Group 23

Muaath Alaraj (malara2)

Piyush Atre (pkatre2)

Nilesh Dadi (vdadi2)

Ryan Fraser (rfraser3)

Mohit Munjal (mmunjal2)

Eshia Rustagi (eshiafr2)

Projects Summary

Project #1 - Joinery

Link:

https://github.com/cardillo/joinery

Description:

A data frame implementation in the spirit of Pandas or R data frames.

Slightly more detail:

In Java, a data analysis library for joining together pieces of data to produce insight.

Project #2 - JBot
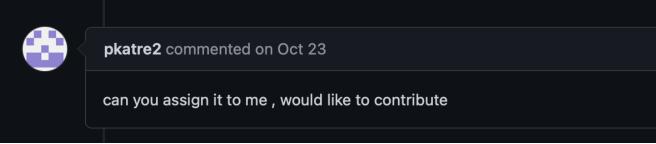
Link:

https://github.com/rampatra/jbot

Description:

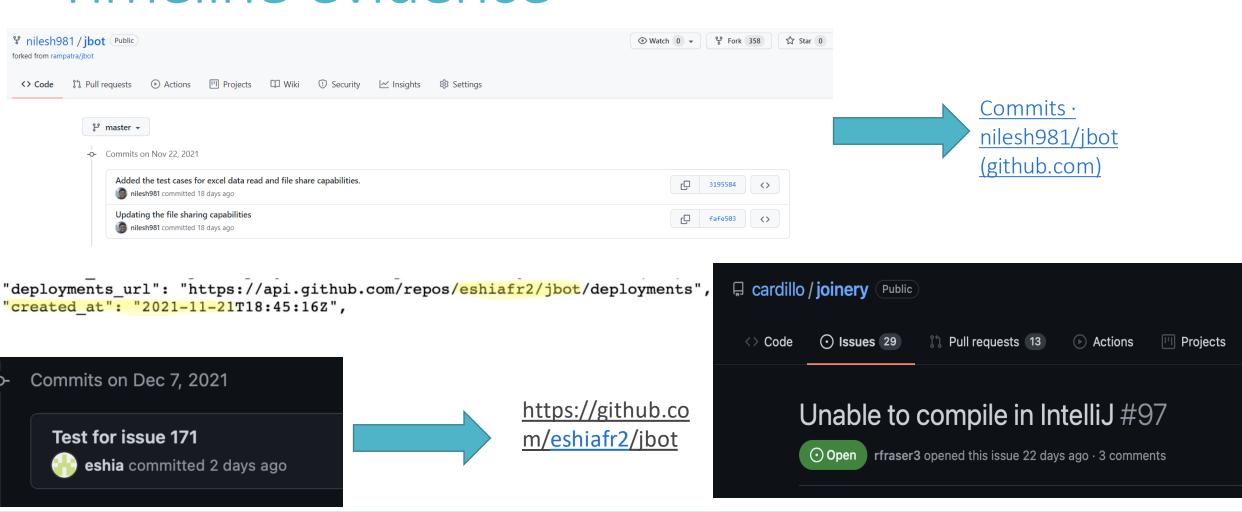JBot makes Slack and Facebook bots in minutes.

Slightly more detail:

Java framework that is open source tool to build bots and custom integrations and provides functionality to make the bot live right away

Timeline

Project Name	Issues Implemented	Student netID(s)	Lines changed	Date Started	Date Completed
Joinery	51 75 83 63 38 93	rfraser3, pkatre2 pkatre2 mmunjal2 rfraser3, pkatre2 mmunjal2, pkatre2 eshiafr2, mmunjal2	+24, 0 +9, 0 +19, -2 0, 0 0, 0 +56, -2	Week 5 Week 4 Week 4 Week 5 Week 5 Week 5	Week 8 Week 7 Week 7 Week 8 Week 7 Week 8
Jbot	197 114 171 172 28	malara2 malara2 eshiafr2 vdadi2 vdadi2	+170, -17 +3, -3 +4, -1 +94, -8 +229, -5	Week 5 Week 3 Week 4 Week 5 Week 3	Week 7 Week 3 Week 8 Week 7 Week 5
Alexa	N/A	Malara2 eshiafr2	Attempted Build Aws implementation	Week 3	N/A
Bumptech Glide	N/A	malara2	Attempted Build	Week 4	N/A

Timeline evidence

pkatre2 commented on Oct 23

can you assign it to me , would like to contribute

nilesh981 / jbot Public
forked from rampatra/jbot

Watch 0   Fork 358   Star 0

<> Code   Pull requests   Actions   Projects   Wiki   Security   Insights   Settings

master

Commits on Nov 22, 2021

Added the test cases for excel data read and file share capabilities.
nilesh981 committed 18 days ago
3195584

Updating the file sharing capabilities
nilesh981 committed 18 days ago
fafe503

Commits · nilesh981/jbot (github.com)

"deployments_url": "https://api.github.com/repos/eshiafr2/jbot/deployments",
"created_at": "2021-11-21T18:45:16Z",

cardillo / joinery Public

<> Code   Issues 29   Pull requests 13   Actions   Projects

Commits on Dec 7, 2021

Test for issue 171
eshia committed 2 days ago

https://github.com/eshiafr2/jbot

Unable to compile in IntelliJ #97

Open   rfraser3 opened this issue 22 days ago · 3 comments

* We worked in our local machine , and did git commit at the end after build was successful.

Timeline evidence

Commits on Nov 21, 2021

Update DataFrame.java
pkatre2 committed 18 days ago
Verified 75eb71e

Update DataFrameSortByTest.java
pkatre2 committed 18 days ago
Verified 3bdec07

Update DataFrame.java
pkatre2 committed 18 days ago
Verified 76c11f9

Update Sorting.java
pkatre2 committed 18 days ago
Verified 16986a2

https://github.com/pkatre2/joinery

add-time-valid...

Commits on Dec 9, 2021

add facebook tests and javadoc
Mistermo716 committed 2 hours ago
5c4ad3b

Commits on Nov 21, 2021

add java doc to tests
Mistermo716 committed 18 days ago
ae0ca4e

fix invalid time for meeting bug
Mistermo716 committed 18 days ago
46b3be6

https://github.com/Mistermo716/jbot/tree/add-time-validation

Commits on Nov 21, 2021

Updating javadoc for added function
Mohit Munjal authored and Mohit Munjal committed 18 days ago
b982ef7

Final Commit
Mohit Munjal authored and Mohit Munjal committed 18 days ago
a2d0897

https://github.com/mmunjal/joinery

* We worked in our local machine , and did git commit at the end after build was successful.

Issues

Joinery: Issue #38

URL: https://github.com/cardillo/joinery/issues/38

NetIDs: Mmunjal2,pkatre2

Feature: interrupt in the shell should stop the currently running statement

Implementation:

By design of kernel, ctrl+c always interrupts the program. To achieve this, we had to use different ctrl commands. We implemented for ctrl+O , by pressing this it will interrupt the currently running execution in the shell. For implementation, we handle using trap function, once this key is pressed it will halt and kill the currently running PID.

Tests added:

JUnit -   1) Run the Shell 2) Load the csv file into DataFrame , enter control+O 3) Operation Should not exit the shell 4) Should only interrupt the current operation.

Difficulties encountered:

had to learn multithreading Java Programming .

Joinery: Issue #51

URL: https://github.com/cardillo/joinery/issues/51

NetIDs: rfraser3, pkatre2

Issue: Export to CSV does not write row names (the index is always dropped)

Root cause: Per Joinery's creator, "writing the index is not supported by the CSV library Joinery uses"

Solution: Add duplicate index column to DataFrame before writing CSV to preserve 1 index after writing CSV

Implementation:
    Loop over the DataFrame's index and add each element to an Object list
    Use that list to create a 1-column DataFrame
    Left-join the 1-column DataFrame and the original DataFrame

Tests added:
    JUnit: assert that the original DataFrame's Column[0] is now the new Column[1]
    JUnit: updated existing test to no longer assume bug is present

Difficulties encountered:
    Pre-existing functions in Joinery have slightly different behavior based on JDK and package versions
    Existing index datatypes make casting to a DataFrame difficult
    Existing tests assume bug is present

Joinery: Issue #63

URL: https://github.com/cardillo/joinery/issues/63

NetIDs: rfraser3, pkatre2

Issue: Enable option to disable default date formatting

Root Cause: original developer forgot to include this option

Solution: this seemed straightforward

Implementation:

   Implemented Flag based approach to disable or enable date formatting.

Tests:

   Enable the option to disable date time format through Shell. 2) create data frame with multiple time formats . 3) Export data into a csv. 4) csv should have all time formats.

Difficulties encountered:

   This one was fairly straightforward.

Joinery: Issue #75

URL: https://github.com/cardillo/joinery/issues/75

NetIDs: pkatre2, mmunjal2

Feature: sort by index

Implementation:
Loop over the DataFrame's index ,compare values and finally append back in sorted order.

Tests added:
JUnit - a) create data frame with multiple columns

b) perform sorting on index of DataFrame

c) dataframe should be sorted on index.

Difficulties encountered:
Joinery Project was not compiling in local , had so many challenges . Finally worked in VM.

Joinery: Issue #83

NetIDs: mmunjal2,pkatre2

Feature: Concatenate two dataframes on same columns

Importance : This issue is important since it allows data manipulation through integration of multiple dataframes. This extensibility improves the capabilities of the joinery project and allows it to perform more advanced operations such as analytics.

Solution: Implement concatenation similar to Python Pandas and allowing user to concatenate either by Inner Join or Outer Join

Implementation: Enhanced concatenate method to support both join type INNER and OUTER. Loop over both dataframes and retain the common columns using LinkedHashSet.

Tests added:

1) Concate operation on 2 Dataframes - a) create data frame A with multiple Columns b) create data frame B with multiple columns c) concate will be cross product of data frame A and B.
2) Concate operation with **OUTER** Join - a) create data frame A with multiple Columns b) create data frame B with multiple columns c) concate will be cross product of data frame A and B.
3) Concate operation with **INNER** Join - a) create data frame A with multiple Columns b) create data frame B with multiple columns c) concate will be inner join product of data frame A and B on joined columns.

Difficulties encountered:

Build failures in local , had to download aspectJ library.

Joinery: Issue #93

URL: https://github.com/cardillo/joinery/issues/93

NetIDs: eshiafr2, rfraser3

Issue: Update writeCsv() function to allow parameter specifying a separator for the CSV

Root cause: Methods that don't have a custom parameter for the CSV

Solution: Update existing methods to take in a separator parameter

Implementation:
    Add the final String separator parameter
    Separate the files based on provided separator parameter

Tests added:
    1) Create Dataframe with multiple columns
    2) Write to csv as tab / pipe etc as delimiter
    3) Open CSV and validate the same output in both cases

Difficulties encountered:
    Building the project was a difficulty and ran into many issues trying to run the tests.
    Understanding the existing codebase and how to implement the custom parameter for the methods.

Jbot Issue #28

URL: https://github.com/rampatra/jbot/issues/28

NetIDs: vdadi2

Issue: Sending files on slack

Root cause: New Feature – Currently Jbot doesn't support upload file capability

Solution: create a method which uses slack web API, meteoroid library to support uploading files using Jbot

Implementation:
Added method to get details such as channel information, user tokens and file information.
Add logic for uploading the file when a user request specific information.

Tests added:
1) User shares the location of the file
2) Bot transmits the file to the user
3) User can read the file using slack or download it locally for consumption.

Difficulties encountered:
Jbot uses a slack RTM API which doesn't support file upload capability, transitioning to Web API and using meteoroid which requires a different version of maven.

Jbot: Issue #114

URL: https://github.com/rampatra/jbot/issues/114

NetIDs: malara2, eshiafr2

Issue: The project is not building with the current set of dependencies defined

Root cause: Springboot version 1.4.0 RELEASE is not compatible with Java 1.8

Solution: Correct the dependencies in the pom.xml file

Implementation:

Add 1.5.3 RELEASE to in the dependencies list. This is the version that is compatible with Java 1.8

Tests added:

The project failed to build and therefore so did the test. However, when built with the new dependencies the project was verified to be working.

Difficulties encountered:

Issues with slack Api integration after successful build

Jbot: Issue #171

URL: https://github.com/rampatra/jbot/issues/171

NetIDs: eshiafr2, malara2

Issue: Function stopConversation(event) is not stopping the conversation

Root cause: Legacy API no longer supports incoming messages

Solution: Verify that the stopConversation(event) works via in depth testing

Implementation:
    Created unit tests to verify that stop event was working
    Ironed out bugs associated with the stopConversation event

Tests added:
    Assert that conversation is stopped only when the user chooses to not request for a meeting upon being asked
    Assert that conversation is stopped always when the user is asked to repeat

Difficulties encountered:
    Building the project was a difficulty and ran into many issues trying to run the tests.
    Understanding why the issue was failing and how to recreate the bug was difficult
    Project needs to start using the new supported Api rather than the deprecated legacy Api for slack

Jbot Issue #172

URL: https://github.com/rampatra/jbot/issues/172

NetIDs: vdadi2

Issue: excel data read

Root cause: There is no way to share an excel file and read the information present in the file

Solution: Provide a method to preview excel information and upload excel file from Jbot

Implementation:
Added method which implements slack Web API, JXL library to upload excel files and read read in excel file
Added logic to preview the excel file information by mentioning row, column

Tests added:
1) Share the file location
2) Bot transmits the excel to the user
3) User asks bot to read a particular cell from excel
4) Bot reads the cell and information is validated by the user

Difficulties encountered:
JXL library has a dependency with springframework due to which okhttp3 was not recognized.

Jbot Issue #197

URL: https://github.com/rampatra/jbot/issues/197

NetIDs: malara2

Issue: Schedule Meeting time accepts any string. Needs time validation

Root cause: There is no form of date validation when receiving time

Solution: provide method that ensures that date/time of a meeting is in the valid format

Implementation:
  Added method to validate date/time to base bot for use in both facebook and slack bots
  Add logic associated with replying to the user about invalid time and loop it back into an event.

Tests added: Total of 4 (2 for Facebook and 2 for Slack)
  Assert that invalid time has a facebook/slack response that their entry was invalid.
  Assert that a valid time for a meeting will respond with a success and meeting event created.

Difficulties encountered:
  Jbot uses a legacy slack api that lost support a couple weeks ago, had difficulty working around it. Incoming messages were no longer functioning.

Static code analysis screenshots

No warnings or issues from our projects code were found in this report.

These are a previous set of issues that the project had from before.

Large list of issues that cannot be resolved within scope of project

PMD Joinery Report

FMB Jbot Report

```
de.flapdoodle.embed.mongo.MongodExecutable
io.undertow.websockets.core.WebSocketChannel
javax.xml.bind.PropertyException
org.glassfish.tyrus.core.WebSocketApplication
javax.management.InstanceAlreadyExistsException
com.caucho.burlap.client.BurlapRuntimeException
org.aspectj.bridge.AbortException
javax.cache.configuration.Configuration
org.springframework.retry.RetryPolicy
org.springframework.retry.backoff.BackOffPolicy
org.apache.tiles.request.servlet.NotAServletEnvironmentException
java.util.prefs.BackingStoreException
org.springframework.security.web.context.AbstractSecurityWebApplicationInitializer
org.springframework.social.twitter.api.Twitter
org.springframework.data.mapping.model.SimpleTypeHolder
org.springframework.data.mapping.model.FieldNamingStrategy
org.springframework.data.mongodb.core.mapping.Document
java.util.IllegalFormatException
javax.mail.internet.AddressException
javax.mail.Message
org.apache.kafka.common.serialization.StringDeserializer
org.apache.kafka.clients.consumer.ConsumerConfig
okhttp3.Cache$urls
kotlin.sequences.IndexingSequence$iterator
```

No warnings or issues from our projects code were found in this report.

These are a previous set of issues that the project had from before.

Large list of issues unrelated to our project (1599 & 1810) that cannot be resolved within scope of project

PMD Jbot Report

- No Issues or warnings added to the list generated by this report.
- Our code matched all the requirements to pass the pmd report.

PMD Joinery Report

- No additional Issue or warning to the report after fixing our issues. It's shown in DIFF command also.

- Our code matched all the requirements to pass the pmd report.

FMB Joinery Report

- No additional Issue or warning to the report after fixing our issues. It's shown in DIFF command also.

- Our code matched all the requirements to pass the FMB report.

Concluding Remarks – Part 1

What lessons did you learn from your project?

Ryan: importance of understanding the code base of a project, its libraries, its dependencies, etc. before beginning work

Muaath: When selecting an open source project make sure it has a long history of being supported and backed.

Piyush : Pair programming , avoid programming by coincidence , importance of automation tools (e.g. PMD , Find Bugs)

Eshia: Contributing to an open-source project, how testing and debugging is relevant and critical

Nilesh: How open source projects run and how one feature might break another, importance of finding bugs and debugging, unit-testing

Mohit - Importance of JUnit test cases , learnt new static code analysis tools and collaborating with diverse team remotely.

What was difficult?

Ryan: getting the project to build and understanding why it wouldn't.

Piyush : team members in diff. Geographical location , hard to co-ordinate meetings.

Muaath: Coordination with everyones busy schedule. Working around jbot's use of a slacks legacy api that stopped working recently.

Eshia: Coordinating meetings across various timezones, reading and understanding how to implement and fix issues in existing codebases, building the projects

Nilesh: Coordinating with different timezones, lack of functionality in RTM api and reliance on third party library, understanding codebases

Mohit : Most difficult part was co-ordination with team as people are in diff. Time zones and working Full time.

What do you wish you could have done (or done differently)?

Ryan: wish I would have been a bit less stubborn and looked for other ways to fix the issue. Would also have been easier to meet with classmates in person to talk through solutions. I would focus less on finding the most efficient solution the first time, and would have started by getting any solution that works, then improving upon that.

Piyush :- Could have focused proportionally on presentation and documentation to showcase equivalent effort spent.

Muaath: I wish I would have chosen a project with a better reputation of being backed.

Eshia: I wish I could've understood the scope of each project that I worked on

Nilesh: Not enough time to select the project, projects had multiple dependencies which made it harder to work on

Mohit : spent more time with my team mates , planned collaboration would have been better .

Concluding Remarks – Part 2

How could your project be extended…what's next?

Ryan: Make existing methods more robust, make new methods we wrote more efficient

Existing methods have nuanced and subtle changes in behavior as packages age – would be interesting to see which methods have changed and if there is a more timeless way to build them

Some of the solutions work quite well, such as writeCsv, but might have performance issues if trying to rapidly create a huge amount of CSVs, create high-frequency logs, etc.

Piyush : More Data and Analytics Use cases can be included (performance optimization when dealing with Dataframes).

Muaath: Jbot needs to be reworked from the ground up to use the new Api slack recommends. This would actually resolve majority of the issues on the github board. (This would require a lot of changes and is out of the projects scope)

Eshia: Fix the pending issues and include more JUnit tests for better debugging

Nilesh: Jbot lacks any kind of AI capability which is quite common across the industry now, it needs to rearchitected to use latest slack API and support cognitive services out of the box.

Mohit - Add more enhancements to Joinery .

Are there any interesting problems or questions that resulted from your work?

Muaath: Learning the codebase of an open source projects takes a lot of time and work.

We spent a few hours on Alexa/bumptech glide and could not get them to work or build correctly.

A lot of work simply does not show up on either the readme or github because of this discovery phase.

I feel everyone in the group put in effort but depending on the issue and its learning curve that really did not show up in the reports.

Ryan: the issue of a project having old or deprecated libraries and packages seems like it could be an entire course on its own

you can assume you know how others will use your code but you can't actually ever know unless they tell/show you

are there tools that can go a step further in "solving" build issues?

what is the best path forward when a core library you rely on has a known bug? Work around it? What if they fix the bug?

Ryan: I had heard developers complain that measuring progress by lines of code written is a bad way to track how much each developer contributed, and now I know why. Sometimes getting 10 lines of code to function correctly can be harder than writing 400 lines of something else. Just like the real-world, it's hard to convey this at times.

Spent over a week just getting Joinery to build so we could even work on it

Spent over a week attempting to use DataFrame.add() method which treats lists differently depending on JDK version (as a single object versus iterating over it)