考えながらクライアントサイドのウェブアプリケーションを作る話

自己紹介

- Name : **azu**
- Twitter : @azu_re
- Website: Web scratch, JSer.info

このスライドは

Faao - ドメイン駆動設計で作るGitHub Issue Client -

を改定したものです。

過去に作ったやつ

- azu/GithubReader: Github Notifications Client for OS X
- azu/github-reader: [node-webkit] GitHub client app - Viewer for Notifications and News Feed.
- azu/github-issue-teev: [NW.js] GitHub Issue Manager(Viewer)

2014-04-29T13:12:09Z
MartinKolarik
MartinKolarik commented on ractivejs/ractive#644

@jameshfisher Yeah that makes sense, but Ractive's isn't a library for input validation. If you need just a simple validation, you can use Ractive's features (observers) to create some validation rules, if you need something more complex, you are free to use any other library/plugin intended for that purpose.

2014-04-29T13:10:41Z
MindTooth
MindTooth commented on mllrsohn/node-webkit-builder#9

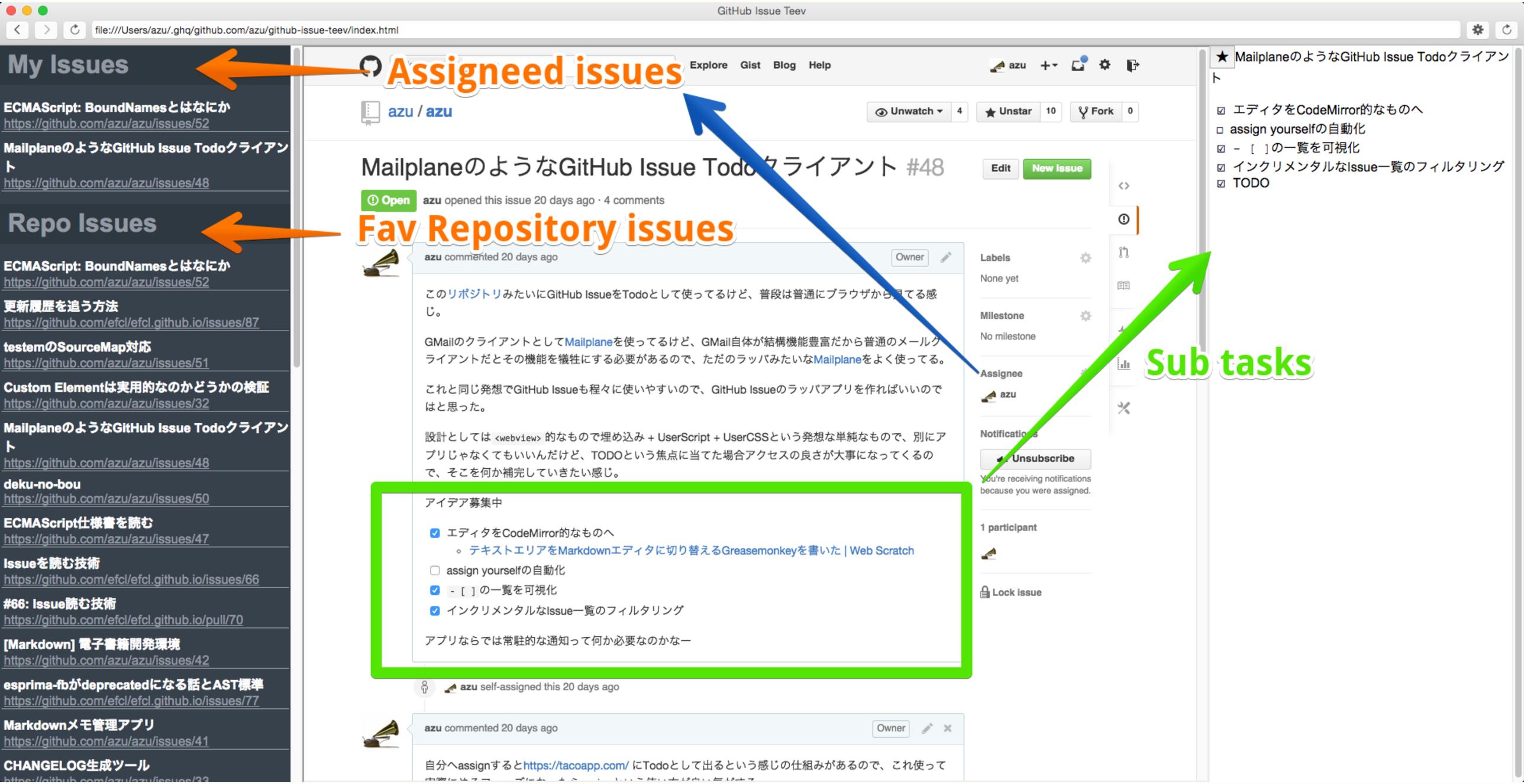
Well, this is the current file: https://gist.github.com/MindTooth/11382714 Also added the `package.json` file within the same Gist. ![screen shot 2014-04-29 at 15 07 50](https://cloud.githubusercontent.com/assets/35828/2829159/57375d38-cf9f-11e3-864b-c0220782b4ba.png) Just

2014-04-29T13:08:41Z
hpique
hpique commented on steipete/PSPDFTextView#9

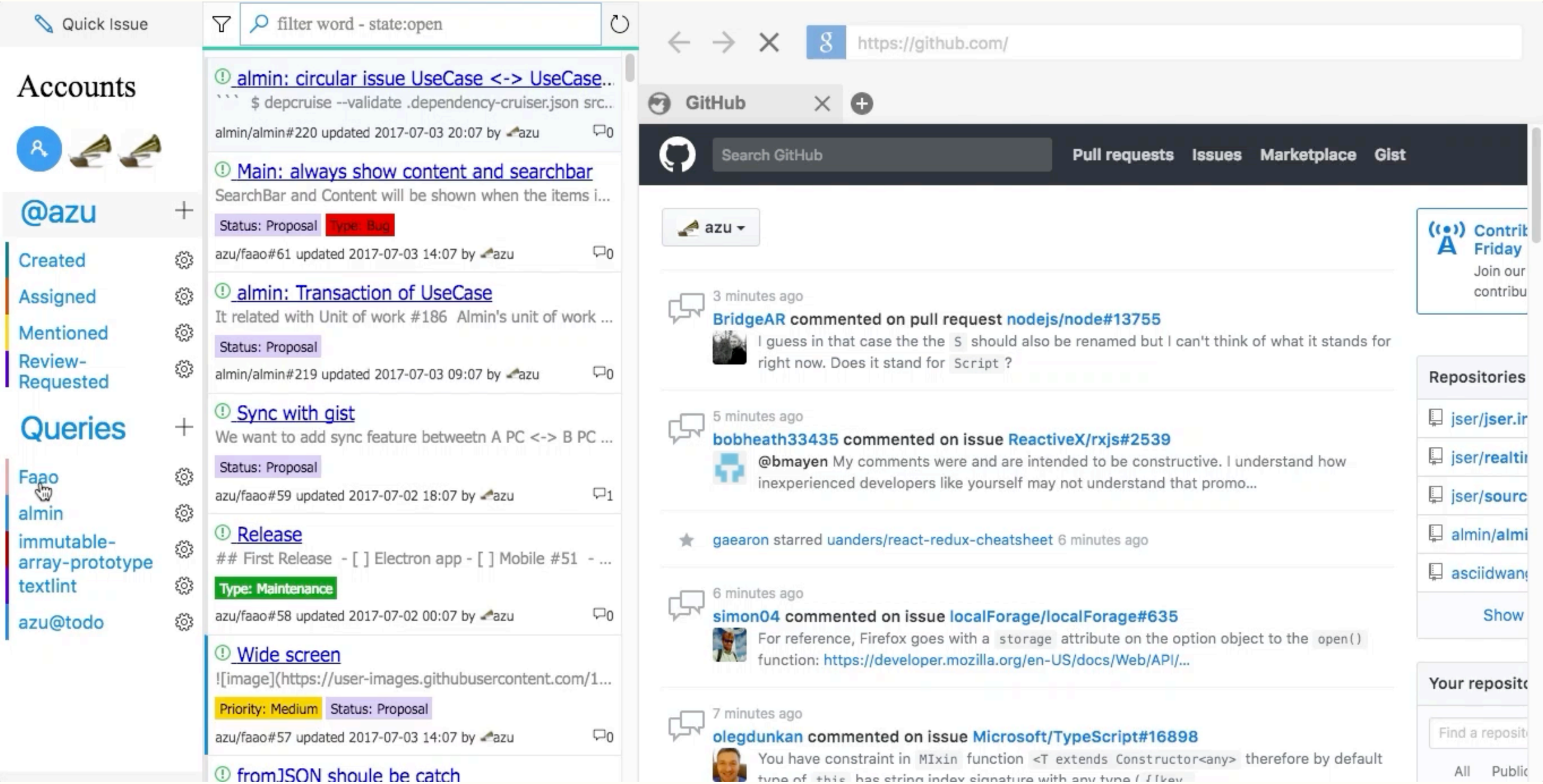
Awesome. Thanks!

2014-04-29T13:07:39Z
reubano
reubano commented on travis-ci/travis.rb#41

This is what worked for me `travis-enc.sh` ```bash #!/usr/bin/env sh -u ENC_FILE='envs.yml' ENVS=$1 USER=$2 PROJECT=$3 encrypt_file () { secret=$1 file=$2


GitHub Issue Teev
file:///Users/azu/.ghq/github.com/azu/github-issue-teev/index.html
My Issues
ECMAScript: BoundNamesとはなにか
https://github.com/azu/azu/issues/52
MailplaneのようなGitHub Issue Todoクライアント
https://github.com/azu/azu/issues/48
Repo Issues
ECMAScript: BoundNamesとはなにか
https://github.com/azu/azu/issues/52
更新履歴を追う方法
https://github.com/efcl/efcl.github.io/issues/87
testemのSourceMap対応
https://github.com/azu/azu/issues/51
Custom Elementは実用的なのかどうかの検証
https://github.com/azu/azu/issues/32
MailplaneのようなGitHub Issue Todoクライアント
https://github.com/azu/azu/issues/48
deku-no-bou
https://github.com/azu/azu/issues/50
ECMAScript仕様書を読む
https://github.com/azu/azu/issues/47
Issueを読む技術
https://github.com/efcl/efcl.github.io/issues/66
#66: Issue読む技術
https://github.com/efcl/efcl.github.io/pull/70
[Markdown] 電子書籍開発環境
https://github.com/azu/azu/issues/42
esprima-fbがdeprecatedになる話とAST標準
https://github.com/efcl/efcl.github.io/issues/77
Markdownメモ管理アプリ
https://github.com/azu/azu/issues/41
CHANGELOG生成ツール
Assigneed issues
Fav Repository issues
Explore Gist Blog Help
azu
azu / azu
Unwatch 4
Unstar 10
Fork 0
MailplaneのようなGitHub Issue Todoクライアント #48
Edit
New Issue
Open
azu opened this issue 20 days ago · 4 comments
azu commented 20 days ago
Owner
このリポジトリみたいにGitHub IssueをTodoとして使ってるけど、普段は普通にブラウザから見てる感じ。
GMailのクライアントとしてMailplaneを使ってるけど、GMail自体が結構機能豊富だから普通のメールクライアントだとその機能を犠牲にする必要があるので、ただのラッパみたいなMailplaneをよく使ってる。
これと同じ発想でGitHub Issueも程々に使いやすいので、GitHub Issueのラッパアプリを作ればいいのではと思った。
設計としては <webview> 的なもので埋め込み + UserScript + UserCSSという発想な単純なもので、別にアプリじゃなくてもいいんだけど、TODOという焦点に当てた場合アクセスの良さが大事になってくるので、そこを何か補完していきたい感じ。
アイデア募集中
エディタをCodeMirror的なものへ
テキストエリアをMarkdownエディタに切り替えるGreasemonkeyを書いた | Web Scratch
assign yourselfの自動化
- [ ]の一覧を可視化
インクリメンタルなIssue一覧のフィルタリング
アプリならでは常駐的な通知って何か必要なのかなー
azu self-assigned this 20 days ago
azu commented 20 days ago
Owner
自分へassignするとhttps://tacoapp.com/ にTodoとして出るという感じの仕組みがあるので、これ使って
Labels
None yet
Milestone
No milestone
Assignee
azu
Notifications
Unsubscribe
You're receiving notifications because you were assigned.
1 participant
Lock issue
Sub tasks
MailplaneのようなGitHub Issue Todoクライアント
エディタをCodeMirror的なものへ
assign yourselfの自動化
- [ ]の一覧を可視化
インクリメンタルなIssue一覧のフィルタリング
TODO

Faao


Quick Issue
filter word - state:open
Accounts
@azu
Created
Assigned
Mentioned
Review-Requested
Queries
Faao
almin
immutable-array-prototype
textlint
azu@todo
almin: circular issue UseCase <-> UseCase...
$ depcruise --validate .dependency-cruiser.json src...
almin/almin#220 updated 2017-07-03 20:07 by azu
Main: always show content and searchbar
SearchBar and Content will be shown when the items i...
Status: Proposal
Type: Bug
azu/faao#61 updated 2017-07-03 14:07 by azu
almin: Transaction of UseCase
It related with Unit of work #186 Almin's unit of work ...
Status: Proposal
almin/almin#219 updated 2017-07-03 09:07 by azu
Sync with gist
We want to add sync feature betweetn A PC <-> B PC ...
Status: Proposal
azu/faao#59 updated 2017-07-02 18:07 by azu
Release
First Release - [ ] Electron app - [ ] Mobile #51 - ...
Type: Maintenance
azu/faao#58 updated 2017-07-02 00:07 by azu
Wide screen
![image](https://user-images.githubusercontent.com/1...
Priority: Medium
Status: Proposal
azu/faao#57 updated 2017-07-03 14:07 by azu
fromJSON shoule be catch
https://github.com/
GitHub
Search GitHub
Pull requests
Issues
Marketplace
Gist
azu
3 minutes ago
BridgeAR commented on pull request nodejs/node#13755
I guess in that case the the S should also be renamed but I can't think of what it stands for right now. Does it stand for Script ?
5 minutes ago
bobheath33435 commented on issue ReactiveX/rxjs#2539
@bmayen My comments were and are intended to be constructive. I understand how inexperienced developers like yourself may not understand that promo...
gaearon starred uanders/react-redux-cheatsheet 6 minutes ago
6 minutes ago
simon04 commented on issue localForage/localForage#635
For reference, Firefox goes with a storage attribute on the option object to the open() function: https://developer.mozilla.org/en-US/docs/Web/API/...
7 minutes ago
olegdunkan commented on issue Microsoft/TypeScript#16898
You have constraint in MIxin function <T extends Constructor<any> therefore by default
Repositories
jser/jser.ir
jser/realti
jser/sourc
almin/almi
asciidwan
Show
Your reposit
Find a reposit
All
Public

Faao - Feature

- Support Modern browser/mobile/Electron(recommenced)
- Support GitHub.com and GitHub Enterprise(GHE)
- Search Issue/Pull Request
  - Search Syntax is same with GitHub Search
- Mixed the result of search
  - e.g.) You can see the results of Created, assigned, mentioned as a single result
  - e.g.) You can see the results of `repo:azu/todo` on `github.com` and `repo:azu-ghe/todo` on GHE as a single result
- Support GitHub User Activity
- Quick to create issue
- Import/Export profile data

目的

- OOSでGitHub Issueをちゃんと扱うものがない
- 技術的目的
  - Almin + TypeScript + DDD[ドメイン駆動設計]である程度の規模のアプリケーションを作りたかった

ドメイン駆動設計 レイヤー化とドメインモデル層を持つ設計や開発思想

規模感(2019-06-04現在)

```
✈  cloc src
     239 text files.
     234 unique files.
       6 files ignored.

github.com/AlDanial/cloc v 1.80  T=1.06 s (219.4 files/s, 18371.8 lines/s)
-------------------------------------------------------------------------------
Language                     files          blank        comment           code
-------------------------------------------------------------------------------
TypeScript                     186           1009            584          10147
JSON                             9              1              0           6795
CSS                             36            135             61            769
Markdown                         2              3              0              6
-------------------------------------------------------------------------------
SUM:                           233           1148            645          17717
-------------------------------------------------------------------------------
```

作戦

- 「ちゃんと考えてちゃんとやる」
- 技術的ショーケースとしての意味合いを持つ
  - ちゃんとモデリングする
  - ちゃんとテストを書く
  - ちゃんとドキュメントを作る

アジェンダ

- 前半: ドメインの関心事
- 後半: 技術的な関心事

DDD（ドメイン駆動開発）

ちゃんとモデリング[モデル]をやる

[モデル] ここでいうモデルはEntityとかValue Objectを含めたドメイン上のモデルクラス

Disclaimer

- DDD は答えではありません
  - ただしドメインの関心事は言語に依存しにくいという特徴がある
  - 設計で困ったときに考えること/実装が言語に依存しにくい
  - C#、Java、F#、Scalaなど別の言語での取り組みも参考にしやすい
  - 永続層は技術的な関心事なので言語によって結構違う(Repository)
- このスライドはクライアントサイドのアプリについてです
  - サーバにおけるDDDとは少し関心が異なる部分があります

クライアントサイドでのDDD

- faao/domain.md at master · azu/faao
- ドメインモデルの寿命が長い
  - 特に何かのサービスに対するクライアントアプリはずっと立ち上げっぱなし
  - ウェブサイトと異なりリロードしなくても動くように作る
- サーバ側の概念とクライアント側の概念は一致しないことがある
  - サーバ(GitHub)的にアカウントに対してGitHub APIのトークンが複数紐づく
  - クライアントからはTokenがあり、そのTokenに紐づくアカウントがいるように見える
    - クライアントからは、トークンがなければアカウントは分からない、アカウントだけ分かってもトークンがないと何もできない

モデリング

- AppUser: アプリケーションのユーザー
- GitHubSetting: TokenやAPI hostなどを含んだセッション情報
- GitHubUser: GitHubのAPIを叩いた結果取得できるGitHubユーザー情報

多くの処理(ユースケース)は

AppUserが**GitHubSetting**を使って 〜〜 する

のようになることが分かってくる

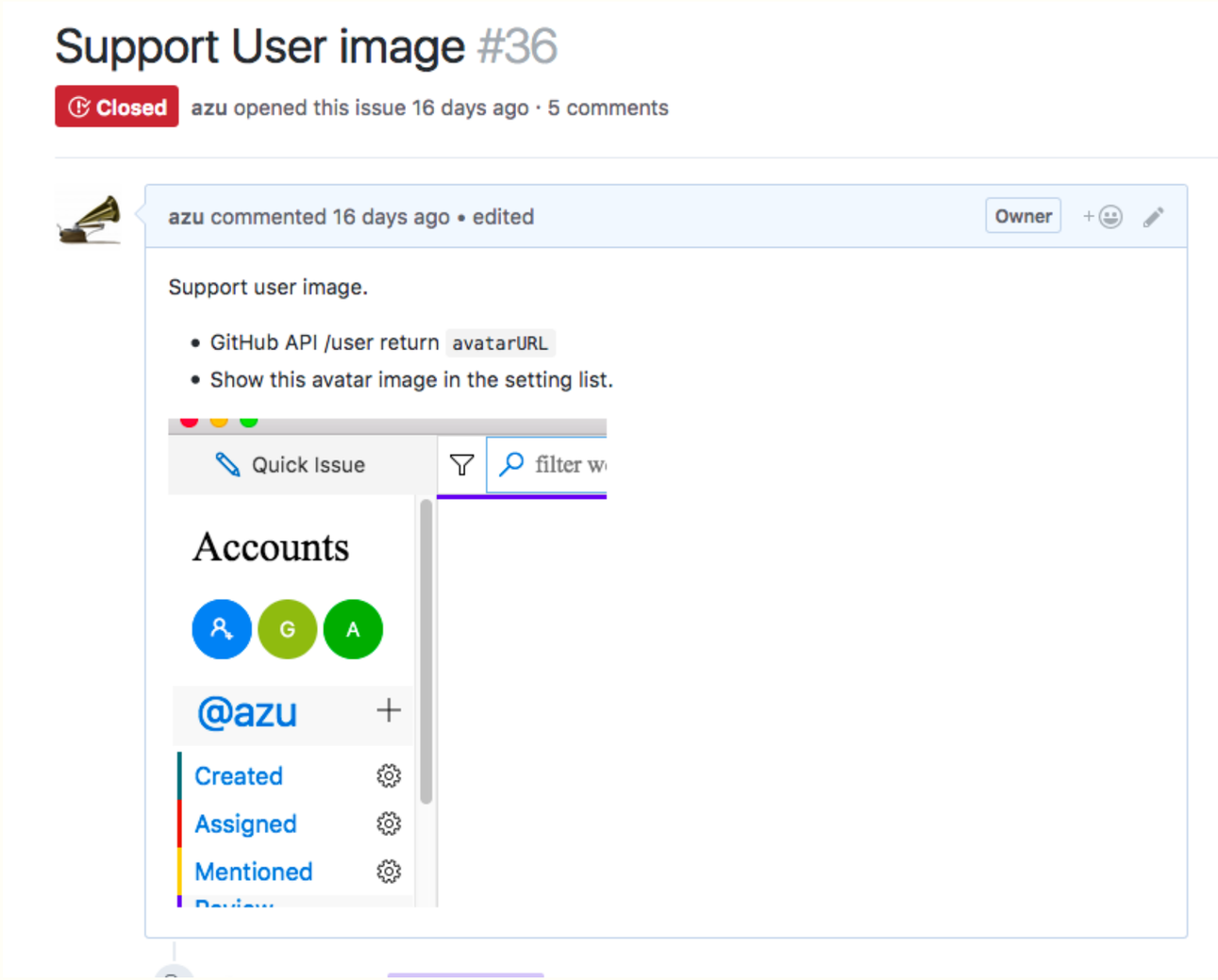
遠回りのモデリング

- 実際モデリングをしっかりやると進みが遅く感じる
  - 一つのモデルが大きくなりすぎないように気を配ったり
- 遠回りしてよかった場合もある
  - 安易なUI起因の値がドメインに流れてくるのを防げる

遠回りの例

- GitHubSetting(Account)にアイコン画像を設定したいというIssue
- 安直にやるなら `GitHubSetting` へ `avatarImageURL` などを追加すれば終わり

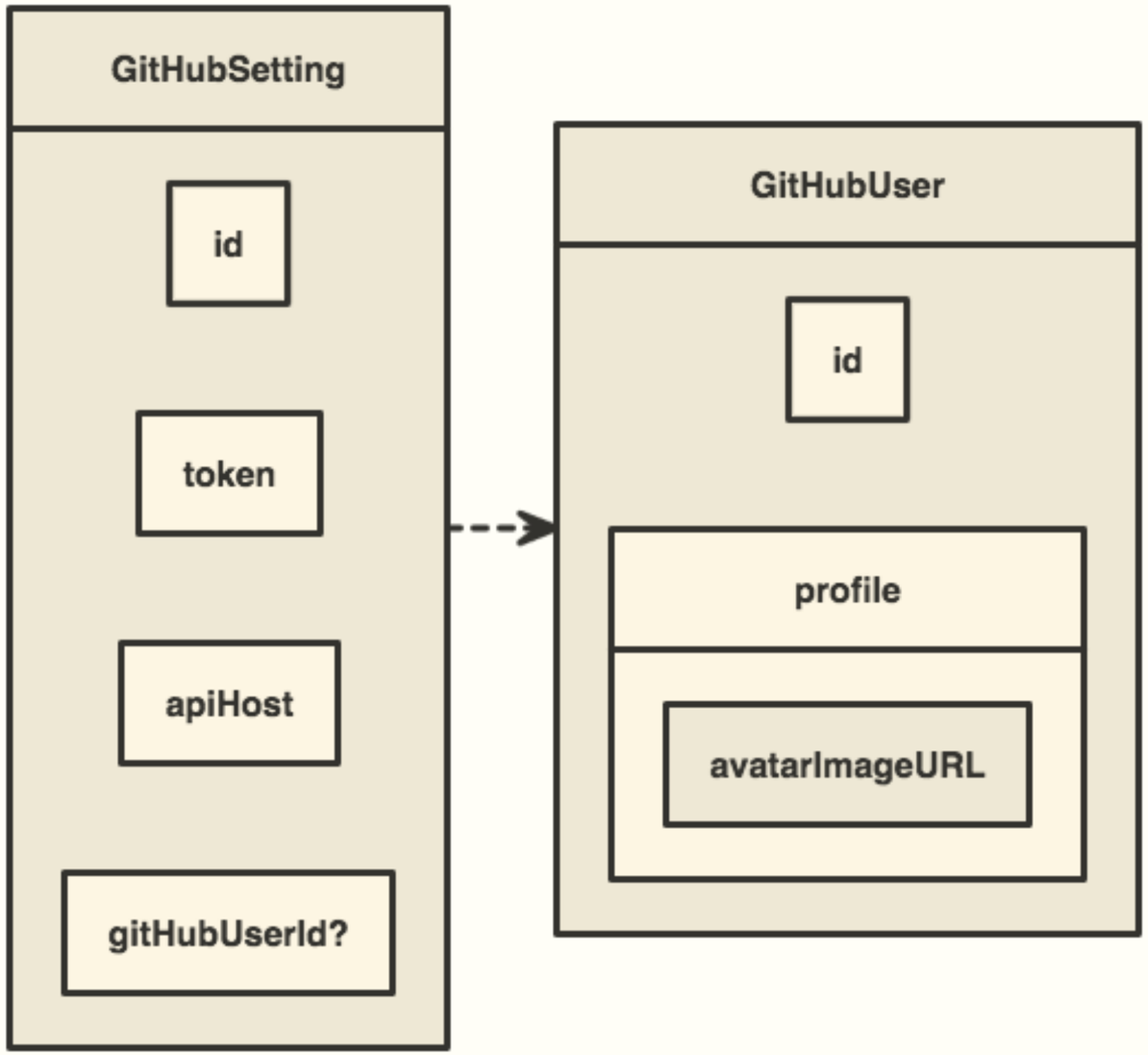
```
interface GitHubSetting {
        id: Identifier<GitHubSetting>;
        token: string;
        apiHost: string;
        webHost: string;
        // ADD?
        avatarImageURL?: string;
}
```



遠回りの例 -> GitHubUser

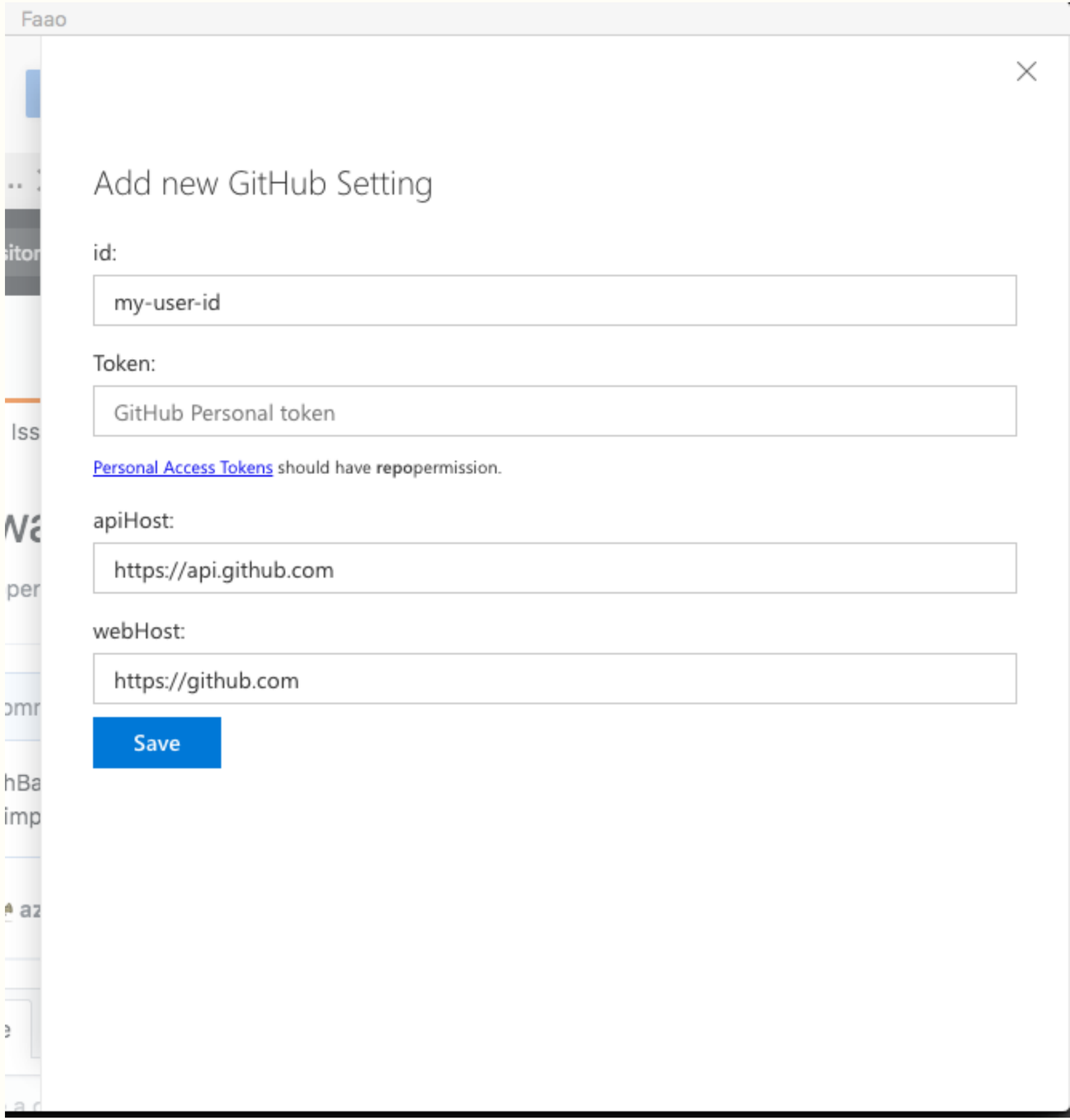
- 後回しにしていて、GitHubUserのActivityを表示したいと思った
- このときに、GitHubUserというモデルが必要で、avatarImageURLはこのGitHubUserのprofileに属するデータであると分かった
- 結果的にGitHubSettingに追加されたのはGitHubUserへのRelationship Id

```
interface GitHubSetting {
      id: Identifier<GitHubSetting>;
      token: string;
      apiHost: string;
      webHost: string;
      // Relationship
      gitHubUserId?: Identifier<GitHubUser>;
}
```



遠回りのモデリング

- GitHubSettingとGitHubUserは想定するライフサイクルが異なった
- GitHubSettingで入力されたTokenを使って、/user APIを叩いてGitHubUserを作る
- 異なるライフサイクルを一つのモデルにまとめると破綻する未来が見えていた
- そのため、UIのためにいきなりモデルを変更するよりちゃんと必要なモデルを考える



Star機能のモデリング

- 表示しているIssue/PRを ⭐ Starできる機能を実装しようとした
  - 目的: 今日作業するIssueをストックしてまとめて表示できる機能をつけたい！
- 表示しているIssueのデータは`GitHubSearchResultItem`を元にしている
  - 正確に言うと`GitHubSearchResultItem`がViewModelを作りそれを元に表示を作る

Faao

Quick Issue

state:open

Accounts

azu

Queries

- faao
- assignee:azu
- jsprimer
- ★Star
- Promise本

Work

- dev
- Work:Me

Options

- When delete query, it should be del...
  Delete a single query. This changes will affect ...
  azu/faao#123 3 days ago 0
- Does not Scroll when user scrolled
  Force scrolling make me frustration.
  azu/faao#122 3 days ago 0
- Memory Usage
  We need to improve memory usage. OSNotice ...
  Type: Feature
  azu/faao#111 9 days ago 1
- AutoClean filter
  https://twitter.com/azu_re/status/1125001718...
  Type: Feature
  azu/faao#112 a month ago 1
- Open Stream and move focus at fir...
  This force focus behavior is bad. We need to i...
  Type: Bug
  azu/faao#110 a month ago 1
- Release
  First Release [x] Electron app [x] Mobile #51 [...
  Type: Maintenance
  azu/faao#58 a month ago 1
- Re-open is failed at some times
  restart and re-open is failed
  azu/faao#109 a month ago 0
- Integrate with browser(URL scheme)
  We want to integrate with Browser UseCase R...
  Type: Feature
  azu/faao#87 a month ago 0

https://github.com/azu/faao/issues/123

Search or jump to... / Pull requests Issues Marketplace Explore

azu / faao

Unwatch 5 Star 53

Code Issues 24 Pull requests 0 Actions Projects 0 Wiki Security Insights Settings

When delete query, it should be deleed from Queries #123

Edit New

Open azu opened this issue 3 days ago · 0 comments

GitHubSearchResultItemを元にしてる

azu commented 3 days ago Owner

Delete a single query. This changes will affect parent queries.

Write Preview

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Close issue Comment

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Notifications
Unsubscribe
You're receiving notificati because you're watching repository.

1 participant

Star機能のモデリング

- `GitHubSearchResultItem` (Value Object) に `isStared` を追加すれば OK?

```
export class GitHubSearchResultItem {
    id: Identifier<GitHubSearchResultItem>;
    body: string;
    html_url: string;
    number: number;
    title: string;
    // ...
+   isStared: boolean;
}
```

Star機能のモデリング

`GitHubSearchResultItem` に `isStared` を追加すればOK?

- No
  - User が `GitHubSearchResultItem` を Star する ということは
  - FaaoにはSearchResultがあり、特定のSearchResultItemをStarしたいわけじゃない
  - ユーザーはそのGitHubSearchResultItemに対応するIssue/PRをStarしたい
  - => つまり GitHubSearchResultItemのidなりを記憶しておきたい
  - さらにいえば、GitHubSearchResultItemをスナップショットとして保存するのではなく、そのGitHubSearchResultItemに相当するItemの最新の状態を一覧したい
  - => これは保存したStarのIDをまとめて検索したいと言い換えできる

Star機能のモデリング

- Add Faao Query by azu · Pull Request #91 · azu/faao
- 新しい種類のQueryとしてモデルを作り実装した
  - Starすると、そのQueryの検索パラメータにIssueのIDを追加する
  - UnStarすると、そのQueryの検索パラメータにIssueのIDを削除する
- Star済み一覧を表示する = StarのQueryで検索する
  - 常にStarしたIssueの最新の状態が表示できる

Quick Issue

filter word - state:open

https://github.com/textlint/textlint/iss

Accounts

azu

Queries

- ★Star
- faao
- textlint

Add List ···

Drop Node.js 6.x

[nodejs/Release: Node.js Foundation Release ...

Type: Breaking Change

textlint/textlint#600 azu@2019-05-01 12:05 0

Integrate with browser

We want to integrate with Browser ## UseCa...

Type: Feature

azu/faao#87 azu@2019-02-11 17:02 0

Query Settings

Select your GitHub setting:

azu

This GitHub setting has token and api host.

Name:

★Star

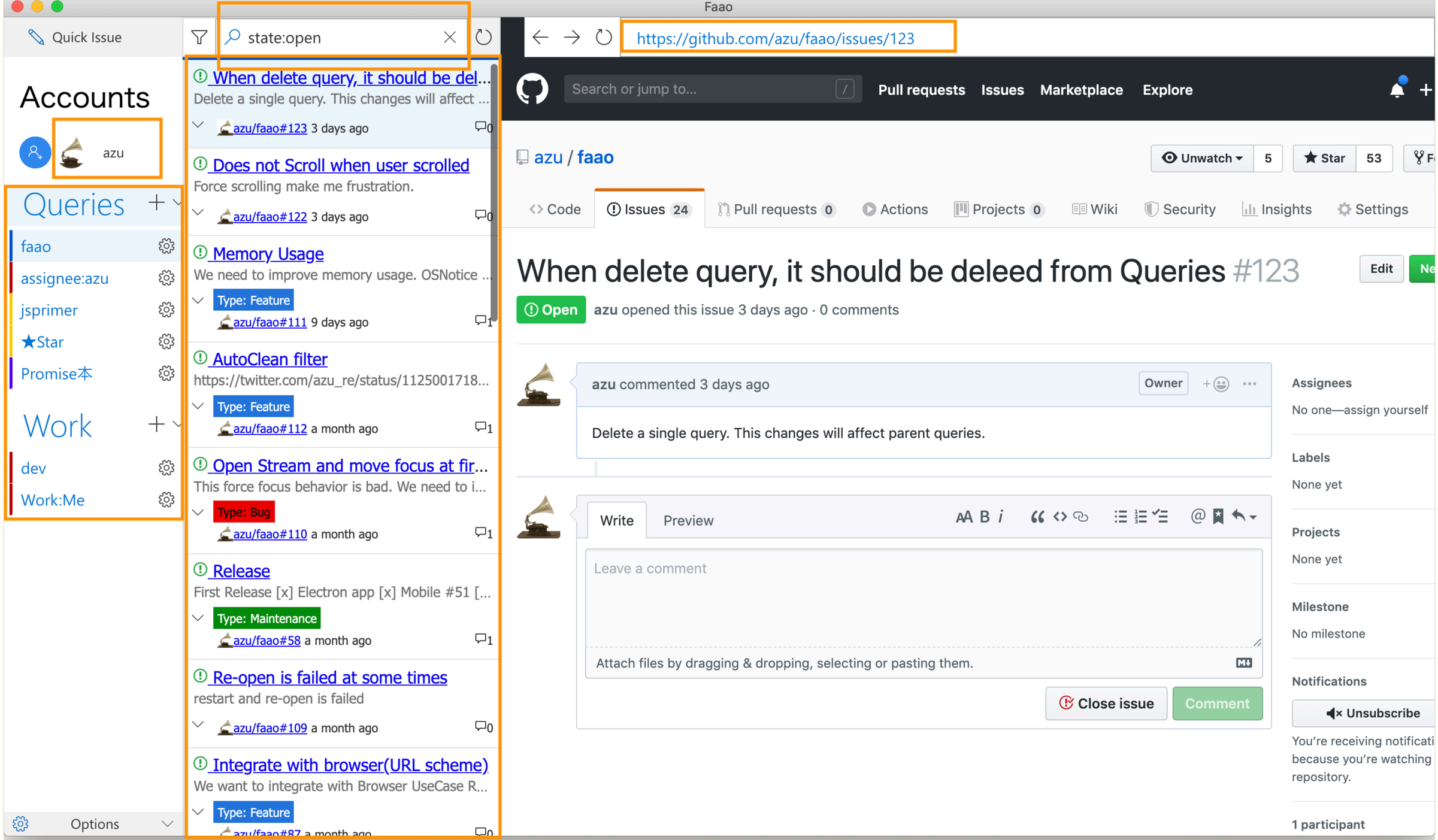
Color:

fccb00

Save

ドメインモデル -> 永続化

Faaoで永続化してる部分


Faao
Quick Issue
state:open
https://github.com/azu/faao/issues/123
Accounts
azu
Queries
faao
assignee:azu
jsprimer
★Star
Promise本
Work
dev
Work:Me
Options
When delete query, it should be del...
Delete a single query. This changes will affect ...
azu/faao#123 3 days ago
Does not Scroll when user scrolled
Force scrolling make me frustration.
azu/faao#122 3 days ago
Memory Usage
We need to improve memory usage. OSNotice ...
Type: Feature
azu/faao#111 9 days ago
AutoClean filter
https://twitter.com/azu_re/status/1125001718...
Type: Feature
azu/faao#112 a month ago
Open Stream and move focus at fir...
This force focus behavior is bad. We need to i...
Type: Bug
azu/faao#110 a month ago
Release
First Release [x] Electron app [x] Mobile #51 [...
Type: Maintenance
azu/faao#58 a month ago
Re-open is failed at some times
restart and re-open is failed
azu/faao#109 a month ago
Integrate with browser(URL scheme)
We want to integrate with Browser UseCase R...
Type: Feature
Search or jump to...
Pull requests
Issues
Marketplace
Explore
azu / faao
Unwatch 5
Star 53
Code
Issues 24
Pull requests 0
Actions
Projects 0
Wiki
Security
Insights
Settings
When delete query, it should be deleed from Queries #123
Edit
Open
azu opened this issue 3 days ago · 0 comments
azu commented 3 days ago
Owner
Delete a single query. This changes will affect parent queries.
Write
Preview
Leave a comment
Attach files by dragging & dropping, selecting or pasting them.
Close issue
Comment
Assignees
No one—assign yourself
Labels
None yet
Projects
None yet
Milestone
No milestone
Notifications
Unsubscribe
You're receiving notificati
because you're watching
repository.
1 participant

永続化

- 検索履歴
- 検索クエリ
- 検索結果など
- 大体のモデルが永続化可能な形になってる
  - オフラインでも表示できるため

ドメインモデルは永続化(技術的制約)を知らずに済むか

Patterns, Principles, and Practices of Domain-Driven Designより

- 妥協なしで行う
  - NHibernate[読]やEntity Frameworkなどのデータモデルとのマッピングできるものを使う
  - モデルをそのままJSONなどにシリアライズして保存できるデータストアを使う
- 妥協ありで行う
  - リポジトリからデータを引くときに、Entityに対して外から値を指しながら復元させる
  - Mementoパターン - Entityのスナップショットとデータモデルをマッピング(今これ)

[読] えぬはいばーねいと

Using a Persistence Framework That Can Map the Domain Model to the Data Model without Compromise

If you are mapping to a relational database in a greenfield environment and you are using an ORM that supports persistent ignorant domain objects, you will be able to map your domain model directly to the data model, as shown in Figure 21-2.

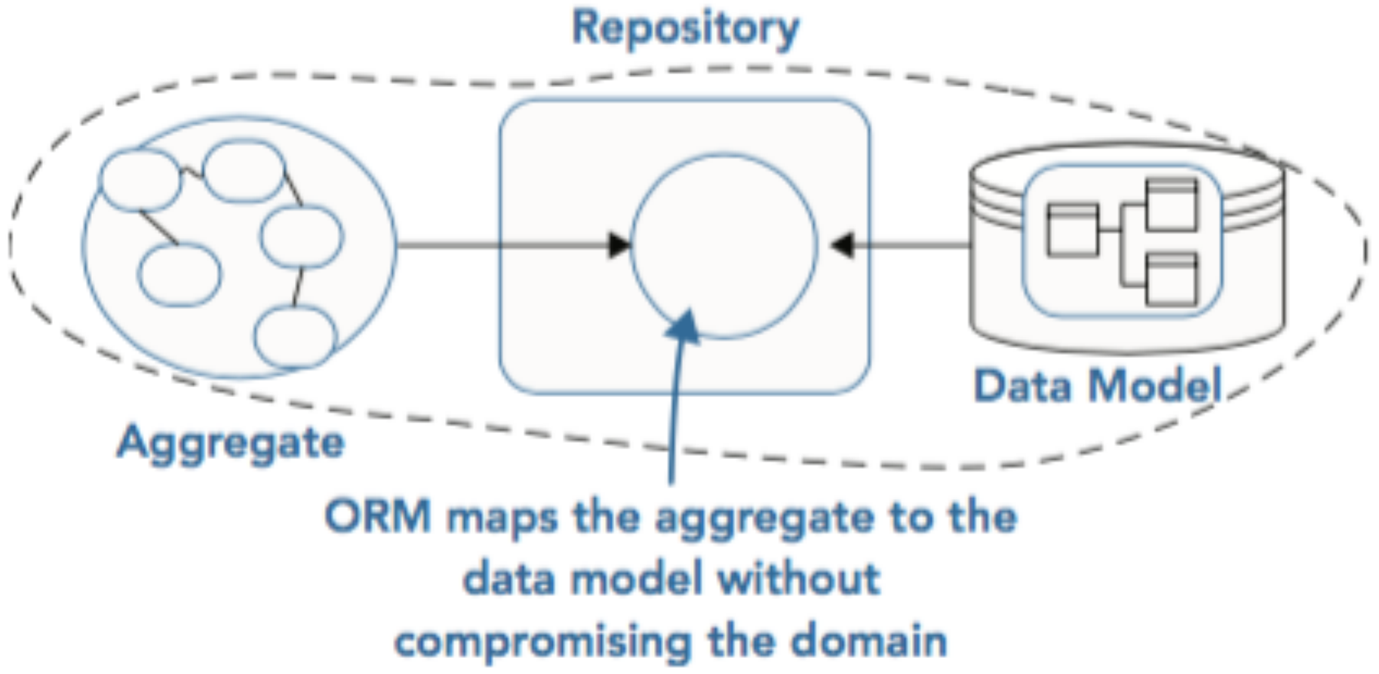


FIGURE 21-2: An ORM maps between the domain and the persistence model.

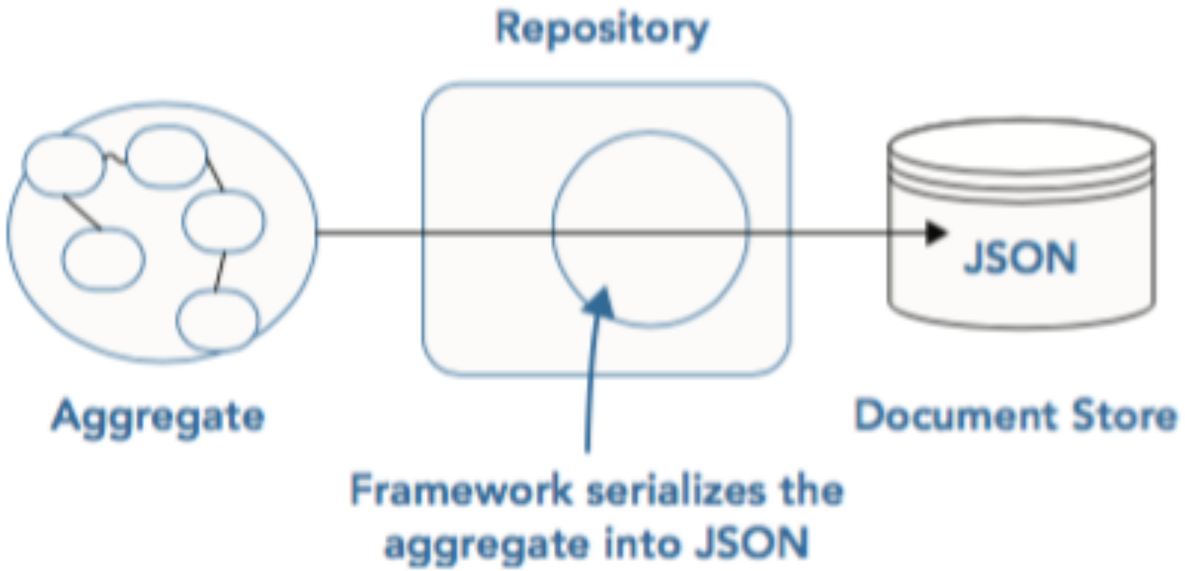


FIGURE 21-3: An aggregate can be serialized and stored.

Using a Persistence Framework That Cannot Map the Domain Model Directly without Compromise

If you are using a persistence framework that does not allow your domain model to be persistence ignorant, you need to take a different approach to the way you persist and retrieve your domain objects so they remain free of infrastructural concerns. There are a number of ways that you can achieve this, but all affect the domain model and the shape of your aggregates. This is, of course, the compromise you need to make your application work.

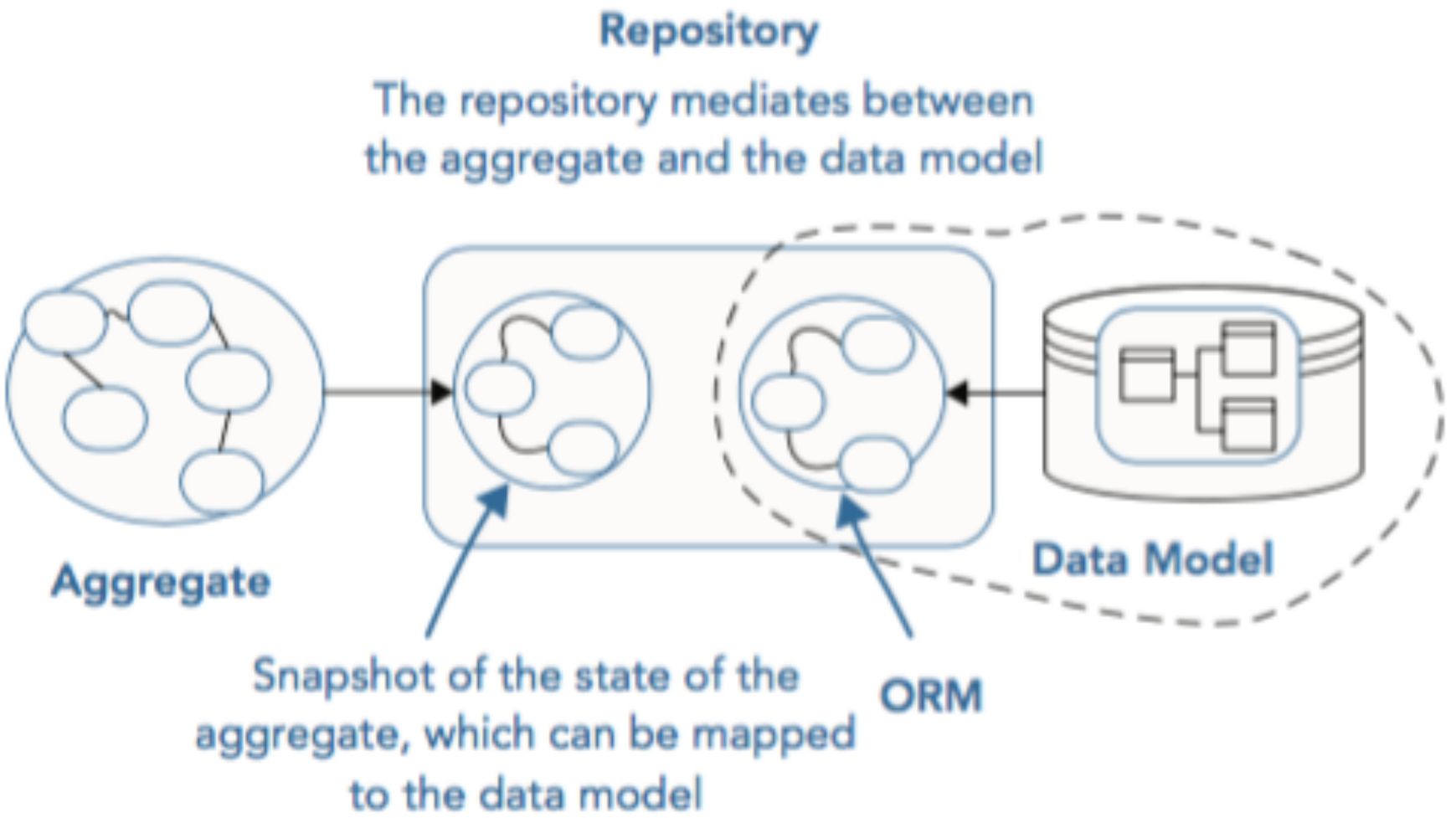


FIGURE 21-4: The memento pattern enables you to map a snapshot of the domain model to the persistence model.

妥協あり/なしの永続化

- ドメインは軽く永続化されることを意識する必要はある
- constructorでincrement idをしていると不整合を生むので駄目
  - constructorでちゃんと`{ id }`なども受け取れるようにする
  - モデルの初期化は面倒になっていくのでFactoryが初期化を担当する

```
// 駄目なケース
let id = 0;
class User {
   constructor(){
      this.id = id++
   }
}
```

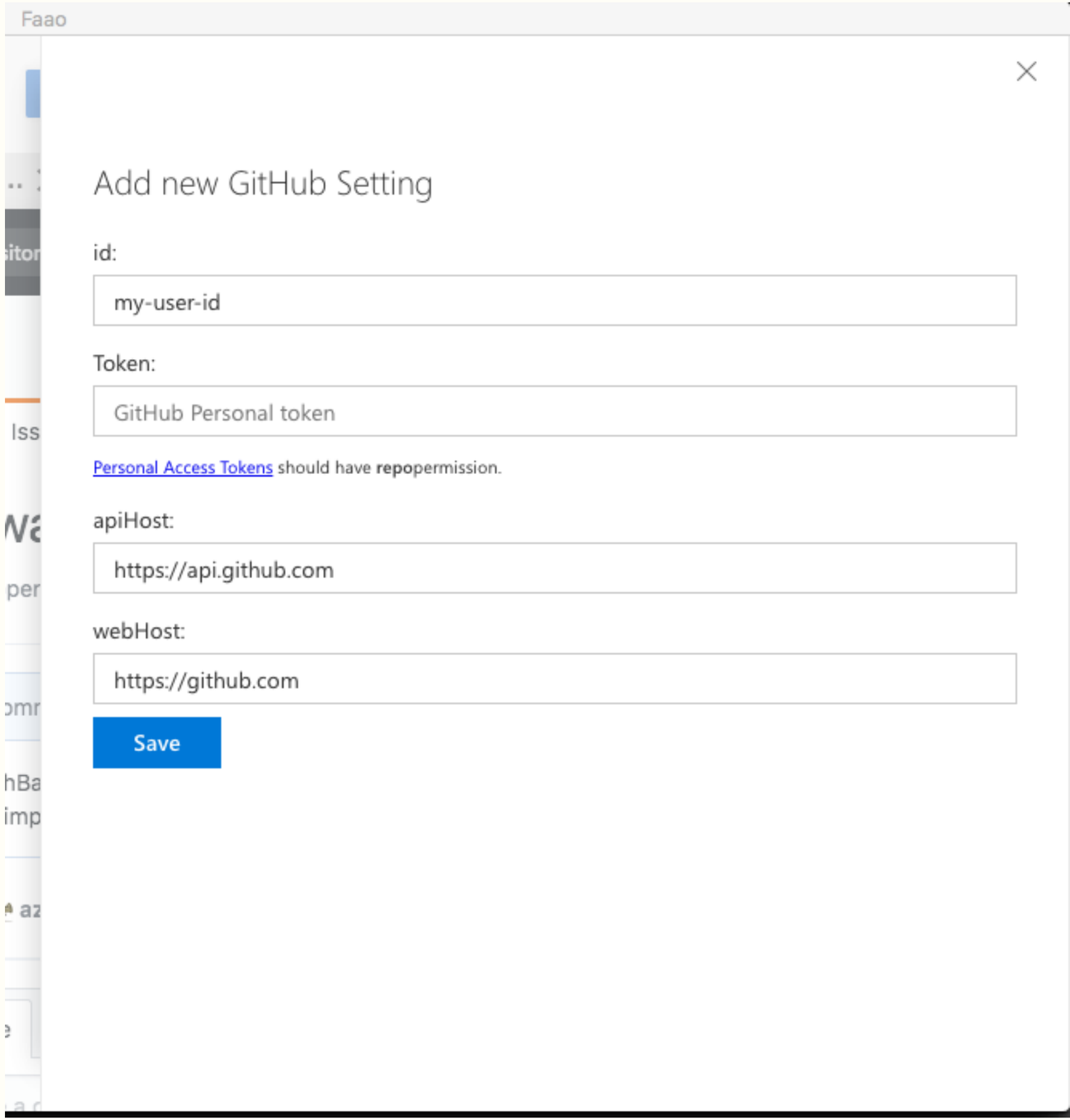
どちらにしてもドメインは軽くは永続化を意識する

- {id}をconstructorで受け取れるようにする
- 永続化を考える場合は、constructor(初期化)に副作用を持たせてはいけない
  - 初期化処理に副作用をもたせると、初期化フローごとに別々の実装が必要

```
// OKなケース
class User {
   constructor({ id }){
      this.id = id;
   }
}
```

スナップショットからの復元

- 今採用してるパターン
- 妥協ありパターンの一種であるTypeScript: Working with JSON · Choly's Blog(Entityに対して外から値を指しながら復元させる)に比べると少し安全で何とか手で書いていけるレベル
- しかしスナップショットが現在のモデルと一致してるとは限らない
- スナップショットのバージョニングなどが必要となっていく
  - フレームワークになってないとそろそろ面倒



Repository

- インメモリで終わる or データが常にサーバにある場合のRepositoryは単純なMap
- モデルの永続化を考えだしたときに大変になるのは、Repository
- モデルも永続化は全く意識はしてない場合、後から概念/構造に変更が出て大変となる
  - 影響度: 概念 > 構造 > 実装...
- ついでに永続化するとIndexedDBなどを使うの非同期処理がやってくる
  - Faaoの実装では初期化と保存のみを非同期にして、Readは同期にした
  - Readを非同期にするとStoreも非同期にする必要がでてきて面倒そうだった

UseCase

UseCase

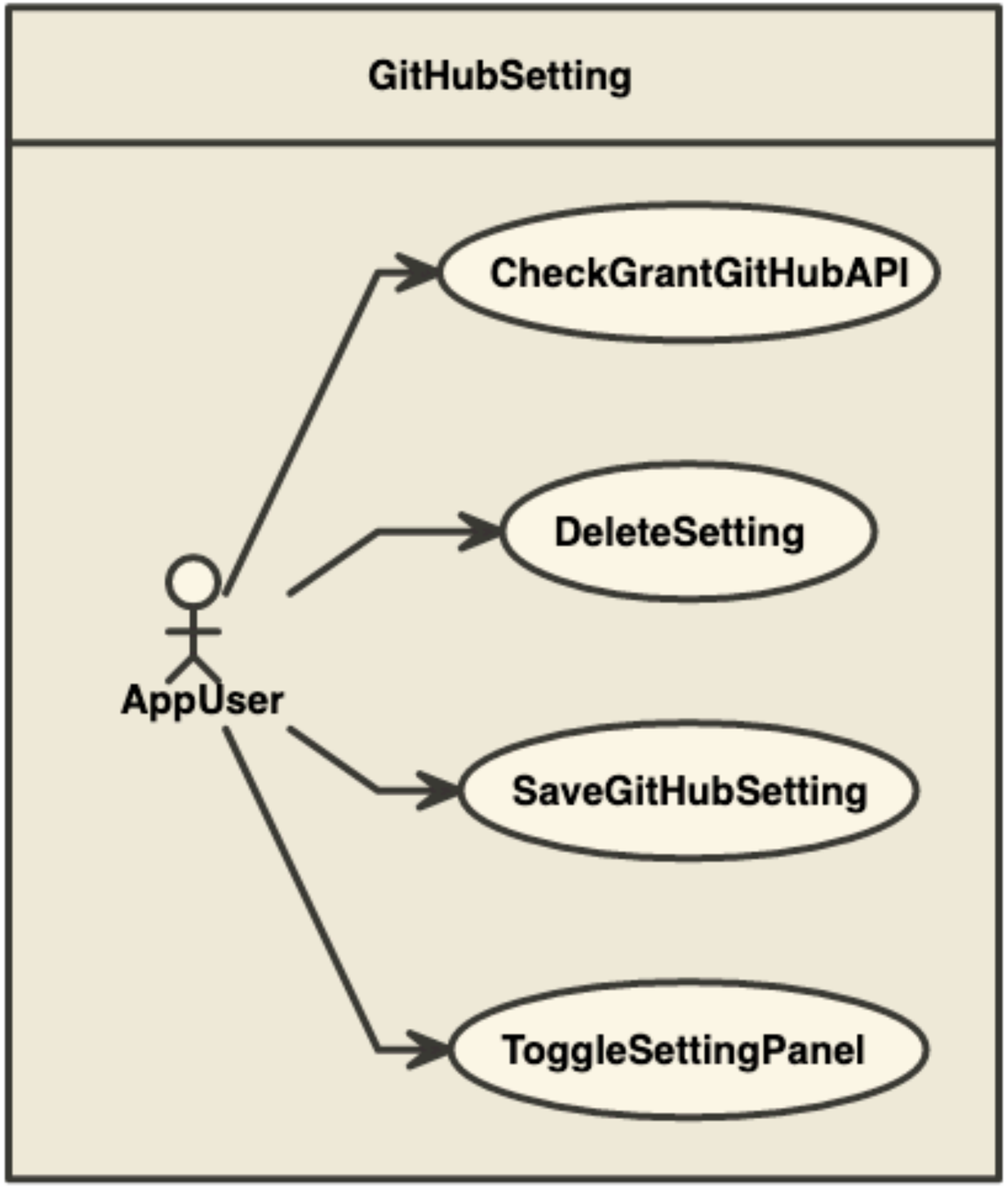
- アプリケーションのドメインを使い、やりたい処理の流れを書くところ
- このアプリのユースケースは
  - GitHubSettingのtoken情報とGitHub APIを使って検索
  - GitHubSettingの作成、保存 などなど
- ユースケースの再利用性
  - 基本的にはしない、拡張ユースケースは使う
  - UseCaseの再利用性 - yoskhdia’s diary

ユースケース図

- Faaoのユースケースはコードから自動生成される
  - Faao - UseCase architecture
- このユースケース図はアプリの全てを表すわけではないけどモデルの整合性の参考にできる

一点、注意が必要なのは、ユースケース記述とユースケース図は異なるということです。このガイドラインはユースケース記述のガイドラインです。

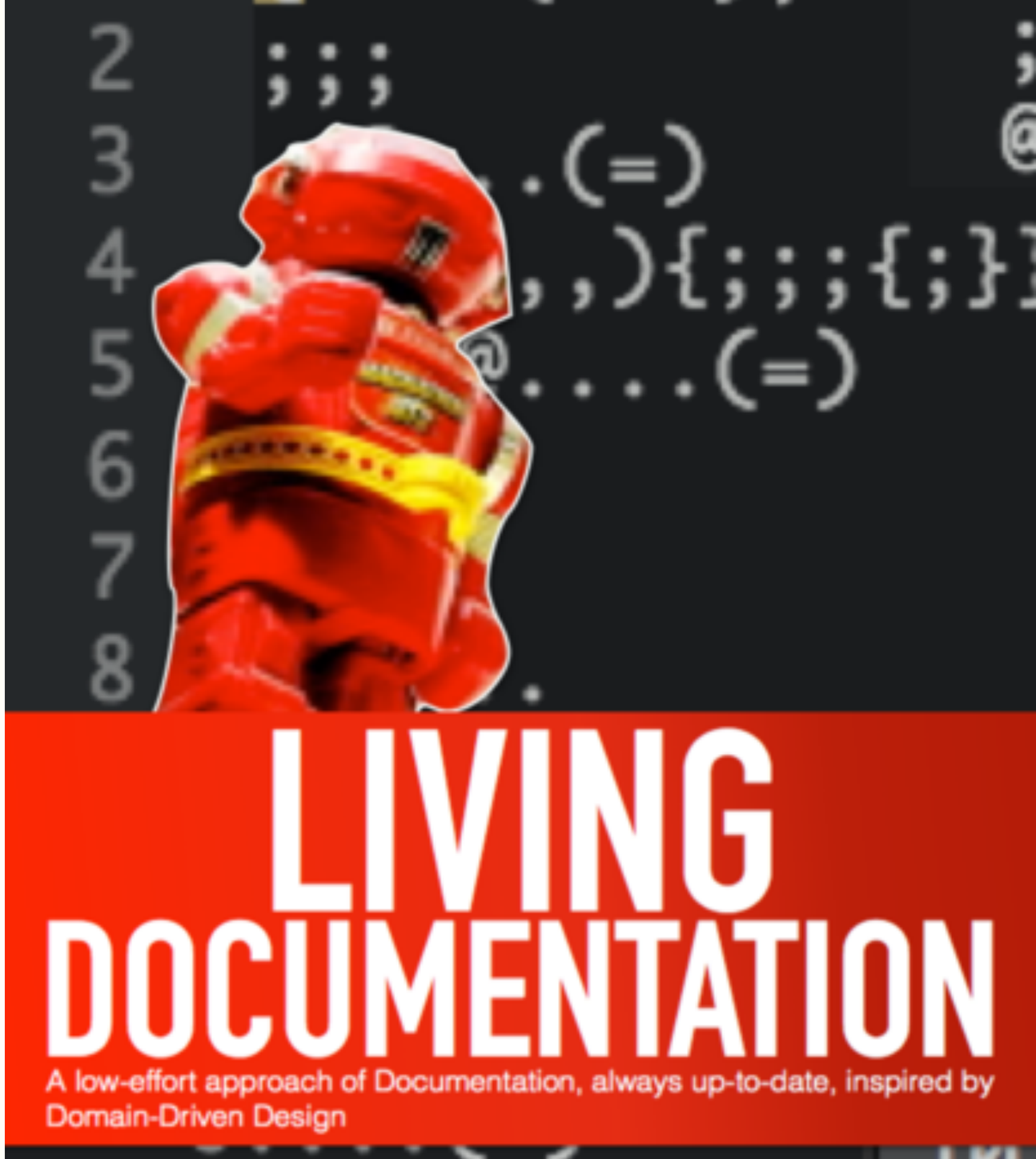
UseCaseの再利用性 - yoskhdia’s diary



ルールとドキュメント

Living Documentation

- Living Documentation by design, with Domain-Driven Design
- https://leanpub.com/livingdocumentation $0〜$40で購入



知識の共有

KnowledgeにはGenericなものとSpecificなものがある。
会社やチーム、プロダクトにおけるSpecificな知識には次のような問題が生まれやすい

- アクセスできない
- 古すぎる
- フラグメント化してる
- 暗黙的になってる
- 理解できない
- 書かれてない

Living Documentation

- これらの問題をLivingなドキュメントで解決するアプローチ
- ドキュメントもコードと同じ速度で成長する
- 良いドキュメントには良い設計が必要
- 良いドキュメントには自動化が必要
- 推測、憶測をドキュメント化しない

LivingDocumentationのコア原則

- Reliable - 信頼性の高いドキュメント
  - single source of truth
  - reconciliation mechanism - ソースが複数の場所にありテストで保証す
- Low-Effort
- Collaborative
  - Conversations over Documentations
  - アクセスできる場所に知識は置く
- Insightful
  - 意図を残す

具体的な問題と対策

- ガイドラインを決めてもそれを自動で守れないと意味がない
  - ツールで検証する
  - コードで検証する
- 更新されない構成図
  - Living Diagram - コードから図を生成するなど
- 更新されないユビキタス言語

守られないルールは価値がない

守られないルールは価値がない

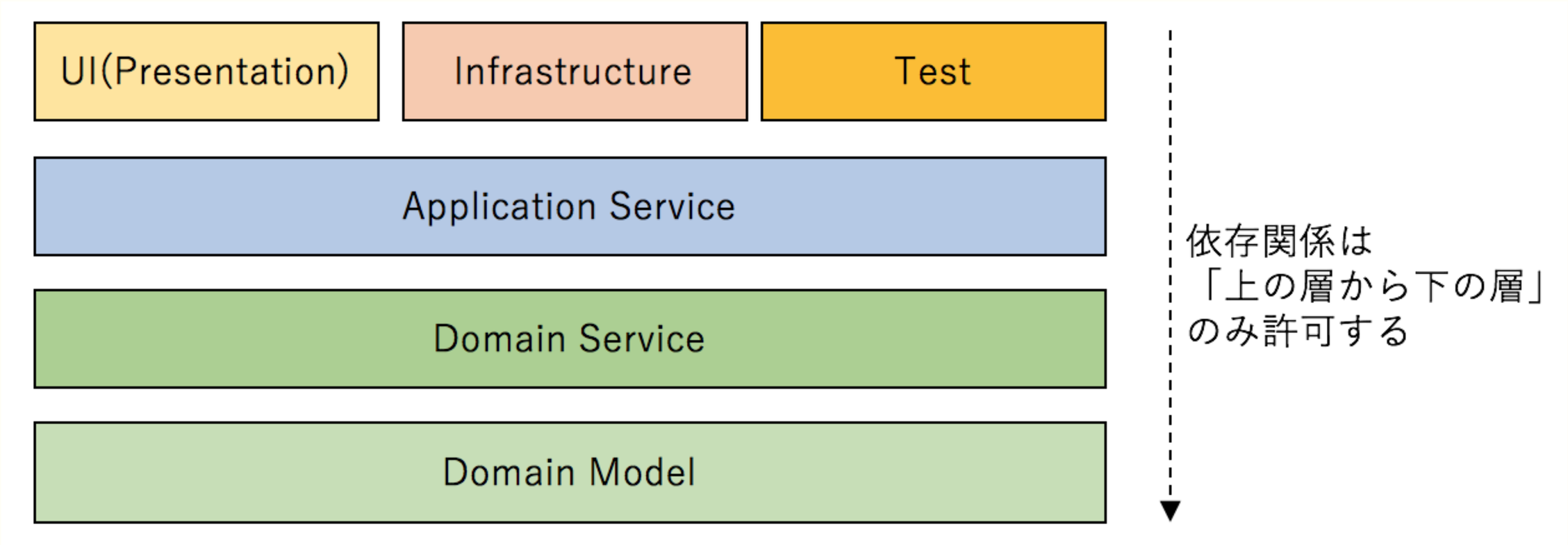
- 最も良いドキュメントはno document
- 必要となった時(ツールがエラーと言った時)に初めて見ることができればいい
  - ESLintがよくできている理由
- eslint, prettier, stylelint, webpack(case-sensitive-paths-webpack-plugin)などなど

例) ルール: ドメインのコードはインフラ(repository)を参照してはいけない

dependency-cruiserを使ってルールをコード化し依存関係を自動チェックする

```
{
  "forbidden": [
    {
      "name": "domain-not-to-depend-on-infra",
      "comment": "Don't allow dependencies from domain to infra",
      "severity": "error",
      "from": { "path": "^src/domain" },
      "to":   { "path": "^src/infra"  }
    }
}
```

補足: オニオンアーキテクチャ



[DDD]ドメイン駆動 + オニオンアーキテクチャ概略 - Qiitaより引用

破れないルールは価値を鈍化させる

破れないルールは価値を鈍化させる

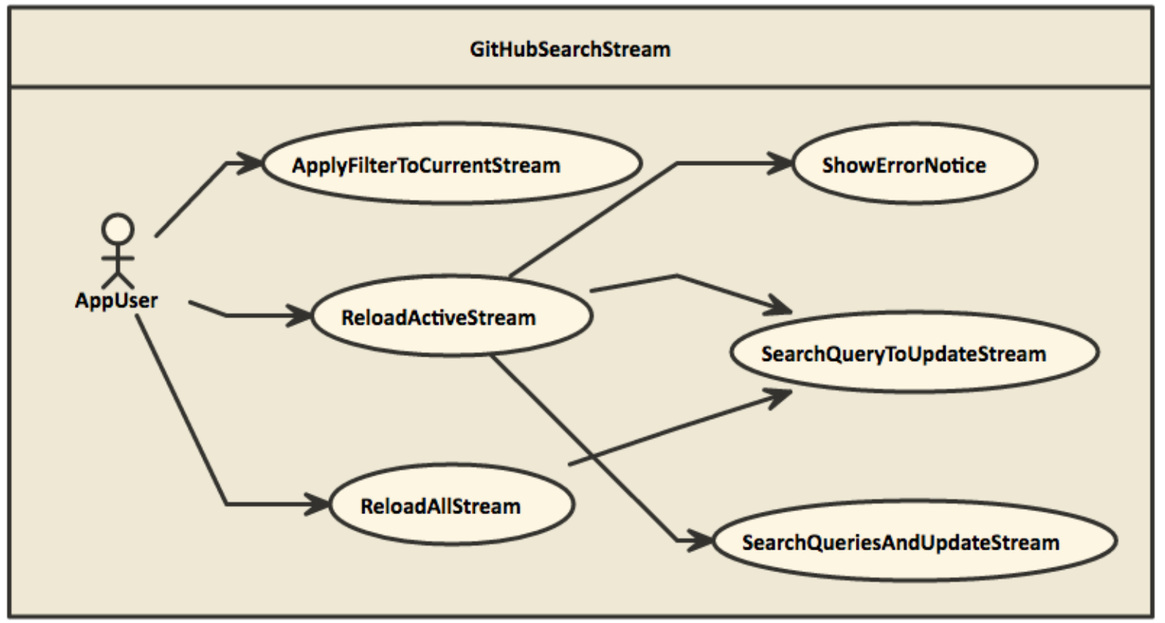
- ルールには例外がつきもの
- そのため、原則が守れないと崩壊してしまうルールよりは、例外を規定することで原則を守れるルールの方がよい。
- 厳密に守りたいルールは、ホワイトリストで例外ルールを管理できた方がいい
  - 例) `eslint-disable`で指定した部分だけ原則を無視できるようにする

Living Documentation

Living Diagram

ユースケース図のLiving Diagram

- Faao - UseCase architecture に全てのユースケース図が自動生成される
- Faaoのソースコードから自動生成
- ファイルからuse-caseを抽出、Text to UMLのnomnomlが食べられる書式にして変換
- almのUseCaseは拡張ユースケースを表現できる
  - ユースケースが別のユースケースを呼び出す
  - UseCaseの再利用性 - yoskhdia’s diary



Living Diagramの使いみち

- おかしなアクターを見つけることができる
  - 「名詞（主語） - 動詞 - 名詞（目的語）」(en)
  - 誰? がおかしいときがある。システムである場合など
- おかしなユースケースを見つけることができる
- 例外処理が抜けているかを見ることができる
  - ユースケースは処理の流れを書く
  - そのため、省かれた処理を見つけ適切にキャッチすると多くのバグが解決できる

2/3のバグはカバレッジを上げると見つかる

- 適切なエラーハンドリングが行われるか、例外を無視してないかをテストしていくことで、全体の2/3のバグが発見できる
  (データ集約型分散システムの論文)

A majority of the production failures (77%) can be reproduced by a unit test.

— *Simple testing can prevent most critical failures | the morning paper*

Living Documentationはドキュメンテーションをコード化する

- Living Documentationとはドキュメントがコードと共に成長できるようにする戦略

詳しくは本を読んで

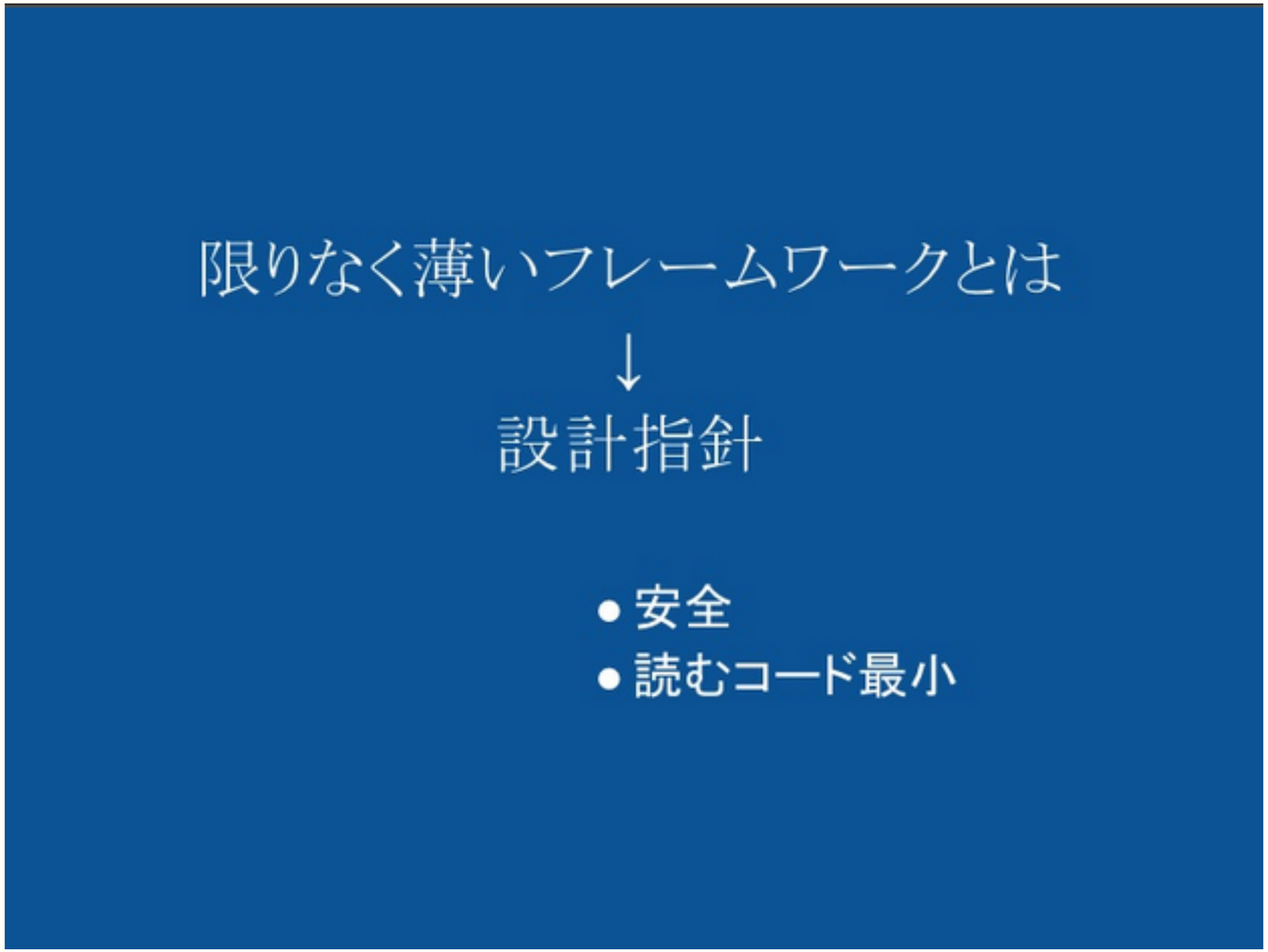
- Living Documentation by... by Cyrille Martraire [PDF/iPad/Kindle]
- Living Documentation by design, with Domain-Driven Designを読んだ | Web Scratch

技術的な関心事

Almin

Almin

- TypeScriptで書き直した
- Alminはフレームワークだが、今回のドメインやRepositoryは自分で書くところなので手出しはしない
- あくまで思考を手助けする(そういう風にかけるというドキュメントがある)



ぼくのかんがえたさいきょうのうぇぶあぷりけーしょんふれーむわーく - YAPC Asia 2011

TypeScript

- まあ普通
- DDDみたいにドメインちゃんと書くなら型ないとつらいよ
  - 型あってもつらいけど

Electronでウェブサイトを先読み

- Electronの`<webview>`は先読みができない
- `BrowserView`を複数作り非表示にして内容だけ先読みする
  - 表示するタイミングになったら表にだす`BrowserView`を変更する
  - Electron 5から1つの`BrowserWindow`が複数の`BrowserView`を表示できるようになった
  - https://electronjs.org/docs/api/browser-window#winaddbrowserviewbrowserview-experimental
  - Slackアプリも`<webview>`ではなく`BrowserView`を使ってる
  - 参考: Slackデスクトップは3.0でBrowserViewに移行した
- 実装: ViewPool.ts

Quick Issue

filter word - state:open

https://github.com/azu/faao/issues/59

Accounts

azu

Queries

faao

Sync with gist

We want to add sync feature betweetn A PC <...

Status: Proposal

azu/faao#59 updated 2017-07-02 18:07 by azu 1

Release

First Release - [x] Electron app - [x] Mobil...

Type: Maintenance

azu/faao#58 updated 2017-07-30 13:07 by azu 1

Wide screen

![image](https://user-images.githubuserconte...

Priority: Medium Status: Proposal

azu/faao#57 updated 2017-07-03 14:07 by azu 0

fromJSON shoule be catch

We should use `fromJSON` with try-catch. be...

Priority: Medium Type: Bug

azu/faao#56 updated 2017-07-03 14:07 by azu 0

Support GitHubUserEvent

- [x] Fetch UserEvents - [x] SHow UserEvents ...

azu/faao#55 updated 2017-07-02 00:07 by azu 1

Does't Update query

1. Update Query setting. 2. But, not reflect thi...

Type: Bug

azu/faao#54 updated 2017-07-02 00:07 by azu 0

Activity

- Click account and show activity - Acitivy is ra...

help wanted

azu/faao#53 updated 2017-08-05 17:08 by azu 4

Perf: introduce GA

メタテスト

- テストケースを生成するテストコード
- azu/large-scale-javascript: 複雑なJavaScriptアプリケーションを作るために考えること
- 特定のクラスやディレクトリに対してルールを守ってるかのテスト
- StoreがちゃんとStoreGroupに登録されてるか、初期Stateを返せてるかなど
- コードを書くと勝手にテストが増えて便利 ⭐

```
import assert from 'assert';
import glob from 'glob';
import path from 'path';
import interopRequire from 'interop-require';
const srcDir = path.join(__dirname, '../src/');
const useCaseFiles = glob.sync(`${srcDir}/js/use-case/**/*UseCase.js`);
describe('MetaUseCase testing', () => {
  useCaseFiles.forEach((filePath) => {
    // UseCaseはファイル名と同じUseCaseクラスを定義している
    const UseCaseName = path.basename(filePath, '.js');
    // ES6 modules と CommonJSどちらでも読めるように
    const UseCaseModule = interopRequire(filePath) || require(filePath);
    describe(`${UseCaseName}`, () => {
      it('UseCaseファイルはクラスをexportsしてる', () => {
        assert(UseCaseModule, `UseCaseファイルはexportsしてる: ${filePath}`);
      });
      it('UseCaseファイルはファイル名と同名のUseCaseを持つ', () => {
        const UseCase = UseCaseModule[UseCaseName];
        assert(typeof UseCase === 'function', 'UseCaseクラスが存在する');
      });
    });
  });
});
```

GraphQL

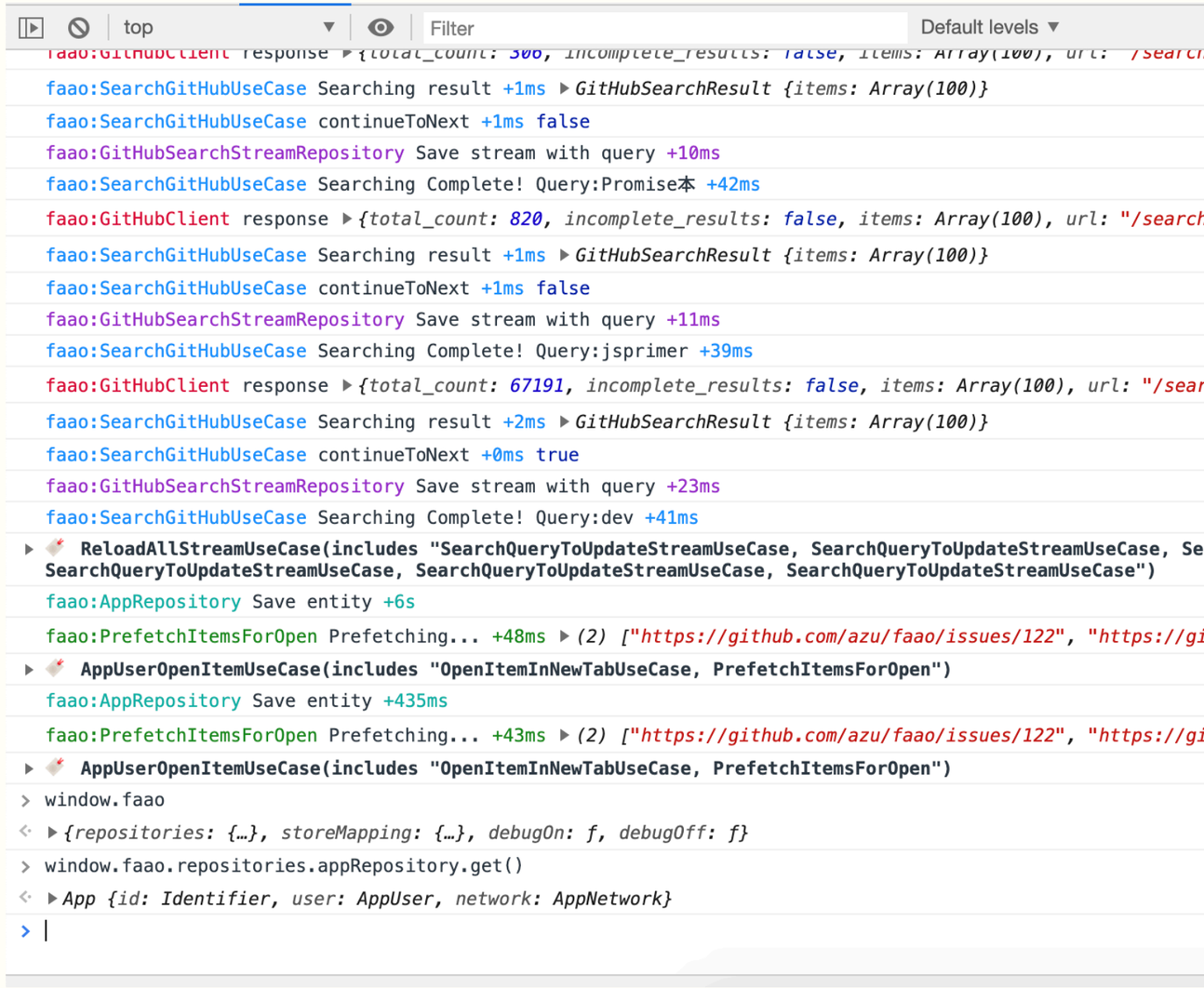
- GitHub API v4はGraphQL
- Star機能の実装にはGraphQLを使った
- GraphQLを使うことで、A,B,C,Dなど複数のissueの詳細情報をまとめて取得できる
  - REST APIではIssueの数だけAPIを叩かないといけない(n+1)
  - 検索結果は通常の検索結果と同じ`GitHubSearchResultItem`(value obejct)として保存される

```
            };
            const queries = query.searchParams.params
                .map(param => {
                    const parsed = param.parsed();
                    if (!parsed) {
                        return;
                    }
                    return `
  ${parsed.type}${parsed.no}: repository(owner: "${parsed.user}", name: "${parsed.repo}") {
    ${parsed.type === "issue" ? "issue" : "pullRequest"}(number: ${parsed.no}) {
      title
      repository {
        url
      }
      comments(last:1){
        totalCount
        nodes{
          url
        }
      }
      id
      number
      author {
        login
        avatarUrl
        url
      }
      state
      locked
      labels(first: 10){
        nodes{
          name
          description
          color
          isDefault
          url
        }
      }
      assignees(first: 10) {
        nodes {
          avatarUrl
          login
          id
          url
        }
      }
      milestone {
        title
        description
        url
        createdAt
        updatedAt
        dueOn
        closedAt
        state
      }
      createdAt
      updatedAt
      closedAt
      url
      body
    }
  }
`;
                })
                .filter(query => query !== undefined);
```

Philosophy

Debuggablity - 状態

- アプリケーションには様々な状態が存在する
- 全てはどこからでも現在の状態を見れるようになってないと不便
  - 簡単に `window.faao` に参照突き刺しておけばいい(逆にそうではないものは `window`にいるべきじゃない)
- Repository
- Store/State
- ViewのState



Debuggablity - データ

- 永続化したデータはいつでもメモリデータベースに切り替えできた方が良い
- テストの度に永続化したデータが消えるとテストしにくい
- FaaoではStorage.tsでいつでもメモリデータモードに入ることができる
  - localForageを使って動的にdriverを切り返す
  - データは元からメモリ上に載っていて、書き込み時にデータベースへアクセスする作りにしたため

Debuggability - イベント

- Stateとイベントを見比べる
- almin-loggerやalmin-devtoolsでUseCaseの実行を確認する
- イベントを見ることは大事
  - Webの世界はイベント駆動
  - DOMには色々なイベントがそれはブラウザによっても違う
  - video-events-debugger

死んでも復活できるようにする

- デスクトップアプリやモバイルアプリはクライアントにDBがある
  - オフラインでも動くようにするため
- マイグレーションが抜けてたりデータがおかしくなると真っ白になって死ぬ
- SPA時代にはよくあることなので、消えてもいいキャッシュを消す仕組みを備える
- 緊急脱出装置: ウェブサイトならClear-Site-Data
- クッキーbombやストレージが壊れてLocalStorageにアクセスできなくて例外が発生するといった問題解決に役立つ(実際壊れる)

まとめ

- ちゃんとやるにはちゃんとやる必要がある
- コードと共にテストやドキュメントも成長する
- それらは自動化されている部分もあればルール化されている部分もある
- モデリングをちゃんと行い、モデルから自動的にドキュメントが生成され、ドキュメントとしてみた時のモデルとしての不整合を検証する

参考

- azu/large-scale-javascript: 複雑なJavaScriptアプリケーションを作るために考えること
- Patterns, Principles, and Practices of Domain-Driven Design 1st Edition
- ユースケース駆動開発実践ガイド
- Living Documentation by… by Cyrille Martraire [PDF/iPad/Kindle]
- ボトムアップドメイン駆動設計 ｜ nrslib
- ビジネス・マネー