Async SQL Adapter and Actix-web middleware for Casbin-rs

– GSoC 2020 Proposal

PERSONAL INFORMATION

First/Last Name: Yi sheng Chai

Email: hackerchai.com@gmail.com

School/University: Chongqing University

Graduation Date: June, 2022

Major/Focus: Electrical Engineering and Automation

Location: Chongqing, China

Timezone: China Standard Time (CST), UTC +8

Github Profile: https://github.com/hackerchai

IRC: easonchai

CV/Resume URL: https://github.com/hackerchai/resume/en-US.pdf

Preferred time of day for virtual/video interview: 7:00-24:00 UTC+8

Open Source Experience:

- Rap-ID provides an algorithm which uses less user inputs but provides even more security and features
- Android App development at OpeingSourceORG
- Software Mirror maintenance at CQU Mirror
- Personal open-source project: docker-rust-web, Magisk-Revoke-China-Certs, soccer_dog_reminder, Canteen-Helper-Backend

GSOC INFORMATION

Have you participated in Google Summer of Code in the past?

No.

Have you applied to GSoC in the past?

No.

Are you applying to any other organizations this year?

Yes.

How many hours will you devote to your GSoC project each week?

I plan to work a full 8 hour work day every weekday.

What are your other summer plans?

No prior commitments as of now. If I'm selected for GSoC, it will be my only job this summer.

TECHNICAL INFORMATION

Have you ever worked with Git?

Yes, I'm fairly proficient with Git. For most of my assignments/projects, I prefer to use Git. So, I believe I have a decent knowledge of Git.

Have you ever utilized IRC?

Yes. I use IRC a lot. IRC is my preferred mode of communication. I tend to resolve majority of my issues and questions on IRC. Also Riot and Telegram are also options.

Have you completed reading source-code of Casbin-rs ?

Yes. I've started reading the source-code of Casbin-rs and had some improvements thought of the project.

Have you ever contributed code to Casbin?

Yes. Though, I've started contributing to Casbin-rs fairly recently. My github username is Hackerchai

- Details of my contributions to Casbin-rs(sqlx-adapter) can be seen here:

  My repo is https://github.com/hackerchai/sqlx-adapter and my pull request is https://github.com/casbin-rs/sqlx-adapter/pull/1

- Details of my contributions to Casbin-rs(diesel-adapter) can be seen here:

  My pull request is https://github.com/casbin-rs/diesel-adapter/pull/18

Do you plan to continue contributing to the Casbin project after GSoC is finished?

I definitely plan to contribute to Casbin even after GSoC. In fact, I've even the features on which I'd like to work immediately after completion of my project towards the end of this proposal, under the heading 'Future Work'.

What can you do?

I have the experience of using Rust for web-backend development. I have used the actix-web framework and diesel middleware. I am able to use MySQL/PostgreSQL proficiently. Besides, I have development experience with docker and CI.

PROJECT INFORMATION

Which project idea sparks your interest and why?

Project : Casbin for Rust interests me most. A lot of rustacean (including myself) really needs an authorization library for web-development. And casbin-rs is an authorization library that supports access control models like ACL, RBAC, ABAC in Rust. Being a developer also, I have used casbin-php in my previous project and it feels smooth and easy to configure. But the casbin-rs haven't come into officially released due to some unfinished features. With the passion of Rust language, I am so willing to accomplish some urgent features. As is known to all that, the asynchronous feature has been becoming a future trending in backend development. But the default database adapter of casbin-rs (diesel) does not support the async operation on database. Besides, the actix-web which is very popular among the rust web-frameworks also needs a authorization middleware, so implementing an actix-actor and middleware is also of important.

Project Description

1. Implement a fully asynchronous adapter: sqlx-adapter using sqlx.
2. Actix actor and actix-web middleware using **RBAC with domain**

Link to ideas page: https://github.com/casbin/SummerOfCode2020

Problem

1. casbin-rs default database adapter doesn't support async database operation.
2. casbin-rs has no actix-embedded support.

Solution

1. Add an new adapter sqlx-adapter using the library sqlx. Async-std and tokio are both supported by the adapter.
2. Write an actix actor for accepting messages of rbac request as well as the actix-web middleware which reads JWT token from HTTP authorization header (To make it simple, JWT token = .) and sends rbac request (subject, domain, object, action) to actix actor for authorization. The middleware will block the request if actix actor denied the request.

Brief description of major parts

- **SQLX adapter**: The adapter will basically provide rbac rules queries support which is built from the ground-up using async/await for maximum concurrency. For security guarantee, The Postgres and MySQL/MariaDB drivers are written in pure Rust using zero unsafe code.
- **Actix actor for rbac request**: In the actix system, actors are objects which encapsulate state and behavior and run within the *Actor System* provided by the actix library. We built this actor for handling the rbac request to return enforcing message.
- **Actix-web middleware**: It will provide a basic middleware to reads JWT token from HTTP authorization header (To make it simple, JWT token = .) and sends rbac request (subject, domain, object, action) to actix actor for authorization. The middleware will block the request if actix actor denied the request.

Difficulty: Medium hard

Deliverables

- Sqlx adapter finish - June 7
- Actix actor finish - July 28
- Actix-web middleware finish - Aug 11

Timeline

May 1- May 23:(about four weeks)

- Learn more about Casbin Community, and discussions with the mentor and community on its interpretation and how it might best be implemented. Collect known issues and improvement advices from community.

- Before May 7: Planning of adapter structure and learn async operation. Design the sqlx adapter project, determine mod design and make sql query improvement.

- Before May 14: Planning of actor structure and learn Actix documents. Design the Actix Actor logic, design Actix Actor structure and start writing documents.

- Before May 21: Planning of Actix middleware. Design the Actix middleware logic, draw architectural diagrams. Start to write documents for implementation.

- Get more feedback from mentors and make a consensus to the project plan.

May 23 - June 7:(two weeks)

- Start to implement the sqlx adapter for casbin-rs. (It's already in progress in https://github.com/hackerchai/sqlx-adapter)

- Find a proper way to rewrite casbin-rs error handle. And then pull request.

- Implement transaction SQL query for reliable database operation.
- Finish full async functions support.
- Unit Test for sqlx-adapter. Then fix the bugs and finish improvements.

MILESTONE: The sqlx adapter can return the correct function or functions with fully async support . (Mid-term Submission)

July 14 - July 28:(two weeks)

- Start to implement Actix actor for casbin.
- Implement async support, using Futures to handle asynchronous message handling.
- Desgin interface for casbin actor which will be easy to used.
- Unit Test for Actix Actor.

July 28 - August 11:(About three weeks)

- Start to implement the Actix middleware.
- Design the logic of JWT authentication which can receive username and domain from request header token.
- Combine with the Actix Actor to handle RBAC request, implement the verification route.
- Support cache(eg.redis) to store the policies in memory to accerelate the proccess.
- Unit Test for Actix middleware with bug fixes.

MILESTONE: At this point, The most part of job have been done. Next, we should do some evaluations about it.

August 11 - August 18:(one week)

- Polish the code again and see what we left behind. And improve the way we handle the data and error.
- CI integration and document works. Polish the document finished in May and make it clear to developer.
- Example code for three modules. Add them into the documents.

- Get feedbacks from mentors and final code review.

August 18 - August 31:(about two week)

- Ensuring the code is integrated and made available on Github and in the crates.io.
- Consult with the mentors for advices and polish the code.
- Wrap up the crates.
- Submit code, proposal and evaluation to mentors.

Future Work (to be done after completion of this project)

First and foremost, I'll provide support for the developed module, because I believe providing support for a module is as important as developing it.

Apart from that, following is something I'd like to work upon after the completion of this project.

So, Sqlite adapter will be a future feature for someone may like the light-weight database.

How will deal with project, task, and time management? Will you utilize software? If so, which tool and why?

I will use a project management system like Trello to manage tasks and deadlines both. I prefer Trello because I've worked on Trello before and the fact that it provides very good interface for adding tasks and their deadlines. This way my mentors can track my progress in a fine grained manner.

How often will you communicate with mentor? How will you communicate with mentors?

I'll have two meetings per week with my mentor. Upon discussion with my mentors, we've decided to have a wechat video Monday and Thursday of every week. It will typically be a 15-20 minutes discussion mainly to catch up with my progress. We'll have more regular discussions on Gitter or wechat itself.

BIO OF ME

I am Yi sheng Chai (Eason Chai), I had learned **2 years of Rust** before and have finished a websocket push client in the computer club of my university. Recently, I am working on a mirror sync tool writing in rust for our school's open-source software mirror

server. Before I get familiar with Casbin I have already used the **Actix-web framework** for backend development. And I have **5 years of Java experience with Android and Spring** and **1 year PHP experience**.

I have prepared for the GSOC since last year. And I have finished my internship in Beijing Pangju inc. I am responsible for the Android APP development and test. I have experience of developing complex software and cooperating with other developers. I am familiar with the technologies such as **Docker, Serverless and CI**. I have the ability of learning new technologies within a short time. And I have a habit of sharing technologies with blog posts (My blog: https://blog.hackerchai.com), I am happy to help people with technologies.

Besides, I am an **open-source lover**. I can use Linux and git proficiently. I started the Linux User Group in Chongqing University. And now I am the main maintainer of the CQU Software Mirror which is used widely among Chinese linux user in southern part. I have contributed to the OpeingSourceORG with the Android app. I am very enthusiastic to contribute myself to the casbin community.