CASTOR: Project DepClean

GSoC 2021 Proposal

Name: Abhay Singh

GitHub Handle: https://github.com/ABHAY0O7

Project Details

DepClean is a tool to automatically remove dependencies that are included in the dependency tree of java projects but are not used in code. It detects and removes all the unused dependencies declared in the pom.xml file of the project or imported from its parent pom.xml. The project has four ideas from the official ideas list of CASTOR and I will work on three of them.

- ❏ Implement a Gradle plugin for DepClean. [See]
- ❏ Offer support for the analysis of multi-module projects.
- ❏ Adding tests, Javadoc, and refactoring of the whole project.

Targets and Description

NOTE: For all the plans, there will be a preplanning and discussion phase which includes discussion of each task's plan or modification of proposed plans by me.

1. Implementation of DepClean Gradle plugin. (Main goal)

Current status and Requirements

Currently, depclean offers a depclean-maven-plugin for maven-based projects only. But as now Gradle is seeming to grasp the future as it has features of both Ant and Maven therefore the project requires the same functional plugin for Gradle-based projects too.

Plan

I planned to extend and distribute this part over two months to achieve the following:

- ❏ Writing code, testing, and completing the plugin to make it able to be merged in the master branch. (1 month)
- ❏ Fixing as many as possible bugs (if there are) that will arise after using the Gradle plugin. (next 1 month, simultaneous with other tasks).

Approach

The process will be divided into 5 components.

- ❏ Plan – Before the starting of GSoC period I will completely analyze and figure out that which part of code that is already written in the unstable gradle_plugin branch can be directly use in my plan like currently the class 'node.java' which is present in 'dt' directory can be very useful in my work as it has methods to get general information of a dependency like it's scope or id. After this, when the GSoC period start first, I will be trying to use most of the features (code) from maven plugin by refactoring (replacing) the keywords, methods or interfaces used from Maven Core 3.6.0 API in maven plugin with so of Gradle API 7.0 which performs the same task in gradle plugin. Then, modification in the core module will be

processed so that it can support both the plugins. With all these implementations I will try to produce the same report as produced by maven for Gradle project. Also, the rules required to write a Gradle plugin will be followed simultaneously. Then considering the fact that Gradle is highly configurable and complex, some new additional features will be added in the gradle module to overcome this complexity.

- ❏ **Code –** Code the mentioned and planned features with some tests.
- ❏ **Test or Analyze –** Some issues will arise when we code or after code and we cannot record them while planning. Also, I will put forward the things which I might notice during the code period to discuss. This may include new features, changes, enhancement, etc. Furthermore, I will look for the errors if there are any, or if the implementation doesn't meet the requisite requirements.
- ❏ **Change –** Implement the suggested changes.
- ❏ **Document –** Make changes in the documentation about the newly implemented feature.

I am planning to make this plugin completely usable in the very first month (before the first evaluation) and as for the remaining time I will fix as many as possible bugs that will arise in the future.

2. Offer support for the analysis of multi-module projects. (Main goal)

Current status and Requirements

Currently, depclean does the analysis of a multi-module maven project in the following way.

I. It analyzes a single module at a time and reports the dependency if it is "potentially unused".
II. The problem is that if a module (A) depends on another module (B) and use a dependency (X) which is declared in A but used in only (B) then the DepClean report generated from module A will remove the dependency X and hence the build on module B will be failed due to missing dependency.
III. Therefore, different analyses and reports are required of dependent modules to prevent losses of dependencies like X in the previous point.
IV. For more information visit here.

Plan

To support analysis of multi-module projects, I planned to include an extra section in the overall analysis report that will tell the dependencies which are shared between dependent modules by comparing used declared and unused declared artifacts of different modules. If this proposed plan works as thought then this task will not take more than 1 week with proper testing and documentation. Also, in the pre-planning phase we can modify the current plan by adding some more ideas like "this analysis should be done only when modules are dependent on each other. (This will save time of analysis)"

Approach

The approach will be straight forward as planned. First, implementation of required will be done and then the testing part. If until now all goes in favour or as planned then documentation will be completed with possible refactoring. And for the rest of the GSoC period bugs (if any) solving will be done along with other tasks.

3. Adding tests, Javadoc, and refactoring of the whole project. (Main goal)

Current status and Requirements

Currently, the project runs Integrated Tests (ITs) with JUnit5 using the maven-it-extension plugin. In the project, there are a total of 4 test classes 1 for depclean-core (no logical existence) and 3 for depclean-maven-plugin and this number is very less according to what is the project size. So, some tests can be implemented. Also, the existing ones can be extended too.

Plan

As for the maven plugin module, there are already some tests written for all the classes, though these tests can be extended but my main focus will be on the core module where ITs are requisite. Here is a set of classes where tests could be implemented.

- ❑ DefaultClassVisitor
- ❑ DefaultAnnotationVisitor
- ❑ DefaultFieldVisitor
- ❑ DefaultMethodVisitor
- ❑ DefaultSignatureVisitor
- ❑ DependencyClassFileVisitor
- ★ The above mentioned classes are completely reliable on Result collector, so there will be no need to include a separate test for it.
- ❑ ClassMembersVisitorCounter
- ❑ ProjectDependencyAnalysis
- ❑ DefaultProjectDependencyAnalyzer
- ❑ NodeAdapter
- ❑ DepCleanMojo (Extend)

I am planning to give at least two weeks for this task. After this if time favours me then I will move to writing new tests or extending the existing tests written for the maven module.

Approach

The classes that I mentioned above will be primary ones and if there are some more classes to be included then we can discuss those in the pre-planning phase. The type of tests that I will be writing is of this(#74) type for most of the classes. Also, as most of the classes in 'asm' directory are reliable on 'ResultCollector' and 'ClassMembervisitorCounter' therefore all the tests for classes in 'asm' directory that I will be writing will be based on these two classes (not completely).

TIMELINE

Following is a breakout of the time I'll spend on different aspects of the Project in two different cases:

In Best Case

This is the expected breakdown when all the things go the same way that I planned

- ❏ 50% - Implementation of Gradle plugin.
- ❏ 20% - Multi-module support.
- ❏ 30% - Adding tests, Javadoc, and refactoring.

In Worst Case

This is the expected breakdown when any task takes more time than expected.

- ❏ 20% - Implementation of Gradle plugin.(Not complete)
- ❏ 30% - Multi-module support.
- ❏ 50% - Adding tests, Javadoc, and refactoring.

★ If unable to complete Gradle plugin (rare case) task then the DepClean bot and DepClean page plan will be adopted and for that the breakout will be MM support - 25%, Test - 35%, bot - 20%, page - 20% and timeline will be updated accordingly.

Along with the following timeline, I plan to write a blog every 2 weeks, to tell my experience, what I learned, problems I faced.

Week	Start Date	End Date	Tasks to be Completed
Community Bonding (17th May – 7th June)			
	17 May	22 May	• Deep discussion about the project and getting more familiar with the community and mentors. • Understanding the project more deeply. • Solving pre-existing issues. • Gather prerequisite knowledge (if required).
Pre-Planning phase(23rd May – 7th June)			
	23 May	30 May	• Discuss and plan the implementation of the Gradle plugin. • Discuss the new features (if) for the plugin. • Gather prerequisite knowledge.

			• Document the plan.
	31 May	2 June	• Discuss the plan for the support of multi-module analysis if the proposed plan didn't work. • Discuss the amendments in the proposed plan (if any). • Finalize and document the plan.
	3 June	5 June	• Discuss about the tests for those classes which are not included by me. • Discuss the amendments in my already proposed plan and approach. • Discuss about where the existing tests can be extended. • Finalize the list of classes and document the plan.
	6 June	6 June	• Update in any of the previous plans (if required). • Finally document all the updated /non-updated plans in one place.

Coding Period I			
Coding Phase 1	7 June	16 July	Summary – • Implementation of complete Gradle plugin. • Testing and documentation of Gradle plugin. • Submit PRs of the completed work before first evaluation.
Week 1	7 June	13 June	• Using the maven plugin code in the Gradle plugin and adding some new code while retaining the same logic. • Modifying the DepClean core and adding some new features to it so that both the plugins can rely on it.
Week 2 & 3	14 June	27 June	• Writing code and adding new features for the Gradle plugin in Gradle module. • Implement the plugin.
Week 4	21 June	27 June	• Writing tests and deploying the whole plugin.

Week 5	28 June	4 July	• Testing, analyzing, and documenting the plugin and make it ready to use.
Evaluation 1	12 July	16 July	• Write documentation for all the implemented features. • Make Pull Requests for work done. • Solving existing or new bugs from new features in the backlog.
Coding Period – II			
Coding Phase 2	17 July	16 August	Summary – • Completing the multi-module analysis task. • Testing and documenting it for usage. • Add tests, documentation, and refactoring of the whole project.
Week 6	17 July	23 July	• Implement the multi-module analysis support as planned. • Adding test for the implemented feature.

			• Writing proper documentation for work done and refactoring where possible.
Week 7 & 8	24 July	6 August	• Adding tests for the classes in the same sequence as listed in plan. • Adding tests for the classes introduced during the pre-planning phase. • Extending tests (if possible) for the existing ones. • Adding proper documentation for all the added tests.
Week 9	7 August	13 August	• Adding Javadoc for the whole project. • Refactor the whole project wherever possible. • Fixing bugs and issues. • Completing pending tasks (if any).
Final Evaluation	16 August	23 August	• Make PR for work done. • Summarize and document all the work done during the GSoC period.

			• Completing all the pending tasks (if any). • Adding Javadocs wherever left. • Fixing bugs and issues in the backlog.
Post GSoC Period			• After GSoC, I would love to keep on contributing and give my best for the organization. My major motivation is to use the experience I gained in these 3 months and utilize the knowledge gained.

Why Me?

I started programming just 4 months ago when my online classes were started for my B. Tech course. Before this, I didn't have any computer/laptop, not even in my high school course and didn't have any idea about programming. But as soon as I explored the world of programming, it became my habit and in just a short period I got my hands dirty on various tech-stacks like started with C/C++ (Ongoing), explored design, android, and web, and currently going with Java. So, I can learn new things fast and effectively.

Along with all this stuff I also explored open-source as soon as possible and started contributing to it with my technical and mental skills. Contributions give me knowledge and confidence to discover new things. Also, I am good at multitasking as in the past 4 months I managed my college hectic schedule and additionally learned all the mentioned tech-stacks and contributed to open source. Also, I recently won second prize in 'GitHero', a git competition organized by our college annual techfest 'APAROKSHA'. This symbolises that I have good git skills and git will not be any problem for me during the contribution.

Currently, I don't have any personal projects in technical life except my knowledge and devotion to programming. I will be glad if I get a chance to contribute to this project and working with such a wonderful community will definitely boost my knowledge and career.

Personal Details

- **Name:** Abhay Singh
- **Current Address:** Aligarh, Uttar Pradesh, India
- **Institution:** Indian Institute of Information Technology, Allahabad (https://www.iiita.ac.in/)
- **GitHub:** https://github.com/ABHAY0O7
- **LinkedIn:** https://www.linkedin.com/in/abhay-singh-5a6b71201/
- **E-Mail Address:** abhaysingh936851@gmal.com [preferred], abhaysingh7895@gmail.com, iec2020080@iiita.ac.in
- **Contact Number:** +91-9368513197 [Only WhatsApp]
- **Discord Tag:** 【『𝕁𝕆𝕂𝔼ℝ』】#0757
- **Time Zone:** (Asia/Kolkata) +0530
- **Mailing list name:** ABHAY SINGH

Area of Study in College

- **Course and Branch:** Bachelor of Technology, Electronics, and Communication Engineering
- **Semester and Year:** Second (2nd) (will start from April 10), First (1st) year.
- **Relevant Completed/Ongoing Courses:**
  - Java (4/5)
  - C/C++ (Ongoing) (4/5)
  - Data Structures and Algorithms (Ongoing) (4/5)
  - HTML & CSS (Basics) (3/5)
  - Android (Basics) (2/5)
  - JavaScript (Ongoing) (1/5)

Career Goals

Once I read, “If you want a change in this world, be the change” since then I want to do something for the world which can cause a change to it. I want to help people with my knowledge and want to make them happy just like JOKER. There was a time when I wanted to be an entrepreneur but as of now I just want to help people irrespective of what I am and what I will be (developer, entrepreneur, or software engineer).

Summer commitments other than this project

Project DepClean will be my only commitment. But, in my college, there is a very rare chance of summer vacation, due to the online semester. My classes will continue till July, 2021 with this(may change) time-table. Considering the project size and no. of tasks, I plan to give 30 - 35 hrs. per week irrespective of my classes and GSoC guidelines. Also, after July there will be

semester-end holidays of 10 days (max), in which I will give 40 – 45 hrs. a week. But, I will not be available during my final semester (3-4 days) exam and when I am shifting from home to college in august (1-2 days) or will be available only for an hour or two. But I will cover this time by devoting some extra hrs. throughout the program.

Link to my tentative academic schedule during April to July. Click here

★ Our college isn't too strict for attendance and I can work on the project even during the classes as they are online. Also, I know some professors who will give me the required attendance, so college during summer will be no issue for me and the project. I will be at least giving the hours that I mentioned i.e. 30 - 35.

Contributions and patches for CASTOR DepClean

I have made a total of 5 PRs as of 13 April 2021. There are a total of 2 + 5 issues which are opened by me in which 2 are currently open and 5 are closed. I already had a good conversation with the mentors at various issues and PRs. Also, I helped some new contributors who are facing some difficulties during their contribution.

Preferred method of communication

My preferred method of communication is email (preferred one), but one can contact me on any of my personal handles that I mention in my personal details.

Previous GSoC Participation

No, this is the first time I am applying for GSoC.

If I am applying for various Organizations, this year

No, I am only applying for CASTOR (Project DepClean) this year.

After GSoC

I will like to keep developing this project after GSoC and will be available to contribute. Even if I am not selected this year, I will like to help this project by resolving issues, suggesting new ideas, and participating in discussions. Also, I am looking to contribute to some new and fresh projects by CASTOR like depanalyzer and depheaven which are currently managed by my mentors.

Also, I usually help out people with code and will love to mentor some young coders, preferably in java and C/C++ projects.

Conclusion

Thank you for reading my Application, I am very interested in taking this project and will love to see it being used in the future. Also working with CASTOR will be a memorable and learning experience for me.

Regards,

Abhay Singh