Question on how to use this framework #303

New issue

Open **datnguyen293** opened this issue on Jun 16, 2018 · 12 comments

datnguyen293 commented on Jun 16, 2018

Firstly I love the idea of the framework and started writing some services. I'm now going to use this framework for a bigger project which has more than 10 services, so I have couple questions:

1. Should I maintain only one project source based and write all services under the "/services/" folder - or each service should have its own project structure which generated by moleculer-cli?
2. REST requests to almost services need to be checked for authorization so I'm thinking of writing an auth library which required by all services. The library provide the most important function "isAuthorized()" function. Where should I put the library? Should it be a github package and included in packages.json?
3. I don't have much experience with NATS server so do you have any suggestions on how to maintain this service stable? Can it be multiple instances behind a load balancer?

Thank you very much,
Dat

icebob commented on Jun 16, 2018 Member

Hi Dat,

1. It's your choice. But I think 10 services are not too much to handle in one project under `services` folder. Btw you can use subfolders too to organize them. In my previous projects, I've used ~40 microservices in one project in one repo. The development is easy and when you deploy it to production you can control with `SERVICES` env var what services want to load. So storing all services in a project doesn't mean that you should deploy all of them together.
2. You can put the authorization to `authorize` method of API Gateway. In this method, you can call other services and it's called for every REST request. So it's a good point to handle auth stuff.
3. NATS supports clustering. And NATS clients supports too, so you don't need any extra coding. Just check NATS.io site, that how you can setup a NATS cluster and setup it in the clients.

Cheers!

👍 1

icebob added the Type: Question label on Jun 16, 2018

FranzZemen commented on Jun 16, 2018 • edited

![capture](https://user-images.githubusercontent.com/6136034/41508746-07ed61ac-720f-11e8-92ce-2d64ccd6fff9.PNG)
Great questions I think we all go through. My thoughts:
1. Both on the backend and front end I've struggled with scaffolding questions. My conclusion after many projects is that while some folder organization is useful, a folder-per-service is useless for small projects, amazing for medium project, and painful for large projects. For large projects, better to have lighter, broader folders for functional areas rather than microservices and use good naming conventions, e.g. someService.microservice.ts for the service function, someService.moleculer.ts for the configuration, someService.spec.ts for the tests. For me a small microservice platform is probably 10-50 microservices, medium is 50 to hundreds, and large is over hundreds and up. Microservices should be small and implement no less or no more than the granular story (message) they are there to satisfy, so that usually means lots.

2. I'd recommend thinking about it as having a REST API gateway whose endpoints perform authentication and authorization, then delegate to reuseable microservices. I've attached a conceptual diagram I'm using in my journey in my team. If you have fine grained authorization before every microservice make sure its actually useful.

3. No opinion on NATS except to make sure that it easily fits within your OS brand/version.
…

👍 5

icebob commented on Jun 16, 2018 Member

@**FranzZemen** the attachment is missing.

datnguyen293 commented on Jun 16, 2018 Author

Thank you all for your good comments and I more understand what need to do. A comment to my question is when I mean 10 services which means 10 different components and each of it has many services/api in there.
I would love to go with @**icebob** suggestion that keep only one project and we can load different service when deploying.

Thanks,
Dat

ghqppio commented on Jun 20, 2018

I believe, managing 10 services under one project is not good practice in terms of scaling.
Let's say I have an image processing service which I'd like to scale? How would you suggest to scale it?
As I see it, if we're taking the docker/kubernetes approach - it's better to divide the services to different projects/containers.
Yes, It's a bit harder to manage. But in terms of scaling specific services, I believe it's reasonable on the long run.

What do you think, guys?

datnguyen293 commented on Jun 20, 2018 Author

@**ghqppio** as far as I understand, @**icebob** suggested to manage all services under one project but in deployment you can start only one/or fews services under the project. That's no conflicts with using docker/kubernetes at all.
Taking further, I think we can using sub-projects for each service under the "/services/" folder would not be a bad choice.

icebob commented on Jun 20, 2018 Member

Yes, that you put all services in a project, you can still run them in separate containers. Example:
https://github.com/moleculerjs/moleculer-examples/blob/master/conduit/docker-compose.yml

Wallacy commented on Jun 20, 2018 Contributor

Hello,

I intended to make a similar thread, I'm glad it already exists.

We are validating the moleculer for use, and one of the doubts that came up was:

In production, it is better to use the moleculer-runner and point out which services are going to be started (in cluster mode for example). Or use a broker.start somewhere with the services you want.

Can I protect others services running moleculer-runner if one service crash?

Any tips to create a restart behavior like PM2?

👍 1

katsanva commented on Jun 22, 2018 Contributor

@**ghqppio** well, you probably can build different docker images from the common codebase for each separate service. but yeah, it's not a really good solution though.

in my own case, it was much easier and better to create 10 repositories with about 10 lines of copy-pasted code each, than create 1 repo, but handle that intelligent build process but have no copy pasted code

soluty commented on Sep 25, 2018

my choice is to use ci tools to shink source code for one service.
i write a service_assemble.js file to do this , and when i push code, ci tool do change code ,remove unuse service ,so i can build it an deploy to a docker image

nicklasfrahm commented on Feb 16, 2019

@**soluty** The script sounds interesting. Would you mind sharing it via gist or similar? I think it would have the potential to become part of the `moleculer-cli`. Something like `moleculer pack` or `moleculer build`, which could potentially create a *virtual* `package.json` and load only required files and dependencies of a single service into a directory that could then be used to create a pure Docker image. This would be similar to the approach that @**katsanva** described.

ccampanale commented on Feb 16, 2019

@**katsanva** My team does the same thing; we have several repositories and we have a template repository (fork of the official template customized for our needs and on our enterprise GitHub) which we used to generate new services with all of our healthcheck and CI scripts, common Dockerfiles, etc.

Technically you could argue that splitting them into separate repos is most in line with the entire idea of microservices in the first place: separation of concerns, data boundaries, etc. Furthermore, in many situations - not my teams, at least not yet - you have a different team managing one or more microservices each. I feel like this sort of "encapsulation" extended to the repository level makes a lot of sense.

All of that being said, one of my favorite things about **Moleculer** is that you have options.

👍 2

Write Preview

Leave a comment

New **Video support! Upload MP4 and MOV file types.** Attach files by dragging & dropping, selecting or pasting them.

Comment

Remember, contributions to this repository should follow its contributing guidelines and code of conduct.

Assignees
No one assigned

Labels
Type: Question

Projects
None yet

Milestone
No milestone

Linked pull requests
Successfully merging a pull request may close this issue.
None yet

Notifications Customize
Subscribe
You're not receiving notifications from this thread.

9 participants