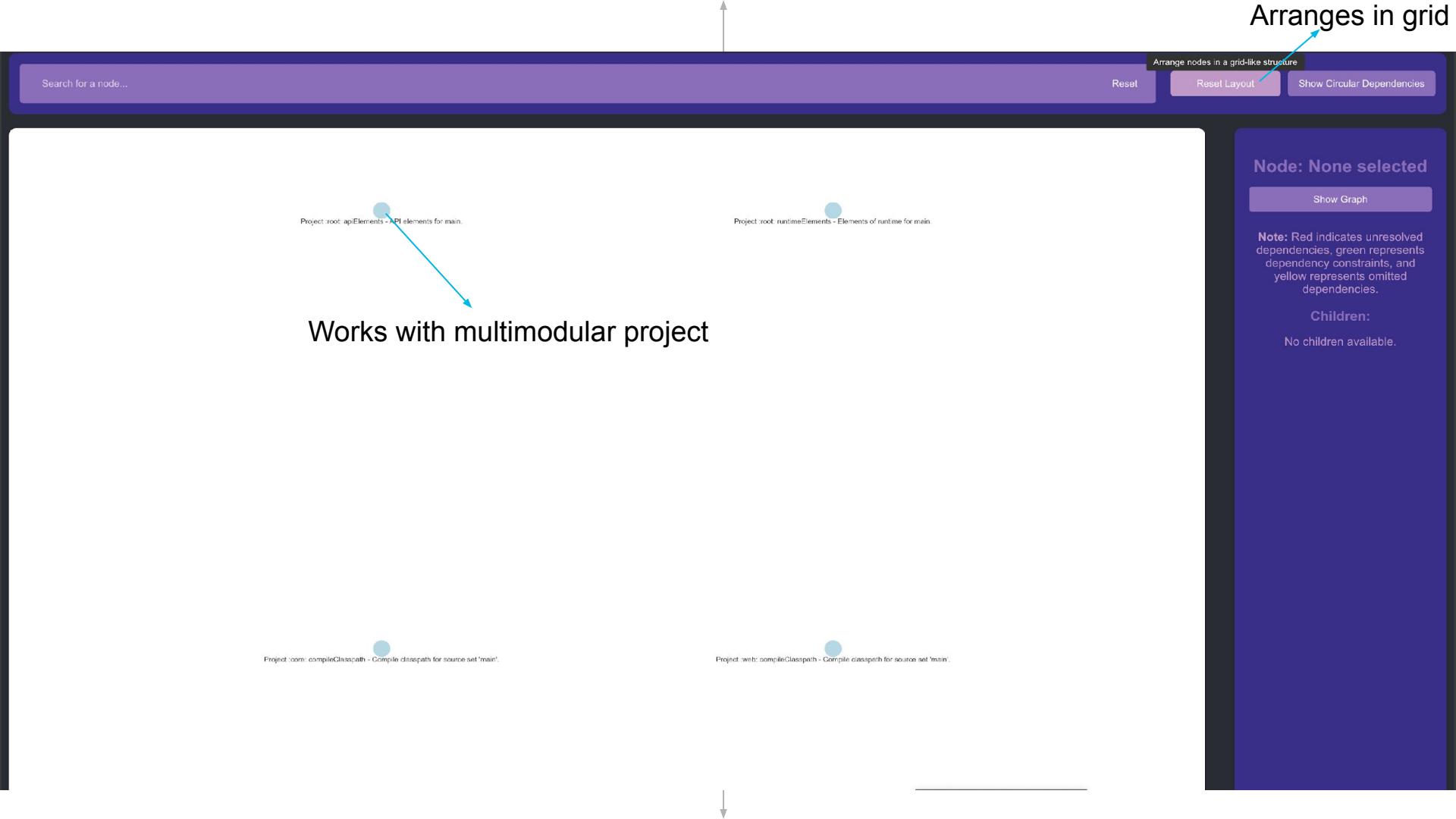
Download
button

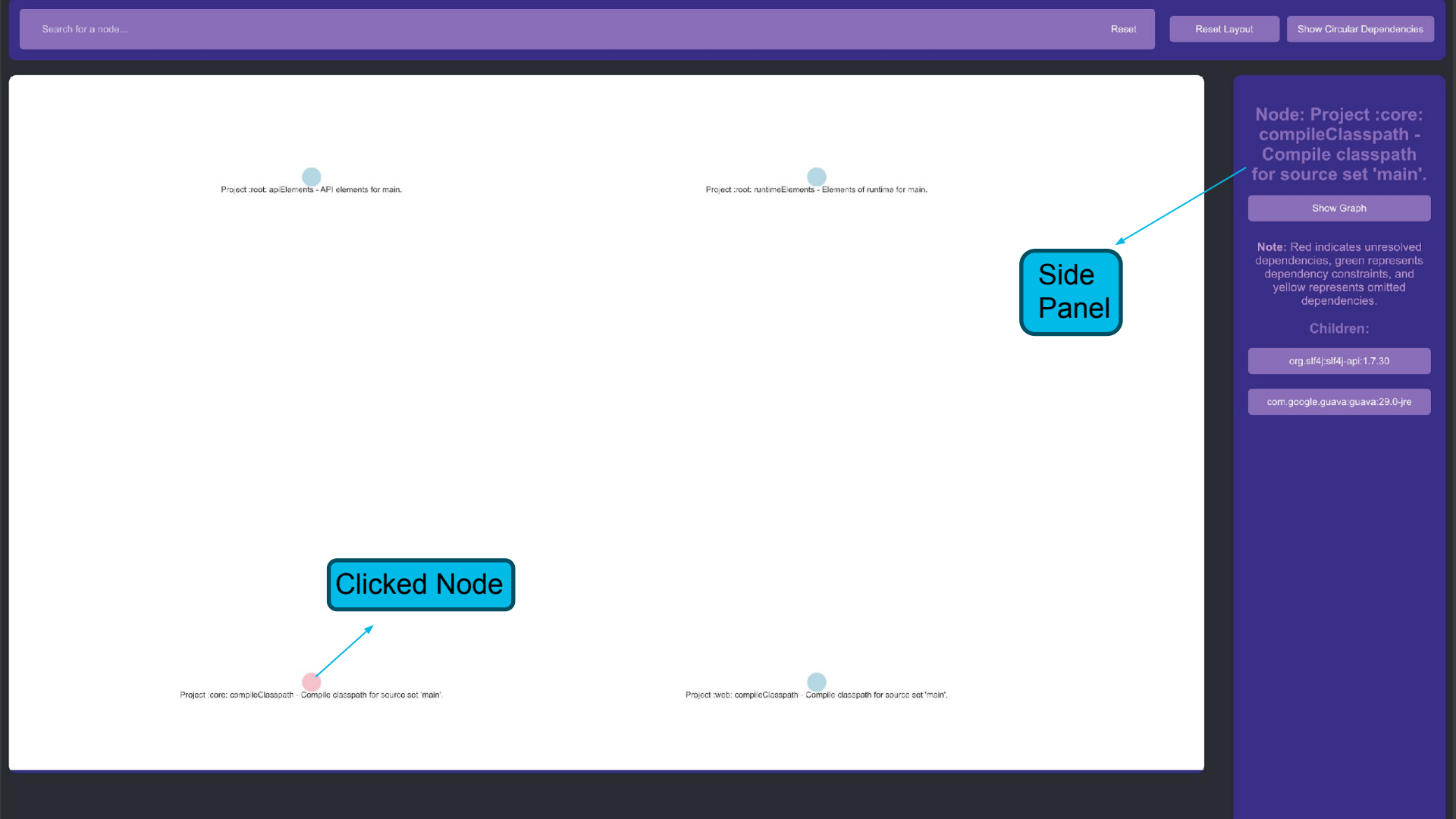
Configurations
/ scopes



Note explains the
meaning of coloured
borders


Dependency Graph Visualizer
Upload your file to generate a graph
dependencies-sitecore-dam.log
Generate Graph


Arranges in grid
Search for a node...
Reset
Arrange nodes in a grid-like structure
Reset Layout
Show Circular Dependencies
Project :root: apiElements - API elements for main.
Project :root: runtimeElements - Elements of runtime for main.
Works with multimodular project
Project :core: compileClasspath - Compile classpath for source set 'main'.
Project :web: compileClasspath - Compile classpath for source set 'main'.
Node: None selected
Show Graph
Note: Red indicates unresolved dependencies, green represents dependency constraints, and yellow represents omitted dependencies.
Children:
No children available.


Search for a node...
Reset
Reset Layout
Show Circular Dependencies
Project :root: apiElements - API elements for main.
Project :root: runtimeElements - Elements of runtime for main.
Side Panel
Clicked Node
Project :core: compileClasspath - Compile classpath for source set 'main'.
Project :web: compileClasspath - Compile classpath for source set 'main'.
Node: Project :core: compileClasspath - Compile classpath for source set 'main'.
Show Graph
Note: Red indicates unresolved dependencies, green represents dependency constraints, and yellow represents omitted dependencies.
Children:
org.slf4j:slf4j-api:1.7.30
com.google.guava:guava:29.0-jre

Search for a node...

Reset

Arrange in Grid

Show Circular Dependencies

Node: Project :web: compileClasspath - Compile classpath for source set 'main'.

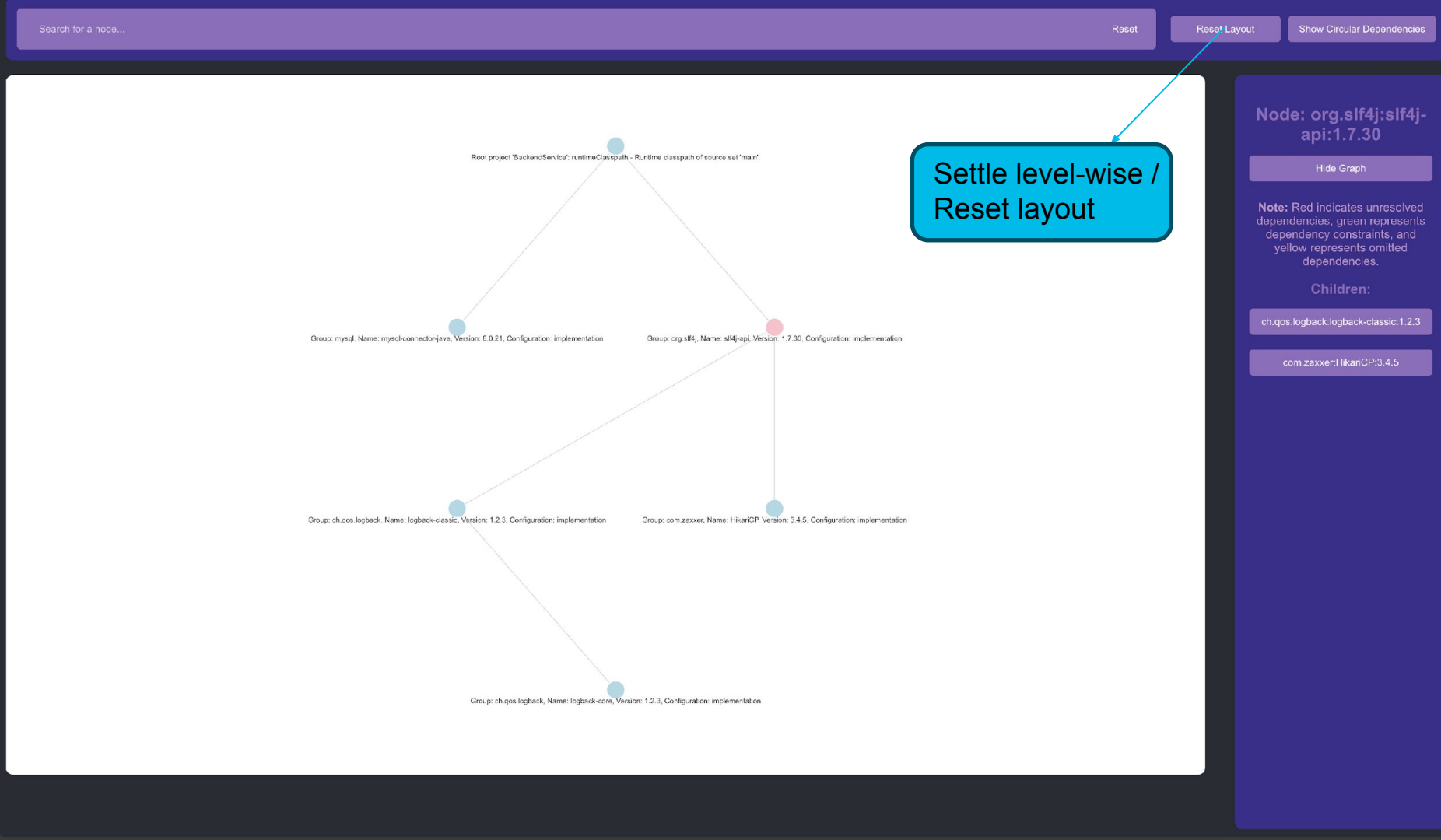
Show Graph

Note: Red indicates unresolved dependencies, green represents dependency constraints, and yellow represents omitted dependencies.

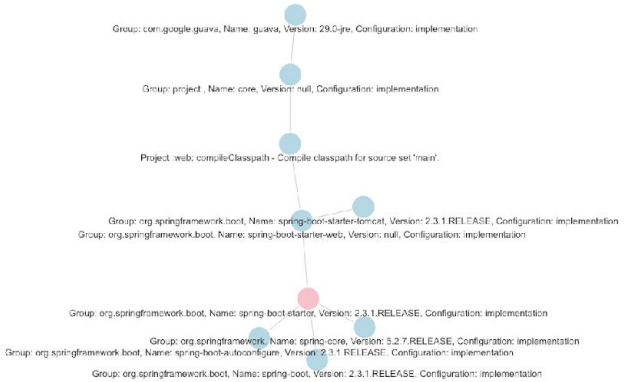
Children:

org.springframework.boot:spring-boot-starter-web

project :core


Search for a node...
Reset
Reset Layout
Show Circular Dependencies
Root: project 'BackendService': runtimeClasspath - Runtime classpath of source set 'main'.
Group: mysql, Name: mysql-connector-java, Version: 8.0.21, Configuration: implementation
Group: org.slf4j, Name: slf4j-api, Version: 1.7.30, Configuration: implementation
Group: ch.qos.logback, Name: logback-classic, Version: 1.2.3, Configuration: implementation
Group: com.zaxxer, Name: HikariCP, Version: 3.4.5, Configuration: implementation
Group: ch.qos.logback, Name: logback-core, Version: 1.2.3, Configuration: implementation
Settle level-wise / Reset layout
Node: org.slf4j:slf4j-api:1.7.30
Hide Graph
Note: Red indicates unresolved dependencies, green represents dependency constraints, and yellow represents omitted dependencies.
Children:
ch.qos.logback:logback-classic:1.2.3
com.zaxxer:HikariCP:3.4.5

Search for a node... Reset Settle Levelwise Show Circular Dependencies



Shows a list of children

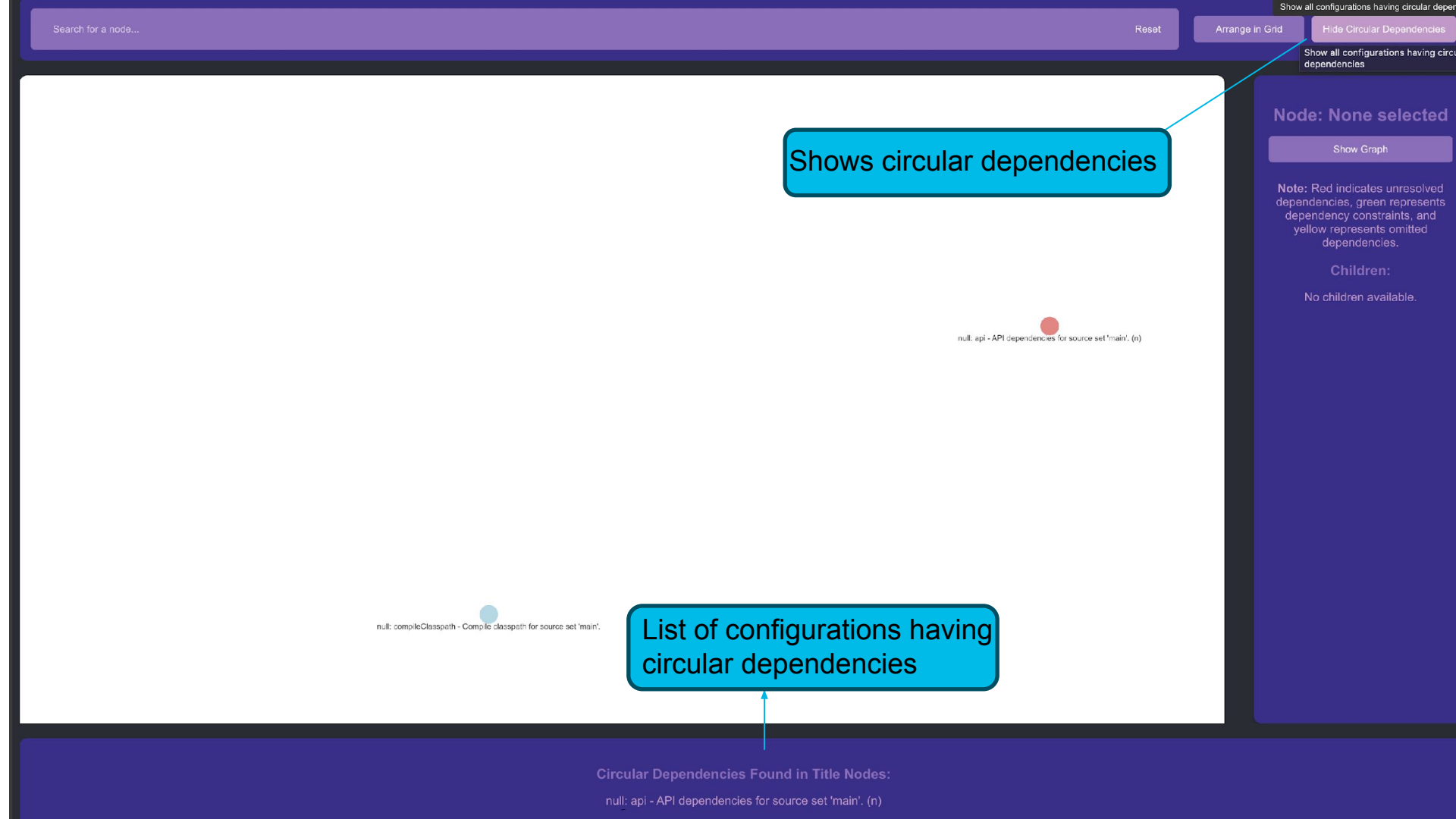
Node: org.springframework.boot:spring-boot-starter:2.3.1.RELEASE

Hide Graph

Note: Red indicates unresolved dependencies, green represents dependency constraints, and yellow represents omitted dependencies.

Children:

- org.springframework.boot:spring-boot:2.3.1.RELEASE
- org.springframework.boot:spring-boot-autoconfigure:2.3.1.RELEASE
- org.springframework:spring-core:5.2.7.RELEASE


Search for a node...
Reset
Arrange in Grid
Hide Circular Dependencies
Show all configurations having circular dependencies
Node: None selected
Show Graph
Note: Red indicates unresolved dependencies, green represents dependency constraints, and yellow represents omitted dependencies.
Children:
No children available.
null: api - API dependencies for source set 'main'. (n)
null: compileClasspath - Compile classpath for source set 'main'.
Shows circular dependencies
List of configurations having circular dependencies
Circular Dependencies Found in Title Nodes:
null: api - API dependencies for source set 'main'. (n)

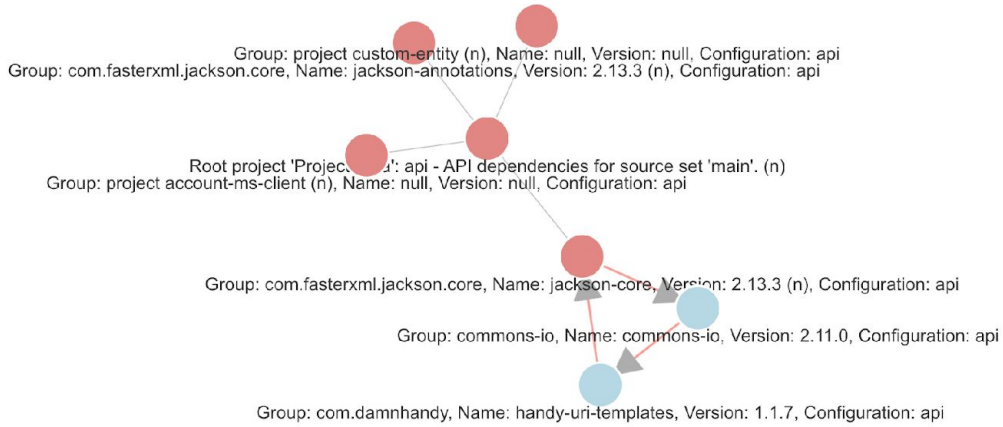
ch for a node...

Reset

Arrange in Grid

Settle Levelwise

Hide Circular Dependencies



Node: Root project 'Project Zeta': api - API dependencies for source set 'main'. (n)

Hide Graph

Note: Red indicates unresolved dependencies, green represents dependency constraints, and yellow represents omitted dependencies.

Children:

com.fasterxml.jackson.core:jackson-core:2.13.3 (n)

com.fasterxml.jackson.core:jackson-annotations:2.13.3 (n)

project custom-entity (n)

project account-ms-client (n)

Circular Dependencies Found in Title Nodes:

Search for a node... Reset Settle Levelwise Show Circular Dependencies

Color Coded Nodes

Node: null: compileClasspath - Compile classpath for source set 'main'.

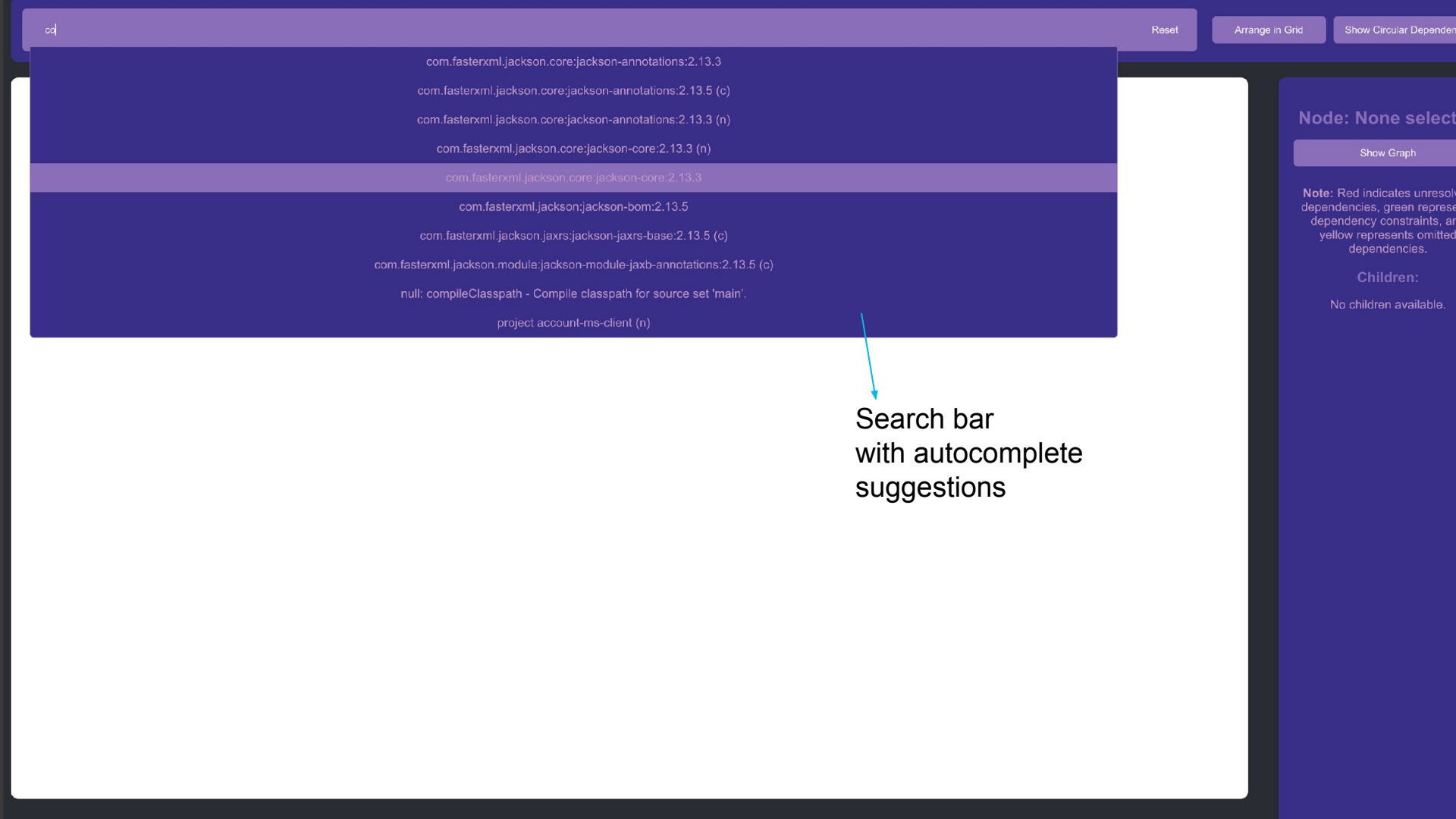
Hide Graph

Note: Red indicates unresolved dependencies, green represents dependency constraints, and yellow represents omitted dependencies.

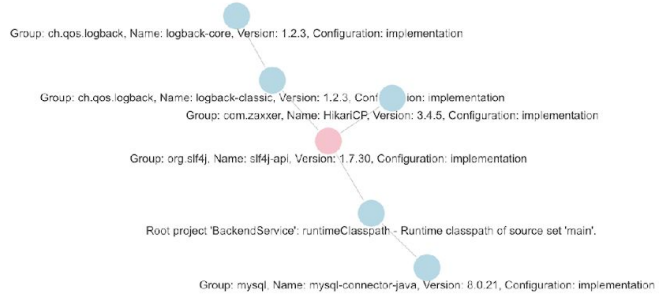
Children:

com.fasterxml.jackson.core:jackson-core:2.13.3

com.fasterxml.jackson.core:jackson-annotations:2.13.3


co
Reset
Arrange in Grid
Show Circular Dependen
com.fasterxml.jackson.core:jackson-annotations:2.13.3
com.fasterxml.jackson.core:jackson-annotations:2.13.5 (c)
com.fasterxml.jackson.core:jackson-annotations:2.13.3 (n)
com.fasterxml.jackson.core:jackson-core:2.13.3 (n)
com.fasterxml.jackson.core:jackson-core:2.13.3
com.fasterxml.jackson:jackson-bom:2.13.5
com.fasterxml.jackson.jaxrs:jackson-jaxrs-base:2.13.5 (c)
com.fasterxml.jackson.module:jackson-module-jaxb-annotations:2.13.5 (c)
null: compileClasspath - Compile classpath for source set 'main'.
project account-ms-client (n)
Node: None select
Show Graph
Note: Red indicates unresol
dependencies, green represe
dependency constraints, ar
yellow represents omitted
dependencies.
Children:
No children available.
Search bar
with autocomplete
suggestions

org.slf4j:slf4j-api:1.7.30 Reset Settle Levelwise Show Circular Dependen



Node: org.slf4j:slf4j-api:1.7.30

Hide Graph

Note: Red indicates unresolv dependencies, green represe dependency constraints, ar yellow represents omitted dependencies.

Children:

ch.qos.logback:logback-classic:1.

com.zaxxer:HikariCP:3.4.5

Graph generated by search bar

com.google.guava:guava:31.0.1-jre

Reset

Settle Levelwise

Show Circular Dependencies

Node:
com.google.guava:guav
jre

Hide Graph

Note: Red indicates unresolved dependencies, green represents dependency constraints, and yellow represents omitted dependencies.

Children:

No children available.

What if searched dependency is present in multiple configurations?

Select a scope for the dependency:

Root project: compileClasspath - Compile classpath for source set 'main'.

Root project: implementation - Implementation only dependencies for source set 'main'. (n)

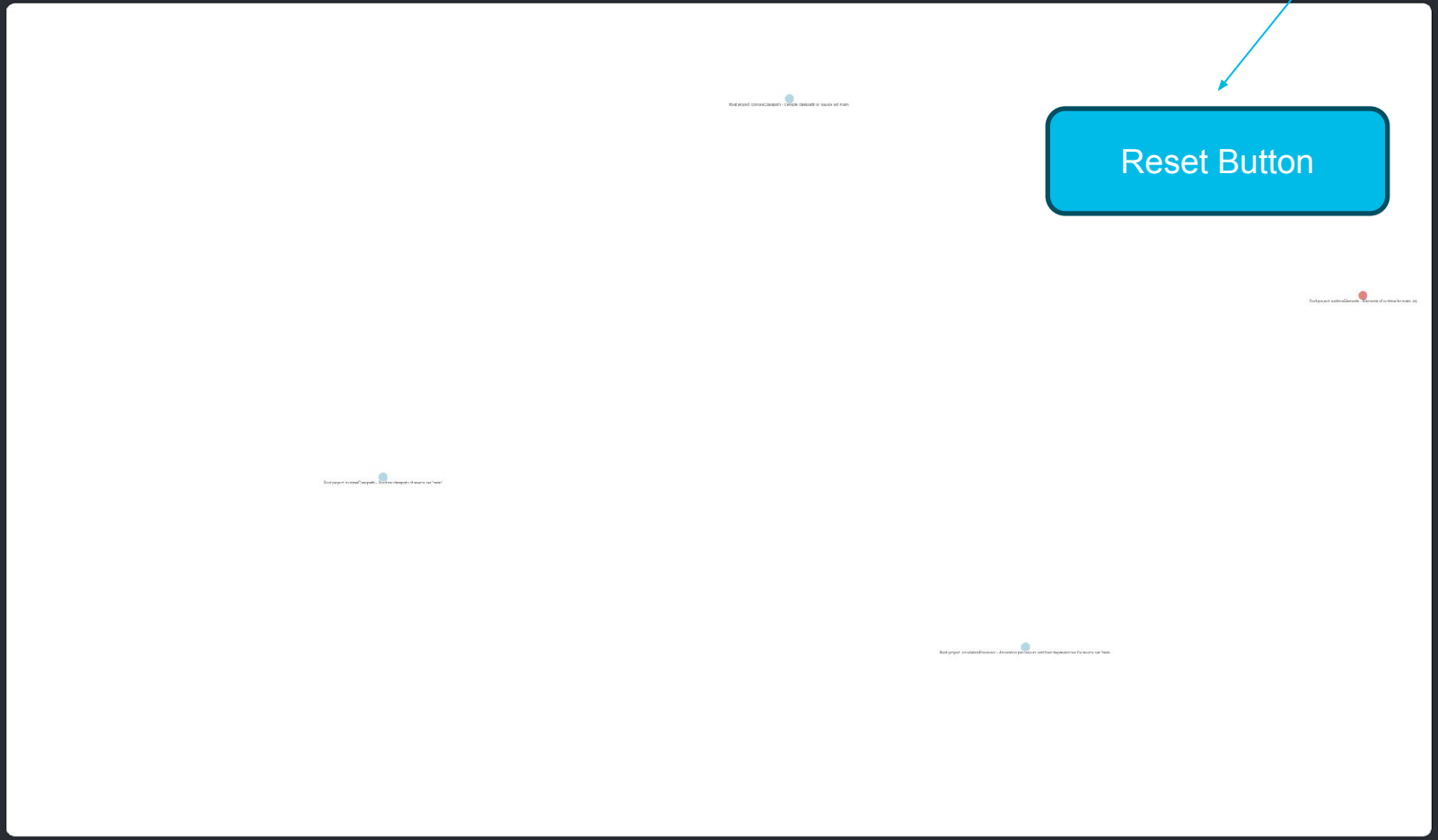
Root project: runtimeClasspath - Runtime classpath of source set 'main'.

Search for a node...

Reset

Arrange in Grid

Show Circular Dependencies



Node: None selected

Show Graph

Note: Red indicates unresolved dependencies, green represents dependency constraints, and yellow represents omitted dependencies.

Children:

No children available.

Arrange nodes in a grid-like structure

Reset

Arrange in Grid

Show Circular De

Hover over feature

Node: None se

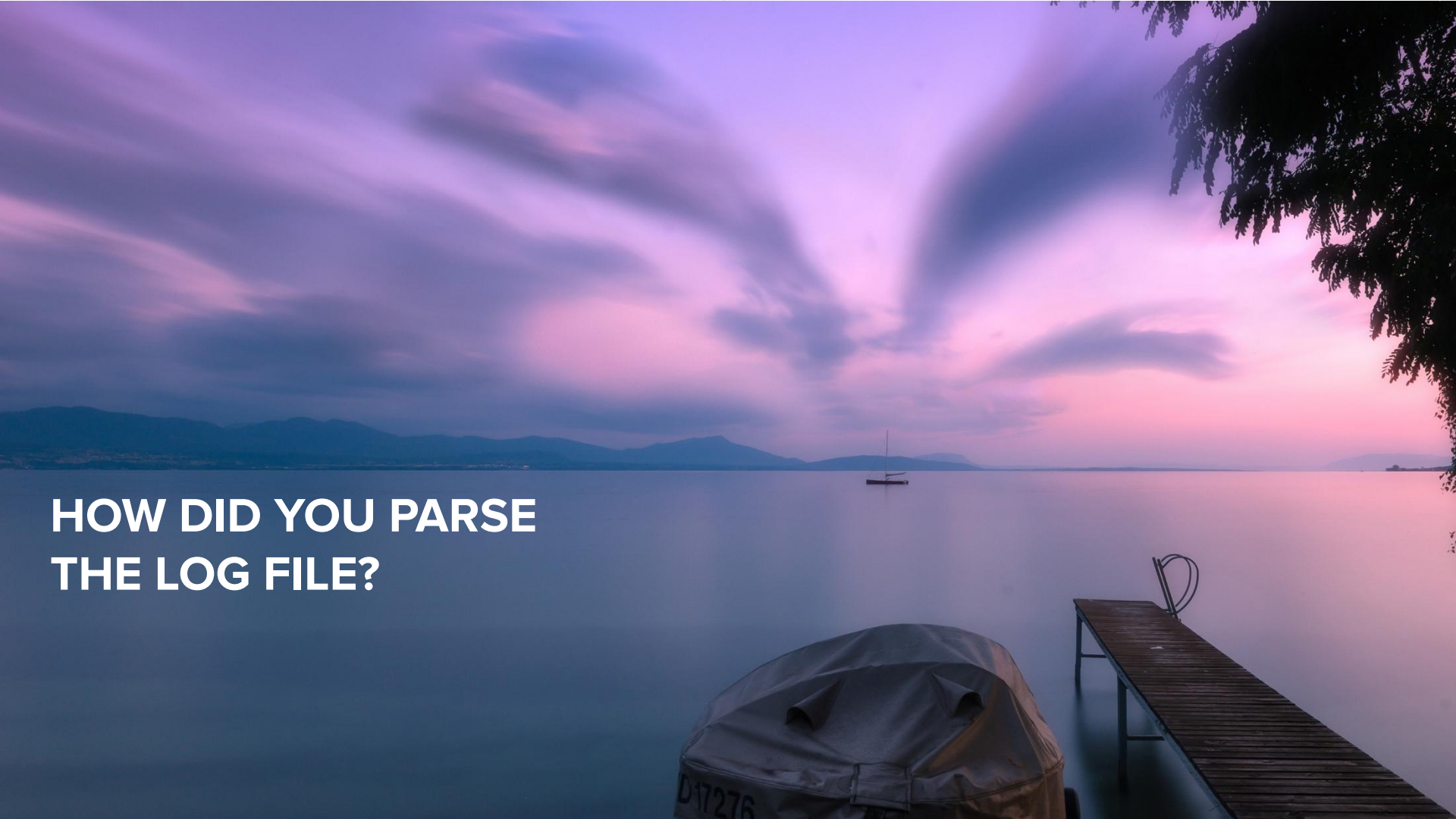
Show Graph

Note: Red indicates u
dependencies, green r
dependency constrai
yellow represents c
dependencies

Children:

No children availa


WHAT ALL IS ALLOWED TO UPLOAD?


HOW DID YOU PARSE
THE LOG FILE?

Parsing logic overview

```
if (line.startsWith("+---") || line.startsWith("\\---")) {
    String dependency = stripPrefix(line).replaceAll( regex: "->.*", replacement: "").trim();
    DependencyNode node = new DependencyNode(dependency, getIndentLevel(line), parseDependencyDetails(dependency, currentConfiguration));
    stack.clear();
    stack.push(node);
    dependencies.putIfAbsent(node.name, new ArrayList<>());
    dependencies.get(currentTitle).add(node.name); // Add this line to identify direct children of the title node
} else {
    int indentLevel = getIndentLevel(line);
    String dependency = stripPrefix(line).replaceAll( regex: "->.*", replacement: "").trim();

    while (!stack.isEmpty() && indentLevel <= stack.peek().indentLevel) {
        stack.pop();
    }

    if (!stack.isEmpty()) {
        DependencyNode parent = stack.peek();
        dependencies.computeIfAbsent(parent.name, k -> new ArrayList<>()).add(dependency);
        dependencies.putIfAbsent(dependency, new ArrayList<>());
        stack.push(new DependencyNode(dependency, indentLevel, parseDependencyDetails(dependency, currentConfiguration)));
    }
}
```


HOW DID YOU
DETECT
CYCLES?

Cycle Detection Algorithm

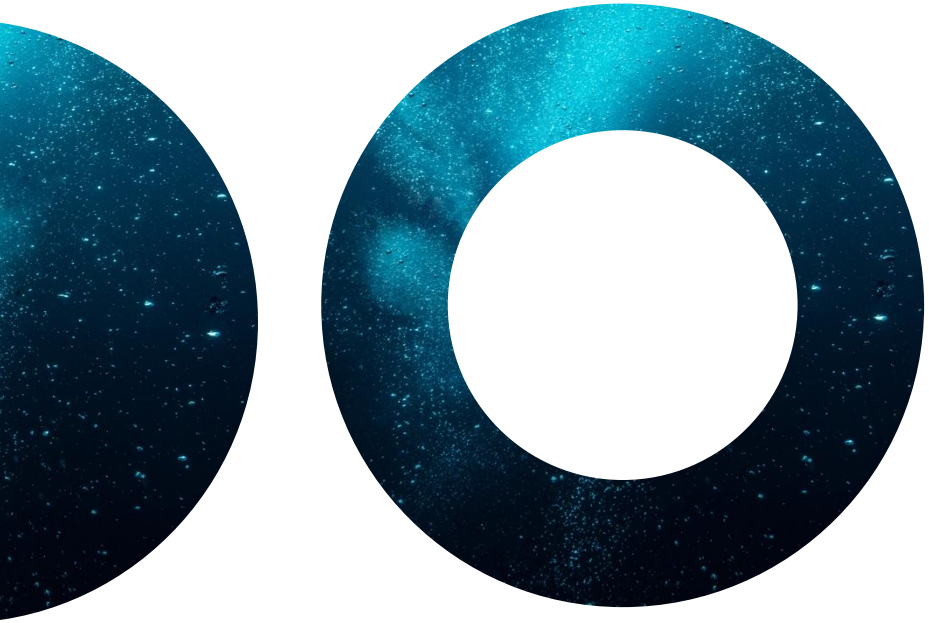
```java
private void detectCycles(String node, Map<String, List<String>> graph, Set<String> visited, Set<String> recStack, List<String> path) {  2 usages  Arman Makhani
    if (recStack.contains(node)) {
        int index = path.indexOf(node);
        if (index != -1) {
            List<String> cycle = path.subList(index, path.size());
            for (int i = 0; i < cycle.size(); i++) {
                circularEdges.add(cycle.get(i) + "->" + cycle.get((i + 1) % cycle.size()));
            }
        }
        return;
    }
    if (visited.contains(node)) {
        return;
    }
    visited.add(node);
    recStack.add(node);
    path.add(node);
    List<String> children = graph.getOrDefault(node, Collections.emptyList());
    for (String child : children) {
        detectCycles(child, graph, visited, recStack, path);
    }
    recStack.remove(node);
    path.remove( index: path.size() - 1);
}
```

Future scope

Use @EnableAsync and Completable future to build the graph incrementally on different threads

Use search bar for different kinds of queries

LINKS TO PROJECT AND DOCUMENTATION:



Project and Documentation Link:

CLICK HERE

Thank You For The Opportunity

arman.makhani@sprinklr.com