
Date
23 May 2022 to 27 May 2022

Call-in Information

- Time: 10:00 am, Eastern Time (New York, GMT-04:00)
  To join the online meeting:
  - https://lyrasis.zoom.us/j/84378615572?pwd=bGUxSjlyRTdjOGt5U1B6L0Yva3RQdz09
    Meeting ID: 843 7861 5572
    Passcode: 556561
    One tap mobile
    +16699006833, 84378615572#,,*556561# US (San Jose)
    +19292256099, 84378615572#,,*556561# US (New York)
  - Dial by your location
    +1 669 900 6833 US (San Jose)
    +1 929 205 6099 US (New York)
    +1 253 215 8782 US (Tacoma)
    +1 301 715 8592 US (Washington DC)
    +1 312 626 6799 US (Chicago)
    +1 346 246 7799 US (Houston)
    877 853 5257 US Toll-free
    888 475 4499 US Toll-free
    Meeting ID: 843 7861 5572
    Passcode: 556561
    Find your local number: https://lyrasis.zoom.us/u/kerqtGDrJ4

Slack
- https://vivo-project.slack.com

Attendees
1. Dragan Ivanovic
2. Georgy Litvinov
3. Benjamin Kampe
4. Mark Vanin
5. Veljko Maksimovic
6. Rolf
7. William Welling

Resources
- Sprint board
- 2022-05 Sprint Stand-up Reports

Agenda
1. Goals:
   a. Surface any fundamental issues
2. Sprint logistics / scheduling
   a. Daily stand-up report in Slack - #dynamic-api-sprint-may-2022 channel
   b. A bi-daily meeting (Monday, Wednesday, Friday) at 10am Eastern Time (4pm CEST)
3. Tools and resources
   a. Branches
   b. Test datasets
   c. Integration tests
   d. Development processes
      i. Git / GitHub processes
      ii. GitHub issues process
4. Tickets - https://github.com/orgs/vivo-project/projects/2/views/1
5. Documentation - VIVO Dynamic API
6. Tracks
   a. Ontology
      i. RDF representation of dynamic actions and resource APIs
   b. Binding
      i. Binding to Java objects representation
   c. Execution
      i. Execution of a dynamic action
   d. RCP Endpoint
   e. REST Endpoint
   f. Security
      i. Definition of dynamic action roles
      ii. Authentication (endpoint and UI)
      iii. Authorization
         1. JWT
         2. Do we need OAuth2 or OpenID Connect ???
   g. Validation
      i. syntax,
      ii. over the ontology,
      iii. generic SHACL rules for all dynamic actions
      iv. custom SHACL rules for specific dynamic actions ???
   h. UI for running a dynamic action
      i. Angular & Angular Material
      ii. Some notes regarding angular from William Welling .
         1. I would recommend against using a openapi specification generator
            a. it likely does not take into consideration angular universal for SEO, central state management, or i18n
            b. it is not likely to be a long lived dependency, popular, or actually work well with the dynamism of custom
               actions
         2. choose between the server side module (both express and angular universal)
            b. https://github.com/nestjs/ng-universal
         3. consider a central state management up front
            a. https://ngrx.io/
         4. choose i18n library (important to know whether we want runtime or build time i18n)
            b. https://ngneat.github.io/transloco/
            c. https://angular.io/guide/i18n-overview
      5. adopt a style guide and have it enforced using linting
         a. have github action check code styles
         b. protect main branch against style guide violations
      6. incorporate best practices and have test coverage in initial commit
         a. have github action that performs code coverage
         b. protect main branch for code coverage decrease
   i. CRUD operations for custom actions
   j. UI for management of custom actions
   k. Exploitation

Slack stand-up template

[VIVO Dynamic API Standup]
Finished yesterday:
[ticket titles and associated GitHub links]
(AND please include brief textual description)
Working on today:
[ticket titles and associated GitHub links]
(AND please include brief textual description)
Blockers:
[brief textual description]

Notes
Discussion about approach for implementation of authorization. Conclusion was that we should create a separate ontology for Vitro AuthZ solution. One, or
two person should work on designing it before we implement that into Dynamic API. Georgy thinks this is a big issue deserving a complete sprint, and we
need a quick and simple solution for forthcoming sprint. That solution might be that users with role Administrator can modify (create, update, delete)
dynamic actions, while a role for execution of a dynamic action is defined by using hasRole object property in the Action class.

Angular and Angular Material will be used for development of UI. William thinks generators are not solution for our case, it should be dynamic UI capable
to adopt on any change in a dynamic action on the fly. Angular Universal will be used for implementation of Server Side Rendering for the improvement of
Performance and Search Engine Optimization. NGRX will be used as a central state management. A runtime i18n is needed.