



A Proposal for a group of accelerated development templates
for building apps upon the Byteball Foundation's core offering
brought to you by the people behind the Quasar Framework.

Revision: Sept. 11 2018

ATTENTION: If you are reading this right now.

This proposal has been split into two proposals, in line with the discussion that took place over at the Slack channel.

The first proposal document ([see here](#)) is about the creation of the “Pure” starter.

The second proposal document ([see here](#)) is for the extension of the “Pure” starter for specialist applications.

What and Why

Elevator pitch: “Love at FirstByte”

Mere inspiration make not the app alone - as they with great ideas may not necessarily great developers be. Especially within the domain of applied cryptography and virtual currency, novice's mistakes are more likely than beginner's luck. **FirstByte** minimizes these risks by enabling those with moderate skill (or a willingness to learn) to produce shippable best-practice apps for all major computing platforms from one code base and in record time. Tailored to meet the needs of both Byteball enthusiasts and professional developers, the core of **FirstByte** is built on the Quasar Framework and byteball.js - and is distributed under the permissive MIT license.

How

Deliverables

Inheritance is a powerful pattern, and this project will leverage it. **FirstByte** in and of itself is a lerna-based monorepo with distinct but similar kit “flavours” (or extensions if you prefer a technical term) hosted at a public git repository where all of the development and CI pipelines are managed. When it is viable, it will be available as a WIP via NPM and YARN as well as from IPFS and DAT.

Flavours

Each flavour of **FirstByte** is tailored for special needs, and the template (from which the flavours were constructed) will be provided so that the concoction of new flavours can be made with a simple pull request. This proposal includes a time budget for three flavours: “Pure”, “Caramel Sweet” and a final one that will be decided by the **Byteball** community’s Popular Vote.

1. Pure - just the basics, for master chefs and those who aspire to greatness
2. Caramel Sweet - the simplest of the bunch, hard to adapt, easy to use
3. Black Salt - with a special focus on security, anonymity and NAT punching
4. Bitterly Complex - rewarding for the connoisseur of files
5. Sour and Severe - so the deals don’t have to be - a contract focussed flavour
6. Umami - generative and distinguished - only for professionals

The **Pure** flavour (and all other flavours, because they are based on it) will be a minimal setup that includes:

- quasar-cli for best in class development
- a collection of maintenance scripts for development, publication and distribution
- multiple test-runners, including Jest, Cypress and Webdriver-based E2E with sample tests
- ESLint with A11Y “lifting”
- All ES6 language features available
- Gitlab pipelines and dockerfiles
- the innate ability to build SPA, PWA, SSR, Cordova and Electron apps
- GraphQL / Apollo / Prisma API and DB
- the Byteball.js library
- integration with now.js for immediate project delivery
- automated documentation of APIs and functions
- robust system configuration and secret management using ENV variables
- developer configurable colors, icons, background images, graphs and animations
- i18n language translation engine

The **Caramel Sweet** flavour makes a number of decisions for the developer regarding authentication, authorization, web storage, internationalisation, routing and uses a dead-simple provisioning approach with the free tier of Zeit now.sh - so that inside a matter of minutes an app can be launched.

For security professionals, **Black Salt** is not intended for use as a web-application, but rather within the trusted domain of the operating system. With ngrok integration, 2FA and token authentication, in-memory decryption of local data, this flavour is a great starting point for privacy and security-first applications.

The **Bitterly Complex** flavour treats files as first-class citizens and allows for several methods of interaction with IPFS, the local filesystem and specialist concerns like uploading, transcoding and even base64.

The **Sour and Severe** flavour is focussed on writing, verifying and using smart-contracts, perhaps for those interested in building trading or exchange platforms.

The **Umami** flavour is for researchers: Generating, creating and visualizing datasets harvested from the DAG (or even master-nodes!), cross-referenced with other sources and creating highly reactive force-graphs, trees and other delta-based sets of information. If the

developer is interested in making predictions about performance, this would be the place to start.

Documentation

In addition to that which is provided by Byteball, Quasar, vue.js and byteball.js, the **FirstByte** pure starter will describe in detail every component, plugin and script. The factors that make each flavour distinct will be described in minutia. Partially generated from JSDoc comments, and partially handwritten by the author, the approach to be maintained is a “living document”, most likely to be built with Storybook.js. Further, the entire documentation will be i18n based and translated using the Utopian.io / Davinci service.

Code Reviews

There will weekly code-reviews via utopian.io that serve the purpose of verifying code-quality, potentially with specific reviewers familiar with the stack. Since the reviewing process is vital to verify that the work has been done and done correctly, a STEEM bonus will be paid to the accepted review as attached to the TASK. This bonus will come from the global budget of this project. Furthermore, three domain-expert code reviews are envisioned, one from each of the major code stakeholders in the project. First is the code review of the lead developer behind the Quasar Framework, who will give early feedback to the architectural approach behind the Pure flavour and the template. Second, the developers behind byteball.js will guarantee an optimal integration of their library. Finally, the lead dev behind Byteball itself will have the opportunity to review the entire implementation.

When

A three month process is envisioned, with integrated documentation and staged deliverables. This means that a small group of invited early adopters and reviewers can use the project as it stands, beginning at the end of October, with the caveat that they are contributing bug reports and are comfortable with pre-release software. The Sections marked with a * are optional, should the Board decide not to offer predefined specialist flavours. The schedule is as follows:

Timeline

Sept 15 (Project kickoff) => 3000 EUR (retainer payment)

- contractual agreement

Sept 16 - Oct 13 (Preparation) => 2000 EUR

- kickoff meeting
- marketing and outreach
- design guide including logo, claim etc.
- in-depth definition of “extra flavours”
- publicity warm-up on Steem and other relevant channels
- generation of pending utopian tasks
- transfer of IP including logo, wordmark and claim to Byteball Foundation

Oct 14 - Oct 21 (Working Phase Begin) => 3000 EUR

- documentation framework instantiation
- creation issue board
- creation of monorepo, including relevant git, NPM org and NPM repo names
- manufacture of “pure” flavour as base repo

Oct 22 - Oct 29 (Integration Phase) => 3000 EUR

- integration of critical libraries
 - byteball.js
 - .env parser
 - GraphQL & Prisma
 - feathers.js (auth)
 - graphing (d3.js / highcharts.js)
 - IPFS & DAT
 - Jest, Cypress, webdriver based e2e for cordova and electron
 - i18n & a11y

Oct 30 - Nov 12 (Template and Expert review) => 9000 EUR

- creation of template for all starter flavours
- code review by Razvan Stoenescu
- code review by Fabien Marino
- code review by Tony Churyumof

* Nov 13 - Nov 20 (Flavourings and Seasonings) => 3000 EUR

- manufacture of 3 flavours in accordance with community vote

* Nov 21 - Nov 28 (Beta) => 3000 EUR

- generation of complete suite of simple example apps using relevant flavours

Nov 29 - Dec 5 (Revisioning) => 3000 EUR

- revision and test snapshot creation

Dec 6 - Dec 13 (Launch) => 1000 EUR (finishing payment)

- final documentation and delivery
- **public launch with a hackathon sponsored by Quasar and Byteball**

Payment

The fees here are listed in EUR, but it is understood that they will be transferred in MB. With the exception of the retainer, these fees are payable at the documented and verifiable conclusion of the bullet points. The sum total is 30000 EUR, and in order to facilitate some degree of transparency, here is some insight into the way the funds will be used.

- Expert Reviews: A total of 4000€ will be paid to the expert reviewers
- Weekly Code Reviews: Via Utopian + 7 Bounties @ 200€ each = 1400€

The remainder of the fee will be used to cover the following costs and split between the development team:

- Coding ca 45%
- Testing: ca 8%
- Documentation: ca 7%
- Overhead: ca. 5%
- Taxes: ca. 35%

Note: This budget revision is ca. 9000€ less than the original proposal, and this is because we feel that helping the community and providing a good basis for them to work is actually in everyone's best interest.

The Future

The primary developer of **FirstByte** will be a passive maintainer for a period of six months after the delivery, or until a new maintainer steps forward. This means that critical bugs will be patched, modules will be kept up to date and PRs merged. No new code will be written, no features added.

Furthermore, a member of the project group will publish weekly updates on Steem with a project account named @FirstByte, using the 50/50 strategy. Accumulated Steem power from liquid rewards will never be sold, instead used as curation for upvoting projects using the **FirstByte** libraries or postings made by appropriate accounts of project stakeholders. STEEM generated from these postings will be used to reward utopian.io tasks focusing on bug hunting and security testing.

Who

Razvan Stoenescu is the founder and lead developer of the Quasar Framework. His role in the **FirstByte** project is primarily that of advisor and code reviewer, as the main focus of his work as a developer is maintaining the Quasar Framework.

<https://github.com/rstoenescu>

Daniel Thompson-Yvetot is the primary developer that will work on **FirstByte**. He is a team member of the Quasar Framework, where he is responsible for the @quasar/testing module, the Art Department and Public Relations.

<https://github.com/nothingismagick>

Kevin Marrec is a member of the staff of the Quasar Framework. He will contribute to the template construction and documentation.

<https://github.com/kevinmarrec>

Utopian contributors are third-party developers who publicly review the weekly check-ins. They receive STEEM payment from Utopian for completing tasks that will also have additional STEEM bounties to be paid from the project budget.

Slack Discussion Notes:

@neversaynever brought up the point that byteball.js “cannot perform validation of hashtree so it cannot check an address balance without having to trust a hub - therefore it looks like a lot of effort for something that doesn’t allow to build an app that cannot receive payments... cause you got no cryptographic proof that the received funds are real”

This issue would need investigation.

Several people commented that the “flavours” needed better description. This has been undertaken.