

Code Issues 13 Pull requests 1 Discussions Actions Projects Wiki Security Insights

main 7 branches 18 tags

efirs fix: wrap error in the error object in response		
.github	fix: cache docker builds	
api	fix: wrap error in the error object in response	
cdc	fix: CDC events were not being processed	
cmd/consistency	refactor: query executor	
config	Kubernetes manifests	
deployment/manifests	fix: remove db from manifests	
docker	fix: cache docker builds	18 days ago
docs/assets	Feat: Update the README to point to the docs website (#19)	7 days ago
internal	fix: wrap error in the error object in response	19 hours ago
keys	fix: allow tests to override system table names	23 days ago
lib	feat: auto-generate primary keys if not set by the user	7 days ago
query	fix: wrap error in the error object in response	19 hours ago
schema	fix: wrap error in the error object in response	19 hours ago
scripts	fix: wrap error in the error object in response	19 hours ago
server	fix: wrap error in the error object in response	19 hours ago
store	feat: improved stream event with decoded op fields	16 days ago
test	fix: wrap error in the error object in response	19 hours ago
util	fix: Rollback failed transaction	21 days ago
value	fix: wrap error in the error object in response	19 hours ago
.gitignore	Allow DELETE method with body (#19)	4 months ago
.gitmodules	fix: use the renamed api repo	last month
.releaserc.yaml	feat: release management	2 months ago
CODE_OF_CONDUCT.md	Adding code of conduct	2 months ago
LICENSE	Initial commit	4 months ago
Makefile	feat: Added getInfo handler to return server metadata	last month
README.md	Feat: Update the README to point to the docs website (#19)	7 days ago
go.mod	fix: upgrade dependencies	5 days ago
go.sum	fix: upgrade dependencies	5 days ago

Clone

HTTPS GitHub CLI

<https://github.com/tigrisdata/tigris.git>

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Download ZIP

About

Tigris is a modern, scalable backend for building real-time websites and apps.

[www.tigrisdata.com/](http://www.tigrisdata.com/)

go, kubernetes, golang, data, streaming, opensource, real-time, database, consensus, transactional-database

Readme, Apache-2.0 license, Code of conduct, 36 stars, 9 watching, 2 forks

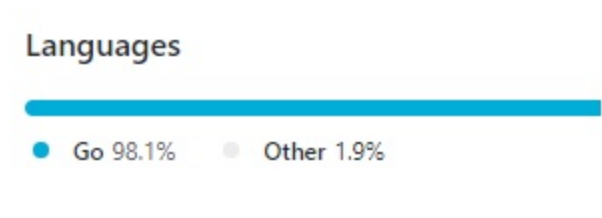
Releases

18 tags

Packages

No packages published

Contributors 6



### README.md

For developers who want to build scalable web and mobile apps fast!

With a zero-ops scalable pub/sub messaging system and fault-tolerant cloud-native database, Tigris provides everything you need to get up and running quickly and efficiently.

go report: A+ Go passing Contributor Covenant 2.1

[Website](#) | [Quickstart](#) | [API Reference](#) | [Slack Community](#) | [Twitter](#)

## Helpful Links

- Key Concepts
- Quickstarts
- Data Modeling
- Guides
- Client Library: Go
- Client Library: Java
- Command Line Interface

## Community & Support

- Slack Community
- GitHub Issues
- GitHub Discussions

## Developing

### Building with Docker Containers

Start local Tigris server listening on `http://localhost:8081` by running:

```
make run
```

This would bring dependencies and server up in the docker containers with all your changes.

### Running tests in docker containers

Tests are executed using `make test`. This runs both unit and integration tests. The FoundationDB container auto-generates the cluster file which is shared between containers using a docker volume.

The docker volume `fdbdata` is mounted at:

- `/var/fdb/fdb.cluster` in `tigris_fdb` container
- `/etc/foundationdb/fdb.cluster` in `tigris_test` and `tigris_server` containers, which is the default location where the client will search for the cluster file

### Building and Testing on Apple M1

Due to various issues with Docker on Apple M1, it is recommended to install FoundationDB on the host and run the server on the host directly as well.

To run the server on the host, required dependencies need to be installed by running:

```
sh scripts/install_build_deps.sh
sh scripts/install_test_deps.sh
```

Once the dependencies are installed, you can start up the server on the host as follows:

```
make osx_run
```

You can run the test on your local machine as follows:

```
make osx_test
```

## License

This software is licensed under the [Apache 2.0](#).