Family Thread Installation Guide

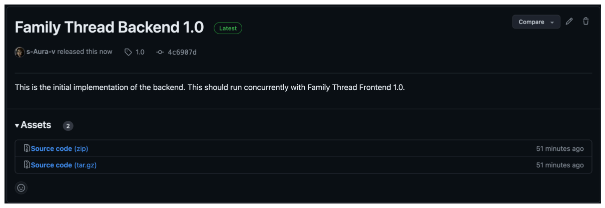
Pre-Reqs:

Docker: https://docs.docker.com/get-docker/
Node: https://nodejs.org/en/download

Download Guide:

Github:

Download and Extract the Family Thread Files
Frontend: https://github.com/FamilyThread/Family_Thread/releases/tag/1.0
Backend: https://github.com/FamilyThread/Family_Thread_backend/releases/tag/1.0



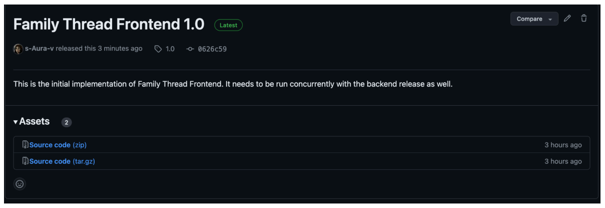
Family Thread Backend 1.0 Latest

s-Aura-v released this now 1.0 4c6907d

This is the initial implementation of the backend. This should run concurrently with Family Thread Frontend 1.0.

Assets 2

Source code (zip) 51 minutes ago
Source code (tar.gz) 51 minutes ago



Family Thread Frontend 1.0 Latest

s-Aura-v released this 3 minutes ago 1.0 0626c59

This is the initial implementation of Family Thread Frontend. It needs to be run concurrently with the backend release as well.

Assets 2

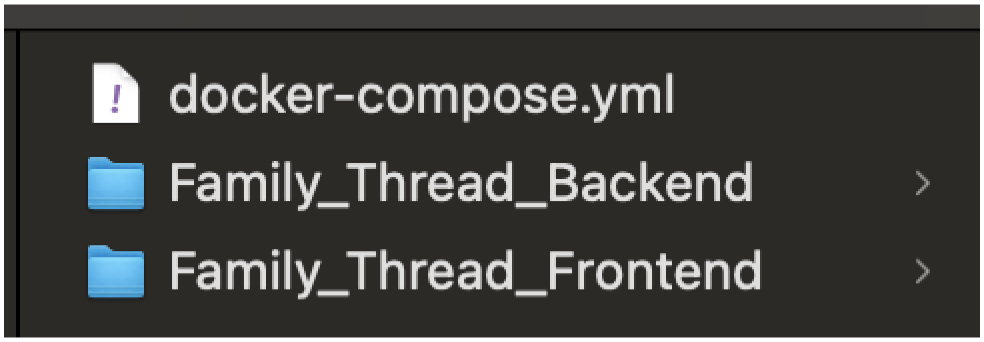
Source code (zip) 3 hours ago
Source code (tar.gz) 3 hours ago

Run Guide:

Quickstart

This is the recommended setting if you do not plan on making any changes. It dockerizes all of the applications at once.

1. In the directory that contains the frontend and backend code, create a docker-compose.yml file containing the following code.



2. Run

   docker-compose up –build -d

```
version: '3.8'

services:
 backend:
   build:
     context: ./Family_Thread_Backend
     dockerfile: Dockerfile
   container_name: 380sp24-ft-backend
   depends_on:
     - mongo
   ports:
     - "27021:27021"

 frontend:
   build:
     context: ./Family_Thread_Frontend
     dockerfile: Dockerfile
   container_name: frontend-react-app
   ports:
     - "5173:5173"
   depends_on:
     - backend

 mongo:
   image: mongo:4.0.4
   command: mongod --port 27020
   container_name: teamb-380-sp24
   ports:
     - "27020:27020"
   volumes:
     - mongodata:/data/db

volumes:
 mongodata:
   driver: local
```

Manual Methods:

Method 1: Docker

Docker containerizes all of the program and lets you run the program with ease. This deployment works with all devices as long as Docker is installed. This is the recommended method.

Setup:

1. Ensure that ports 5173, 27020, and 27021 are free
   a. Can be configured
2. Ensure that you have Docker downloaded

Run:

1. Enter the project directory on terminal
2. Change directory to backend
   a. Run: docker-compose up -d
      Alt: docker compose up -d
      Some devices may not require a hyphen between docker-compose.
3. Change directory to front-end
   a. Run the following commands:
      docker build -t [IMAGE NAME] .
      docker run -d --rm -p 5173:5173 --name [CONTAINER NAME] [IMAGE NAME]

Ending the program:

1. Open the file directory containing the project files in the terminal

2. Change directory to back-end
    a. Run: docker-compose down
       Alt: docker compose down
       Some devices may not require a hyphen between docker-compose.
3. Change directory to front-end
    a. Stop:
       docker container stop [CONTAINER NAME]
       docker image rm [IMAGE NAME]

Method 2: Self Start

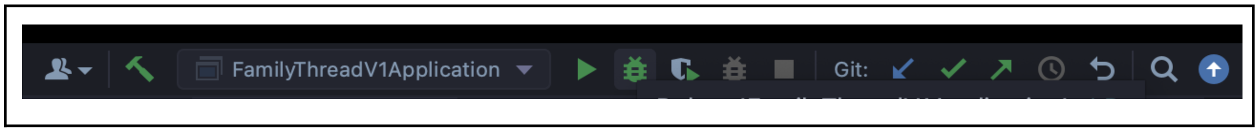
You can also start the program using your preferred IDE. The following guide was done using WebStorm and IntelliJ. This is riskier as it requires that you have all of the proper files installed beforehand.

Setup:

1. Have an IDE of your choice installed

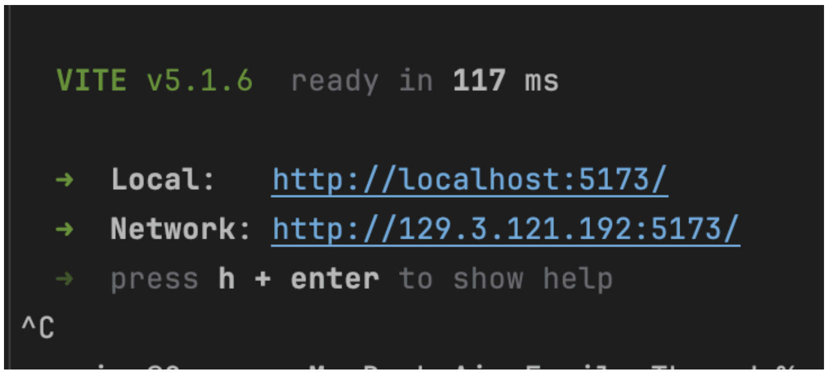
Run:

1. Open the front-end code using WebStorm
2. On the IDE terminal, enter the following command:
   npm i
   npm run dev
3. Open back-end code using IntelliJ
    a. Enter src -> main -> java -> utils
       Run the FamilyThreadV1Application



Ending the program:

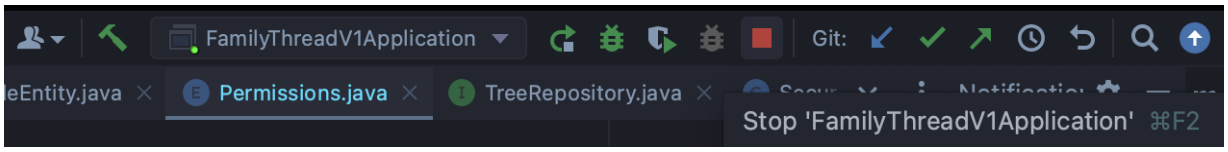
1. On the front-end terminal,
   Alt: press CTRL + C



```
VITE v5.1.6  ready in 117 ms

➜  Local:   http://localhost:5173/
➜  Network: http://129.3.121.192:5173/
➜  press h + enter to show help
^C
```

2. On the back-end program,
   stop the Java program.


FamilyThreadV1Application
Git:
leEntity.java
Permissions.java
TreeRepository.java
Stop 'FamilyThreadV1Application' ⌘F2

Configurations:

Default Ports:

React-App	5173
MongoDB	27020
Google oAuth / Springboot Server Port	27021

Customize Deployment:

1. Go to backend-code
   a. Enter src -> main -> resources
   b. Enter application.properties
      i. Adjust the following code to change email info

```
spring.mail.host=smtp.gmail.com
spring.mail.port=587
spring.mail.username= [GMAIL ACCOUNT]
spring.mail.password= [PASSWORD]
spring.mail.properties.mail.smtp.auth=true
spring.mail.properties.mail.smtp.starttls.enable=true
```

      ii. Adjust the following code to change Database

```
spring.data.mongodb.database = [MONGO DATABASE NAME]
spring.data.mongodb.uri= [MONGO DATABASE URL]
                    Example: mongodb://[SERVER]:[PORT]/
```

      iii. Adjust the following code to change OAuth settings

```
spring.security.oauth2.client.registration.google.client-id= [ClIEND ID]
spring.security.oauth2.client.registration.google.client-secret= [SECRET]
```

      iv. Adjust the following code to change connection to frontend

```
frontend.url = [SERVER-URL]
server.port=  [SERVER-PORT]
```

2. Change Docker Settings
   a. Enter docker-compose.yml
      i. app: container-name
      ii. Adjust the following code

```
services:
 app:
   build:
     context: .
     dockerfile: Dockerfile
   container_name: [CONTAINER NAME]
   depends_on:
     - mongo  # Ensure the MongoDB service is started
first
   ports:
     - [PORT FOR SPRINGAPP] // Default: 27021
 mongo:
   image: [MONGO VERSION] //Default = mongo:4.0.4
   command: mongod --port [PORT]
   container_name: [CONTAINER NAME FOR MONGO]
   ports:
     - [PORT FOR MONGO] // Default: 27020
   volumes:
     - mongodata:/data/db
volumes:
 mongodata:
   driver: local
```

b. Change compiled app-name
   i. Enter pom.xml
   ii. Go to the 3rd line before the end of the file

```
<finalName>[COMPILED FILE NAME]</finalName>
```

   iii. Enter Dockerfile in root directory

```
ENTRYPOINT ["java", "-jar", "target/[COMPILED FILE
NAME"]
```

Part 2: Frontend Changes

1. Go to front-end code
2. Enter Dockerfile in root
   a. EXPOSE [FRONTEND PORT-NUM]
3. Enter src->config
   a. Open constant.ts
   b. Change the "backend_url" to the desired Springboot port