Iroha is a distributed ledger project, that aims to provide a development environment where C++ and mobile application developers could contribute to Hyperledger. The project seeks to complement Fabric, Sawtooth, and other potential projects, being a framework with pre-defined set of commands, permissions and queries that can be used with various client libraries to easily create applications for desktop and mobile platforms.

Iroha is inspired by Japanese Kaizen principle — eliminate excessiveness (muri), Iroha has essential functionality for asset, information or identity management, at the same time being an efficient and trustworthy byzantine fault-tolerant tool for your enterprise needs.

Key Characteristics

Permissioned network; written in C++; Client libraries in Java, Python, JS, Swift; BFT consensus algorithm YetAnotherConsensus (YAC); ready-to-use set of commands and queries, Multi-signature transactions.

Documentation

- README
- Wiki
- Documentation

Project Management

Proposals can be created in Jira or suggested and discussed in chat. Those who can implement the proposals then write the code. If people want to know what to work on, they can ask in a special contributors chat or create an issue to get attention. Github projects are used to manage release versions.

Repositories

- iroha (core project)
- iroha-ios (iOS SDK)
- iroha-android (Android SDK)
- iroha-javascript (JavaScript SDK)
- iroha-python (Python SDK)
- https://github.com/hyperledger/iroha-ed25519
- https://github.com/hyperledger/iroha-scala
- https://github.com/hyperledger/iroha-dotnet
- https://github.com/hyperledger/iroha-java
Archived

- iroha-ametsuchi (flatbuffer database)

Communication

Mailing List

- iroha

Chat (for questions and ephemeral discussions)

- #iroha
- Gitter
- Telegram

Meeting

- Discussions are conducted in chat, to allow for worldwide participation. We also create meetings to discuss some questions. Please subscribe to the mailing list or join the chat to keep informed!

Related Pages

- iroha.tech

History

- Proposed by Makoto Takemiya (Soramitsu), Toshiya Cho (Hitachi), Takahiro Inaba (NTT Data), and Mark Smargon (Colu)
- Approved by the TSC on October 13, 2016
- Moved out of Incubation on May 18, 2017

Good first issues

<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>T</th>
<th>Assignee</th>
<th>P</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>IR-573</td>
<td>Refactor pending txs storage tests</td>
<td>✓</td>
<td>Unassigned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IR-438</td>
<td>(Re)Move TransactionSequence classes</td>
<td>✓</td>
<td>Unassigned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IR-403</td>
<td>Separate BatchHashEquality and MstState</td>
<td>✓</td>
<td>Joseph Nicholas Alcantara</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IR-402</td>
<td>Remove unwanted public key format validation from unit tests</td>
<td>✓</td>
<td>Unassigned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IR-358</td>
<td>Add information about used ports in Deployment documentation</td>
<td>✓</td>
<td>Unassigned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IR-324</td>
<td>Rework YAC-related unit tests to use supermajority checker mock</td>
<td>✓</td>
<td>Unassigned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IR-280</td>
<td>Fix UB sanitizer issues (Segfaults) and identify false positives</td>
<td>✓</td>
<td>Unassigned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IR-279</td>
<td>Fix data race sanitizer issues (thread sanitizer) and identify false positives</td>
<td>✓</td>
<td>Unassigned</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Examples of additions for this page - based on Sara Garifullina request to enliven the page:

- Task lists can be assigned user etc.
- This is already in use for pages for validating an action is complete

**Example Poll**

@KellyCooper for poll request additions

Yes, this location works for a poll

No, arrange for a poll to be added to the template

I think it would be great if we move it to the pool example page and then add to decision pages

Also, charts, galleries, roadmap planner, etc. If you have data or content I can mock up.

**Recent space activity**

- **Sara Garifullina**
  - Suggestions and ideas for HL Iroha 1.x and 2.0! created Jan 24, 2020

- **Mikhail Boldyrev**
  - Custom ordering of query results created Jan 03, 2020
  - Iroha and OSS-fuzz updated Oct 17, 2019 • view change

- **Konstantin Mumichev**

**Space contributors**

- Sara Garifullina (52 days ago)
- Mikhail Boldyrev (72 days ago)
- Konstantin Mumichev (150 days ago)
- Kelly Cooper (210 days ago)
- John Siegrist (241 days ago)
- ...

---

<table>
<thead>
<tr>
<th>Issue</th>
<th>Description</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>IR-278</td>
<td>Fix AddressLeak sanitizer issues and identify false positives</td>
<td>Unassigned</td>
</tr>
<tr>
<td>IR-257</td>
<td>Make a MockBlockStore possible in Storage</td>
<td>Unassigned</td>
</tr>
<tr>
<td>IR-240</td>
<td>Create python examples: can_add/subtract_domain_asset_qty</td>
<td>Unassigned</td>
</tr>
<tr>
<td>IR-197</td>
<td>Unnecessary Protobuf backend usage in irohad</td>
<td>Unassigned</td>
</tr>
<tr>
<td>IR-122</td>
<td>Leave default value if some of env value IROHA_POSTGRES_* do not exist</td>
<td>Unassigned</td>
</tr>
<tr>
<td>IR-81</td>
<td>Wrong response on getAccountAssetTransactions query (a non-existing account or asset)</td>
<td>Unassigned</td>
</tr>
<tr>
<td>IR-80</td>
<td>Force peer check that it is launched with the same key as it has in genesis block</td>
<td>Unassigned</td>
</tr>
<tr>
<td>IR-78</td>
<td>Genesis block should pass stateless validation</td>
<td>Unassigned</td>
</tr>
</tbody>
</table>