Server

- Backend: http://ec2-3-138-116-248.us-east-2.compute.amazonaws.com:5000
- Frontend: http://ec2-3-138-116-248.us-east-2.compute.amazonaws.com:8000
- Currently serving: http://augur.chaoss.io/api/unstable

Problem

- Need: Extend Augur's new frontend and update the getting started document.
- Purpose: Users can view more health metrics about a project and a clearer getting started document of Augur.
- Client Base: Students, developers, researchers, and data scientists.

Deliverable

- Show the number of pull requests on Augur's new frontend.
- Show visualization about committers, default branches, forks, stars, and watchers on Augur's new frontend.
- Make an outline for what we plan to change in the getting started document.

Diagrams

- Create design diagrams like the ones from class. Our first step is to find a software program that we can use to draw these diagrams: https://www.diagrams.net/
- Each individual contributes to all diagrams, through independent design, design collaboration, and careful peer review of finished diagrams.
- Compile the diagrams into the group document.
- Each diagram should have a title and caption describing what it shows.

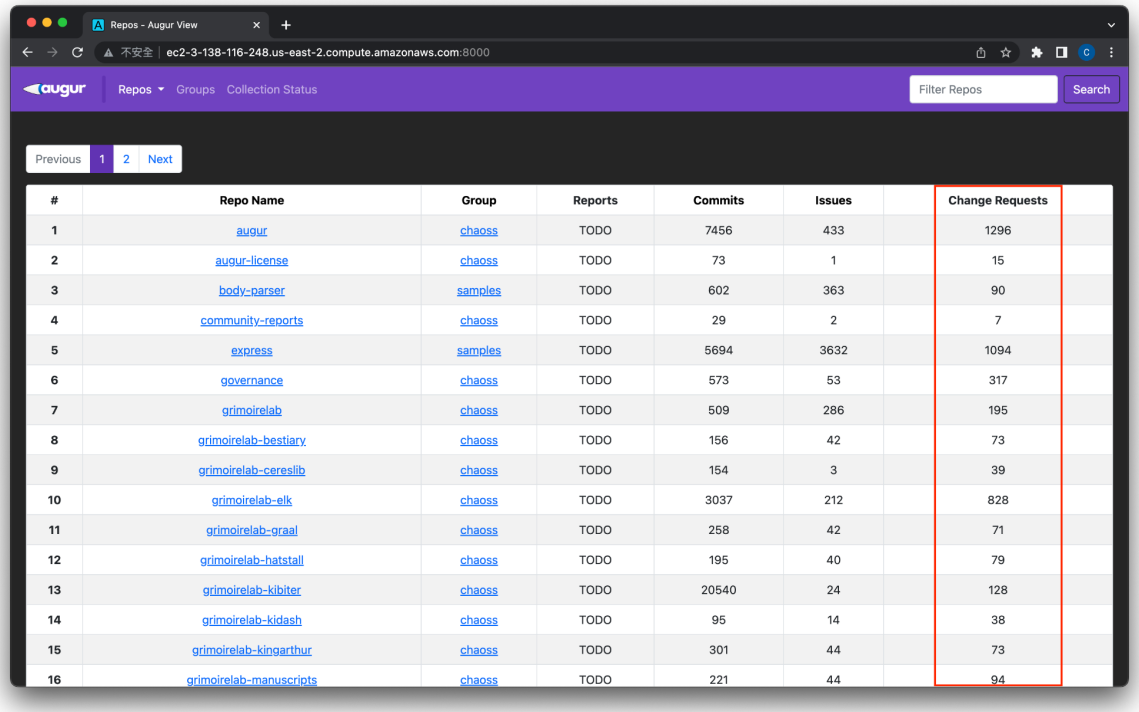
System Constraints

- Web server software.
- Database management system.
- Physical servers with compatible OS installed to host the webserver and DBMS with network and internet connectivity.
- Client computers with a web browser and Internet access.

Users / Activities / Relevant Data / Constraints

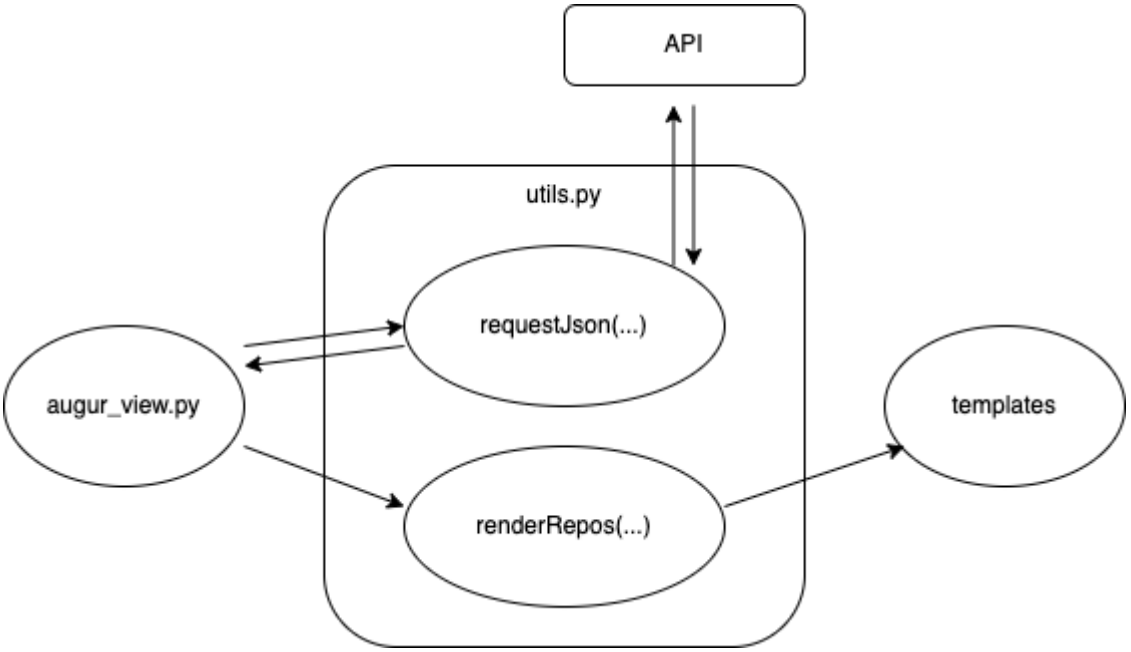
- Users can view Augur's backend.
- Users can view API about repositories.
- Users can view API about groups.
- Users can view API about repo_info.
- Users can view Augur's new frontend.
- Users can view the hello world page.
- Users can view the number of pull requests on the repos-table page.
- Users can view visualization about committers, default branches, forks, stars, and watchers on the collection status page.
- Users can view a clearer getting started document about Augur.
- Developers should manually test new features, including API, GUI, and check for updates to the getting started document.

Hello World version of our project - Show the number of pull requests on the frontend



#	Repo Name	Group	Reports	Commits	Issues	Change Requests
1	augur	chaoss	TODO	7456	433	1296
2	augur-license	chaoss	TODO	73	1	15
3	body-parser	samples	TODO	602	363	90
4	community-reports	chaoss	TODO	29	2	7
5	express	samples	TODO	5694	3632	1094
6	governance	chaoss	TODO	573	53	317
7	grimoirelab	chaoss	TODO	509	286	195
8	grimoirelab-bestiary	chaoss	TODO	156	42	73
9	grimoirelab-cereslib	chaoss	TODO	154	3	39
10	grimoirelab-elk	chaoss	TODO	3037	212	828
11	grimoirelab-graal	chaoss	TODO	258	42	71
12	grimoirelab-hatstall	chaoss	TODO	195	40	79
13	grimoirelab-kibiter	chaoss	TODO	20540	24	128
14	grimoirelab-kidash	chaoss	TODO	95	14	38
15	grimoirelab-kingarthur	chaoss	TODO	301	44	73
16	grimoirelab-manuscripts	chaoss	TODO	221	44	94

Wireframes - Build a full working model of our entire project



Development

- Get API data using the requestJson("metadata/repo_info") function.
- Store the information about repositories and feed them to templates.
- Render the data on templates.
- Displaying other health and safety metrics can be done in similar steps, but using different APIs.

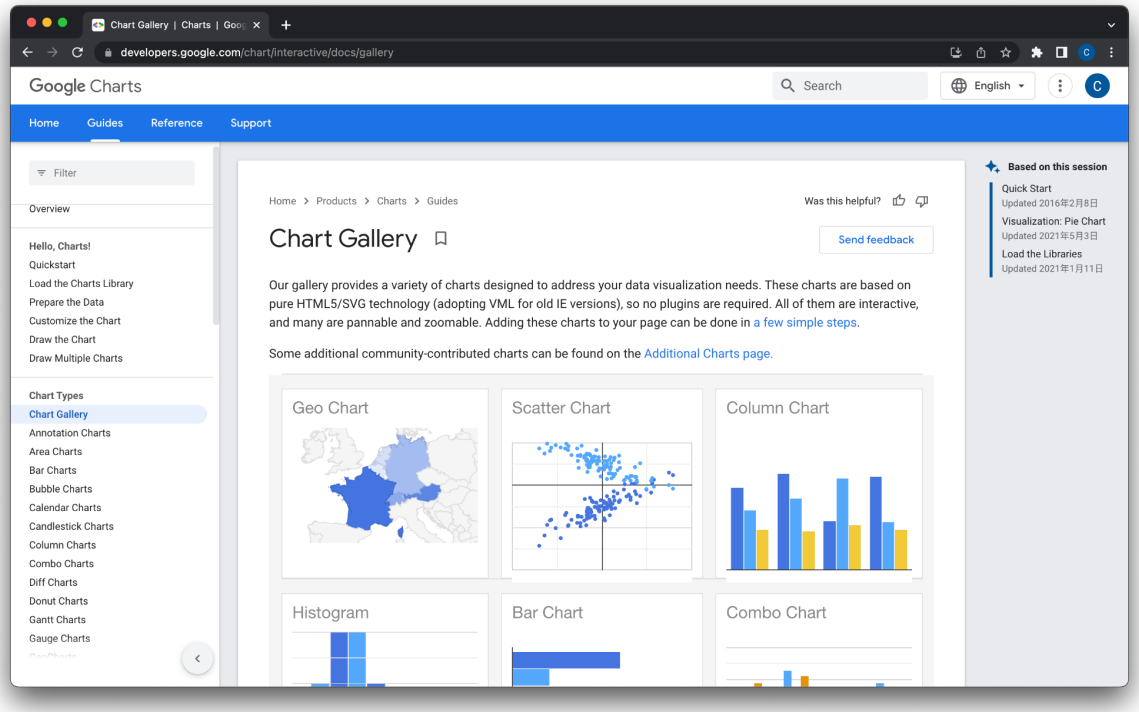
Stubs - Show the number of pull requests on the frontend

Repo Name	Group	Reports	Commits	Issues	Change Requests
Augur	chaoss	TODO	7459	433	1296
Augur-license	chaoss	TODO	73	1	15
Body-parser	samples	TODO	602	363	90

All classes with method signatures named - Hello World example

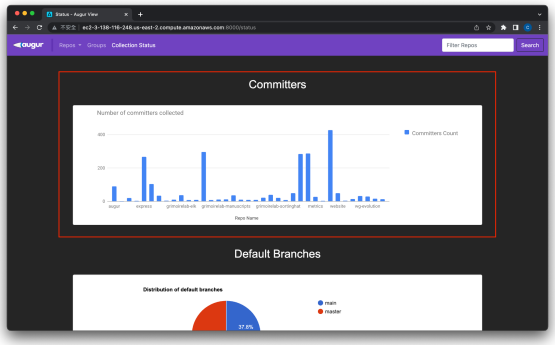
```
"""
Hello World example:
    This is an example demonstrating the use of modules.
"""
@app.route('/hello-world')
def hello_world():
    return render_module("hello-world", title="Hello, World!")
```

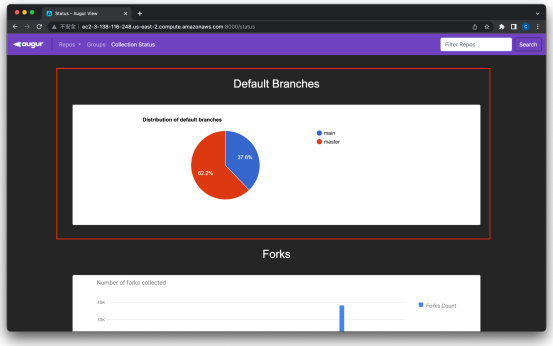
Use google charts API to get data visualization - Users can view visualization about committers, default branches, forks, stars, and watchers on the collection status page

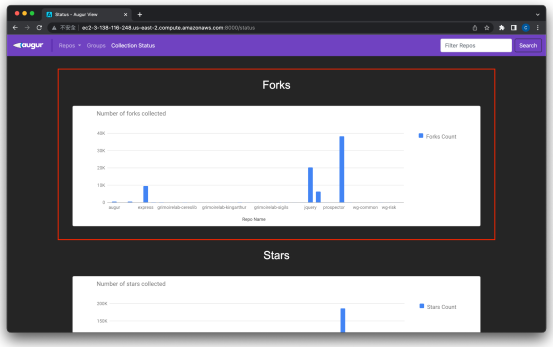


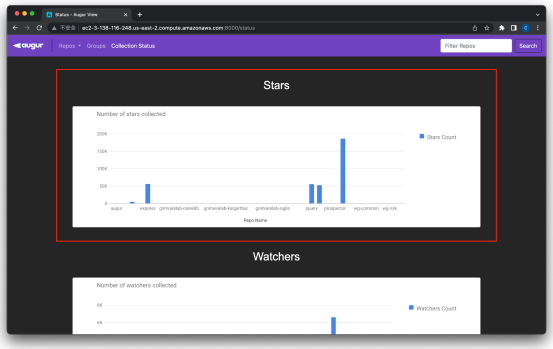

```
var table = new google.visualization.Table(document.getElementById('table_div'));
table.draw(data, {showRowNumber: true, width: '100%', height: '100%'});
google.visualization.events.addListener(table, 'select', function() { … });
```

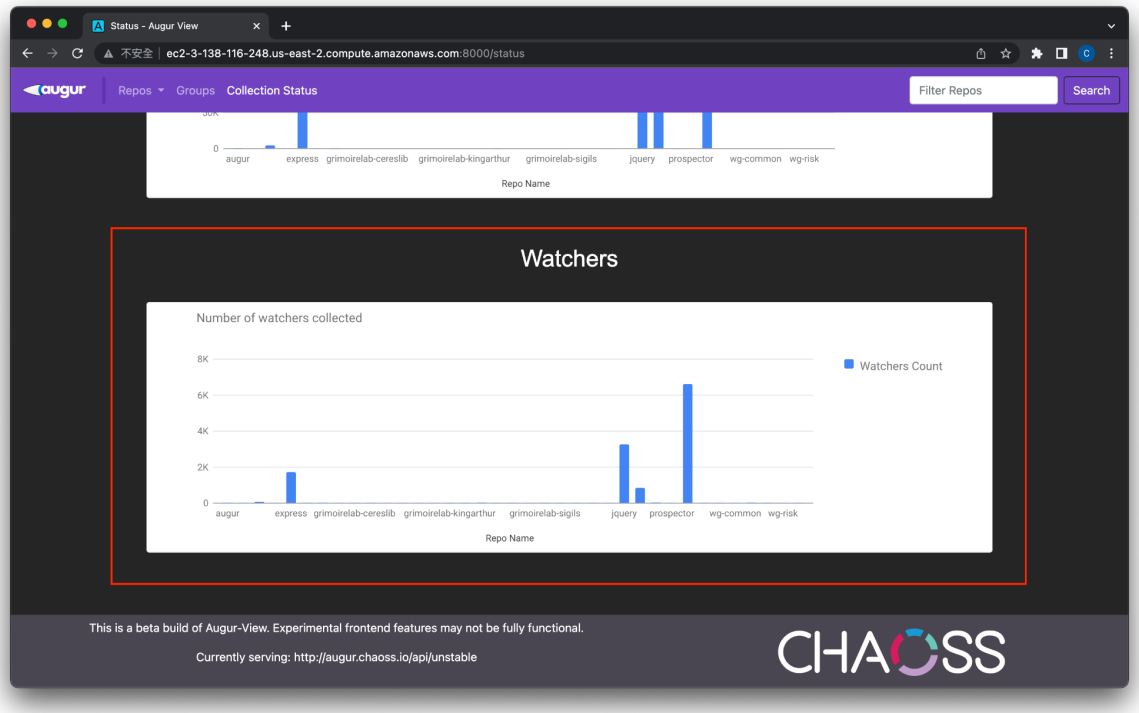
Between 20% and 50% of all the code is working - Show visualization about committers, default branches, forks, stars, and watchers on Augur's new frontend









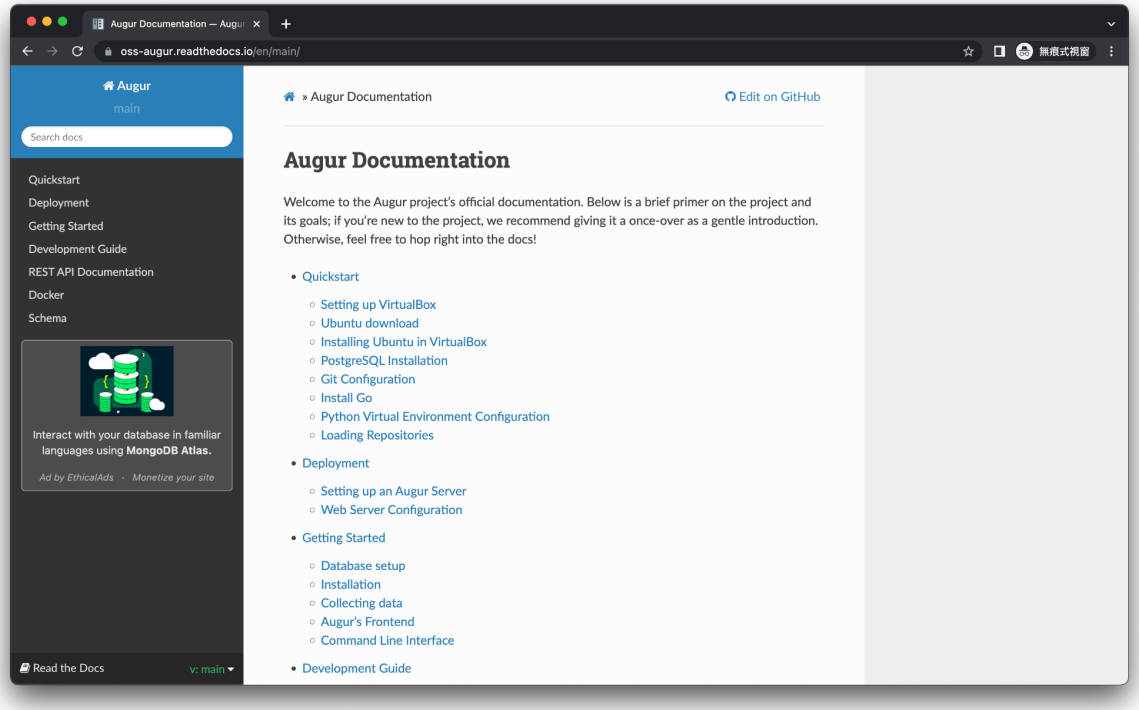


Update the getting started document

For setting up the DevOps environment, the instructions use Ubuntu 20.4 rather than the 18x version which is no longer supported. Ubuntu 20.4 is preferred for the PostgreSQL Installation because it has long-term support. It is also usable for the 18x version which makes it even more available. For the Git Configuration, every platform should use the command line login to cache the Git Credentials for the LINUX user who is operating Augur. By performing this step, the Facade Commit Counting Diesel is prevented from stalling the command line prompt when the repositories move or disappear.

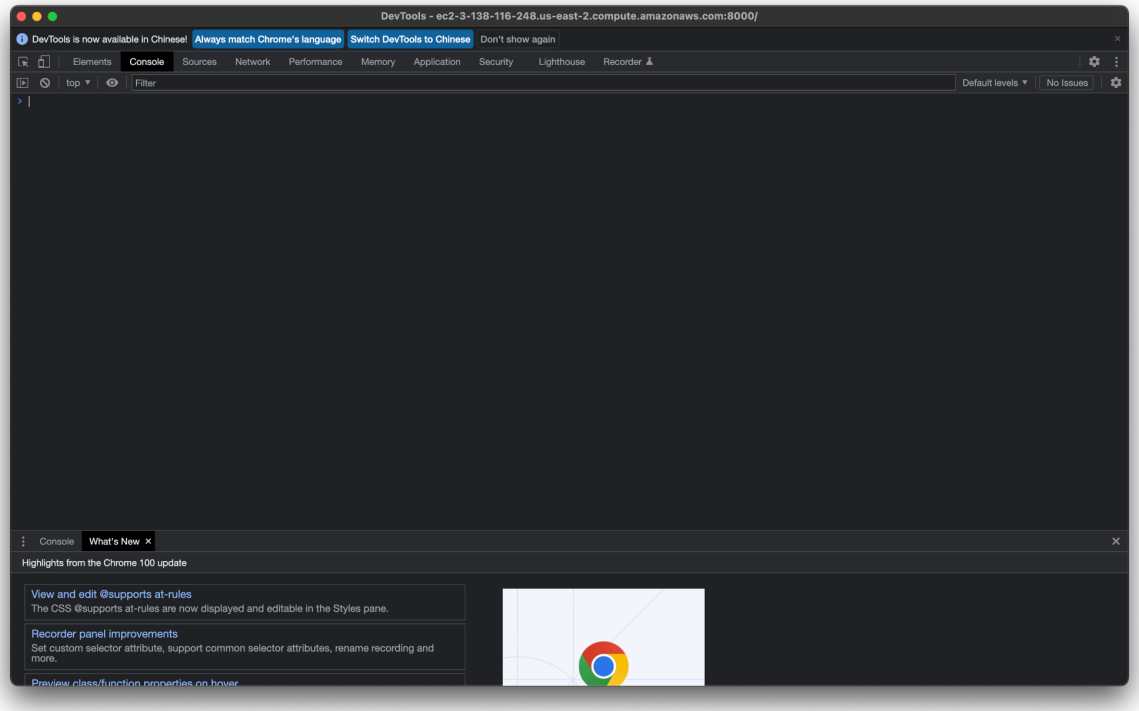
The limitation for the Python Virtual Environment Configuration is that the software version of the python should be more than 3.8, or else it won't be supported. Since the Augur installation is only done on the macOS, Ubuntu, and Fedora, it does not support Windows directly. For the installation in Windows, the use of docker images is required. Otherwise, the user should install the virtual machine with the supported operating system. For the installation from the source, PostgreSQL should be set up for the storage of the data collected by Augur. After that, the Augur application server should be installed and configured. The Augur data that collected workers should be installed and configured.

For the deployment of the Augur's frontend, we must first run the Augur config init-frontend with the python environment. The user has to enter Augur's home directory and run the npm install and then the npm run build in the frontend directory. The good thing is that the command line interface is provided in the Augur for interaction with the Augur installation.

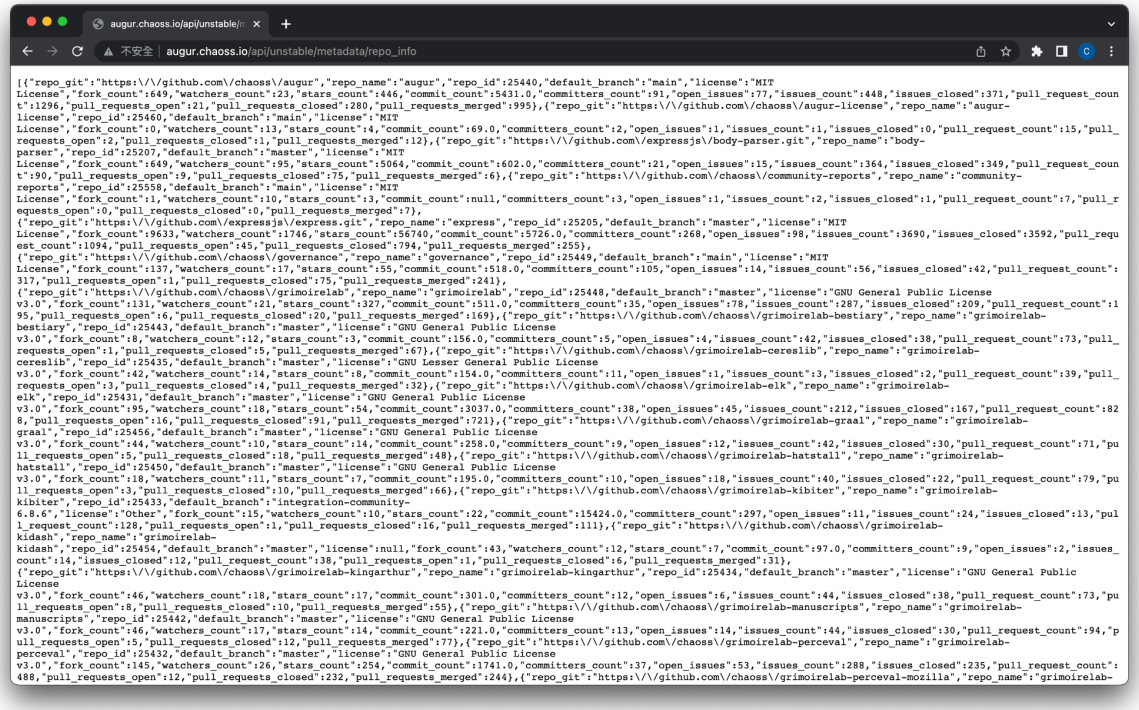


Testing Plan - Extend Augur's new frontend

1. Check that Augur's new frontend displays new features correctly.
2. Use the browser's DevTools to check the error messages in the console.

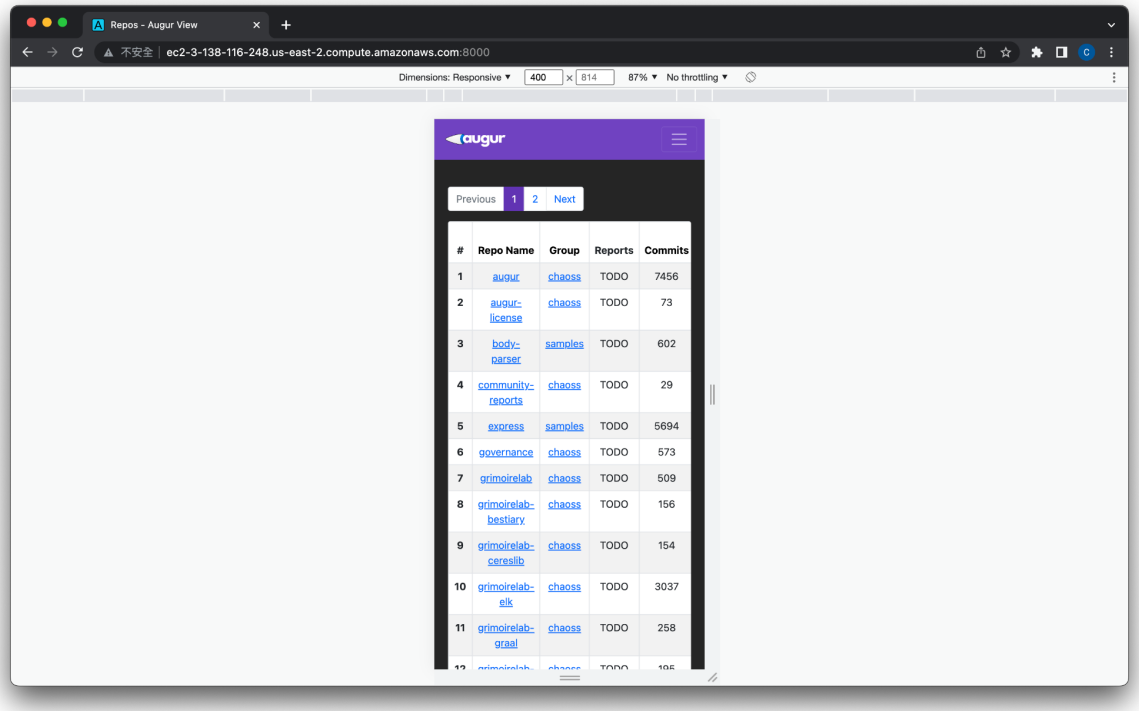


3. Check that the results displayed on Augur's new frontend are consistent with the data from Augur's API.



[{"repo_git":"https:\/\/github.com\/chaoss\/augur","repo_name":"augur","repo_id":25440,"default_branch":"main","license":"MIT License","fork_count":649,"watchers_count":23,"stars_count":446,"commit_count":15431.0,"committers_count":91,"open_issues":77,"issues_count":448,"issues_closed":371,"pull_request_count":1296,"pull_requests_open":21,"pull_requests_closed":280,"pull_requests_merged":995},{"repo_git":"https:\/\/github.com\/chaoss\/augur-license","repo_name":"augur-license","repo_id":25460,"default_branch":"main","license":"MIT License","fork_count":0,"watchers_count":13,"stars_count":4,"commit_count":69.0,"committers_count":2,"open_issues":1,"issues_count":1,"issues_closed":0,"pull_request_count":15,"pull_requests_open":2,"pull_requests_closed":1,"pull_requests_merged":12},{"repo_git":"https:\/\/github.com\/expressjs\/body-parser.git","repo_name":"body-parser","repo_id":25207,"default_branch":"master","license":"MIT License","fork_count":649,"watchers_count":95,"stars_count":5064,"commit_count":602.0,"committers_count":21,"open_issues":15,"issues_count":364,"issues_closed":349,"pull_request_count":90,"pull_requests_open":9,"pull_requests_closed":75,"pull_requests_merged":6},{"repo_git":"https:\/\/github.com\/chaoss\/community-reports","repo_name":"community-reports","repo_id":25558,"default_branch":"main","license":"MIT License","fork_count":1,"watchers_count":10,"stars_count":3,"commit_count":null,"committers_count":3,"open_issues":1,"issues_count":2,"issues_closed":1,"pull_request_count":7,"pull_requests_open":0,"pull_requests_closed":0,"pull_requests_merged":7},
{"repo_git":"https:\/\/github.com\/expressjs\/express.git","repo_name":"express","repo_id":25205,"default_branch":"master","license":"MIT License","fork_count":9633,"watchers_count":1746,"stars_count":56740,"commit_count":5726.0,"committers_count":268,"open_issues":98,"issues_count":3690,"issues_closed":3592,"pull_request_count":1094,"pull_requests_open":45,"pull_requests_closed":794,"pull_requests_merged":255},
{"repo_git":"https:\/\/github.com\/chaoss\/governance","repo_name":"governance","repo_id":25449,"default_branch":"main","license":"MIT License","fork_count":137,"watchers_count":17,"stars_count":55,"commit_count":518.0,"committers_count":105,"open_issues":14,"issues_count":56,"issues_closed":42,"pull_request_count":317,"pull_requests_open":1,"pull_requests_closed":75,"pull_requests_merged":241},
{"repo_git":"https:\/\/github.com\/chaoss\/grimoirelab","repo_name":"grimoirelab","repo_id":25448,"default_branch":"master","license":"GNU General Public License v3.0","fork_count":131,"watchers_count":21,"stars_count":327,"commit_count":511.0,"committers_count":35,"open_issues":78,"issues_count":287,"issues_closed":209,"pull_request_count":195,"pull_requests_open":6,"pull_requests_closed":20,"pull_requests_merged":169},{"repo_git":"https:\/\/github.com\/chaoss\/grimoirelab-bestiary","repo_name":"grimoirelab-bestiary","repo_id":25443,"default_branch":"master","license":"GNU General Public License v3.0","fork_count":8,"watchers_count":12,"stars_count":3,"commit_count":156.0,"committers_count":5,"open_issues":4,"issues_count":42,"issues_closed":38,"pull_request_count":73,"pull_requests_open":1,"pull_requests_closed":5,"pull_requests_merged":67},{"repo_git":"https:\/\/github.com\/chaoss\/grimoirelab-cereslib","repo_name":"grimoirelab-cereslib","repo_id":25435,"default_branch":"master","license":"GNU Lesser General Public License v3.0","fork_count":42,"watchers_count":14,"stars_count":8,"commit_count":154.0,"committers_count":11,"open_issues":1,"issues_count":3,"issues_closed":2,"pull_request_count":39,"pull_requests_open":3,"pull_requests_closed":4,"pull_requests_merged":32},{"repo_git":"https:\/\/github.com\/chaoss\/grimoirelab-elk","repo_name":"grimoirelab-elk","repo_id":25431,"default_branch":"master","license":"GNU General Public License v3.0","fork_count":95,"watchers_count":18,"stars_count":54,"commit_count":3037.0,"committers_count":38,"open_issues":45,"issues_count":212,"issues_closed":167,"pull_request_count":828,"pull_requests_open":16,"pull_requests_closed":91,"pull_requests_merged":721},{"repo_git":"https:\/\/github.com\/chaoss\/grimoirelab-graal","repo_name":"grimoirelab-graal","repo_id":25456,"default_branch":"master","license":"GNU General Public License v3.0","fork_count":44,"watchers_count":10,"stars_count":14,"commit_count":258.0,"committers_count":9,"open_issues":12,"issues_count":42,"issues_closed":30,"pull_request_count":71,"pull_requests_open":5,"pull_requests_closed":18,"pull_requests_merged":48},{"repo_git":"https:\/\/github.com\/chaoss\/grimoirelab-hatstall","repo_name":"grimoirelab-hatstall","repo_id":25450,"default_branch":"master","license":"GNU General Public License v3.0","fork_count":18,"watchers_count":11,"stars_count":7,"commit_count":195.0,"committers_count":10,"open_issues":18,"issues_count":40,"issues_closed":22,"pull_request_count":79,"pull_requests_open":3,"pull_requests_closed":10,"pull_requests_merged":66},{"repo_git":"https:\/\/github.com\/chaoss\/grimoirelab-kibiter","repo_name":"grimoirelab-kibiter","repo_id":25433,"default_branch":"integration-community-6.8.6","license":"Other","fork_count":15,"watchers_count":10,"stars_count":22,"commit_count":15424.0,"committers_count":297,"open_issues":11,"issues_count":24,"issues_closed":13,"pull_request_count":128,"pull_requests_open":1,"pull_requests_closed":16,"pull_requests_merged":111},{"repo_git":"https:\/\/github.com\/chaoss\/grimoirelab-kidash","repo_name":"grimoirelab-kidash","repo_id":25454,"default_branch":"master","license":null,"fork_count":43,"watchers_count":12,"stars_count":7,"commit_count":97.0,"committers_count":9,"open_issues":2,"issues_count":14,"issues_closed":12,"pull_request_count":38,"pull_requests_open":1,"pull_requests_closed":6,"pull_requests_merged":31},
{"repo_git":"https:\/\/github.com\/chaoss\/grimoirelab-kingarthur","repo_name":"grimoirelab-kingarthur","repo_id":25434,"default_branch":"master","license":"GNU General Public License v3.0","fork_count":46,"watchers_count":18,"stars_count":17,"commit_count":301.0,"committers_count":12,"open_issues":6,"issues_count":44,"issues_closed":38,"pull_request_count":73,"pull_requests_open":8,"pull_requests_closed":10,"pull_requests_merged":55},{"repo_git":"https:\/\/github.com\/chaoss\/grimoirelab-manuscripts","repo_name":"grimoirelab-manuscripts","repo_id":25442,"default_branch":"master","license":"GNU General Public License v3.0","fork_count":46,"watchers_count":17,"stars_count":14,"commit_count":221.0,"committers_count":13,"open_issues":14,"issues_count":44,"issues_closed":30,"pull_request_count":94,"pull_requests_open":5,"pull_requests_closed":12,"pull_requests_merged":77},{"repo_git":"https:\/\/github.com\/chaoss\/grimoirelab-perceval","repo_name":"grimoirelab-perceval","repo_id":25432,"default_branch":"master","license":"GNU General Public License v3.0","fork_count":145,"watchers_count":26,"stars_count":254,"commit_count":1741.0,"committers_count":37,"open_issues":53,"issues_count":288,"issues_closed":235,"pull_request_count":488,"pull_requests_open":12,"pull_requests_closed":232,"pull_requests_merged":244},{"repo_git":"https:\/\/github.com\/chaoss\/grimoirelab-perceval-mozilla","repo_name":"grimoirelab-

4. Ensure that new features do not affect existing functions.
5. Check if it is displayed correctly on a variety of devices, including smartphones, tablets, and computers.



6. Check if it is displayed correctly on various browsers, such as Google, Firefox, Safari, Edge, etc.
7. Peer review to ensure that the development approach, including feature usage, code reuse, and coding style is consistent with the original project.
8. If time is available, write test cases for unit and integration tests for the project to ensure test coverage of 70% or more.
9. If time is available, deploy the project to Jenkins for continuous integration and continuous delivery.
10. If time is available, use SonarQube to check the code smells of the project.

Testing Plan - Update the getting started document

1. Understand all the contents of the original document.
2. Check the spelling and grammar to review any ambiguity or inconsistency between what functionality it performs and what it is supposed to do.
3. Follow the steps in the new document to actually install Augur once and check if it can be installed successfully.
4. Show the new document to people outside of our team members to ensure they can understand it.
5. Examples will be needed in case of any problem that occurs to the user, particularly novice users who may check the documentation for any confusion.
6. If time is available, use defect reporting tools and tracking tools.