The Art of Building Node.js Projects at scale

Raymond Feng
STSM/Architect@IBM, Co-creator of LoopBack

Node+JS Interactive 2018
About me

• STSM@IBM API Connect - Architect of LoopBack (a popular open source Node.js API platform)

• Co-founder of StrongLoop (A Node.js startup acquired by IBM in 2015) - One of the creators of LoopBack

• A passionate Node.js developer - transitioned from an Enterprise Java developer

GitHub: raymondfeng Twitter: cyberfeng LinkedIn: raymondfeng
Let’s run a few polls

- Do you work on Node.js applications, frameworks, or modules?
- Do you think your Node.js project is large scale?
- What are the characteristics of a large scale Node.js project?
A large scale Node.js project

- # of Teams, Developers, Users
- # of NPM Packages, Github Repos
- # of Years & Releases of Development and Maintenance
LoopBack - a large scale Node.js framework

• 11,000+ Github Stars
• 50+ NPM modules
• 150,000+ monthly downloads
• 8 Full-time maintainers and many community contributors
• Created 5 years ago
• 3 major releases and V4 is out this week
V4 - The next generation of LoopBack

Rewriting the framework from ground-up
Organizing your code base

NPM Packages
Github Repositories
Many packages + repos
(LoopBack 1.x- 3.x)

Large number of Packages (and Repositories)
npm link
CI Complexity
Monorepo manage by Lerna (LoopBack 4.x)

1 Repository, Multiple Packages

1 PR, Multiple Packages

Simplified CI

https://github.com/lerna/lerna
https://github.com/strongloop/loopback-next

lerna cli or npm scripts

lerna bootstrap
lerna version && lerna publish from-git
lerna run …
lerna exec …
lerna ls

lerna.json

```json
{
  "lerna": "3.3.0",
  "packages": [
    "benchmark",
    "docs",
    "examples/*",
    "packages/*",
    "sandbox/"
  ],
  "packages": [
    "@loopback/docs",
    "@loopback/example-hello-world",
    "@loopback/example-log-extension",
    "@loopback/example-rpc-server",
    "@loopback/example-soap-calculator",
    "@loopback/example-todo-list",
    "@loopback/example-todo",
    "@loopback/authentication",
    "@loopback/boot",
    "@loopback/build",
    "@loopback/cli",
    "@loopback/context",
    "@loopback/core",
    "@loopback/dist-util",
    "@loopback/http-caching-proxy",
    "@loopback/http-server",
    "@loopback/metadata",
    "@loopback/openapi-spec-builder",
    "@loopback/openapi-v3-types",
    "@loopback/openapi-v3",
    "@loopback/repository-json-schema",
    "@loopback/repository",
    "@loopback/rest",
    "@loopback/service-proxy",
    "@loopback/testlab",
  ]
}
```
TypeScript Project References

- https://github.com/strongloop/loopback-next/pull/1636

- lerna -> TypeScript project references

```json
{
    "$schema": "http://json.schemastore.org/tsconfig",
    "extends": "../build/config/tsconfig.common.json",
    "compilerOptions": {
        "composite": true
    },
    "references": [
        {"path": "../openapi-spec-builder/tsconfig.json"},
        {"path": "../repository/tsconfig.json"},
        {"path": "../testlab/tsconfig.json"},
        {"path": "../context/tsconfig.json"},
        {"path": "../core/tsconfig.json"},
        {"path": "../http-server/tsconfig.json"},
        {"path": "../openapi-v3/tsconfig.json"},
        {"path": "../openapi-v3-types/tsconfig.json"}
    ]
}
```
Using the right language

Plain JavaScript vs. TypeScript
Plain JavaScript
(LoopBack 1.x-3.x)

Hidden Contracts
Steep Learning Curve
Need Help From Others
TypeScript
(LoopBack 4.x)

Contracts
Self Documenting Code
Compiler / IDE Support
See the difference

```javascript
/**
 * @param {Object} query Filter for search
 * @param {Object} options Options
 */
async findOne(query, options) {
    // code
}

async findOne(query?: Filter, options?: Options): Promise<T | null> {
    // Code
}

/**
 * Query filter object
 */
export interface Filter {
    where?: Where;  // The matching criteria
    fields?: Fields;  // To include/exclude fields
    order?: string[];  // Sorting order for matched entities.
    limit?: number;  // Maximum number of entities
    skip?: number;  // Skip N number of entities
    offset?: number;  // An alias for 'skip'
    include?: Inclusion[];  // To include related objects
}
```
Managing Artifacts and their Dependencies

Registration & Resolution
Lookup vs. Injection
LoopBack 4 core - an IoC container

Universal Registry
Key Based Binding
Dependency Injection
(Async) dependency injection

- controllers.UserController
- repositories.UserRepository
- utilities.PasswordHasher
- UserController
- UserRepository
- PasswordHasher
- NewPasswordHasher
Declare dependencies

- Constructor/Property/Method Parameters
- Async/Sync/Optional/…

```typescript
export class UserController {
  constructor(
    // @inject('repositories.UserRepository')
    @repository(UserRepository) protected repo: UserRepository,
    @inject('utilities.PasswordHasher') protected hasher: PasswordHasher,
  ) {}
}
```
Achieving Extensibility

Extension points & extensions
Extensibility

- Lots of Artifacts
- Unknown Future Extensions
- Cannot Know them ahead of time
An example of extension point/extensions
Connect extensions to its extension point

Contribute Artifacts

Key Based Binding

Packaged As A Component
Chaining Responsibilities

Sequence of Actions
Sequence of Actions

- Middleware style (one way vs. cascading, Express)
- A coordinator, such as bootstrapper, to walk through multiple phases and invoke different methods (@loopback/boot)
- Functional composition (@loopback/rest Sequence)
Improving User Productivity

Conventions & CLI
@loopback/cli

- npm install -g @loopback/cli
- lb4 --help
- lb4 app
- lb4 extension
- lb4 controller
- lb4 datasource
- lb4 model
- lb4 repository
- lb4 service
- lb4 example
- lb4 openapi
Summary

- What characteristics do large scale Node.js projects have in common?
- How to build your Node.js project at scale?
  - Scale the code base with modularity without complexity
  - Scale the developers with language/IDE productivity
  - Scale the runtime with certain patterns
  - Scale the users with conventions and tools
Thank you!

strongloop/loopback-next

v4.loopback.io

@StrongLoop
Acknowledgement

• The LoopBack team at IBM and our contributors from the community - making good things happen in open source

• Taranveer Virk - allowing me to reuse some of his slides from NodeSummit 2018

• Iconography by FontAwesome is licensed under CC BY 4.0