

# THE ART OF COMPOSABILITY AND EXTENSIBILITY:

*Crafting a foundation for frameworks and applications in Node.js with TypeScript*

QUESTION

Do you work on a large scale Node.js  
Application / Framework?

QUESTION

**What are the characteristics of a  
large scale Application /  
Framework?**

## CHARACTERISTICS OF A LARGE SCALE APPLICATION / FRAMEWORK



# of Developers



# of Packages



Maintained for Years

LOOPBACK :: A NODE.JS API CREATION FRAMEWORK



RE-WRITING THE FRAMEWORK FROM THE GROUND UP

SCALE OF LOOPBACK :: IT'S A LARGE FRAMEWORK

11,000+ GitHub Stars

SCALE OF LOOPBACK :: IT'S A LARGE FRAMEWORK

50+ Packages

SCALE OF LOOPBACK :: IT'S A LARGE FRAMEWORK

8 Full-Time Maintainers

+

Community Maintainers

SCALE OF LOOPBACK :: IT'S A LARGE FRAMEWORK

Created 5 Years Ago

SCALE OF LOOPBACK :: IT'S A LARGE FRAMEWORK

# 3 Releases

# Language of Choice

JAVASCRIPT IS GREAT, BUT ...



Hidden Contracts



Steep Learning Curve



Need Help From Others

# TypeScript

## BENEFITS OF OPTIONAL TYPES VIA TYPESCRIPT



Contracts

Self Documenting Code



Compiler / IDE Support

## JAVASCRIPT :: WHAT IS QUERY AND OPTIONS?

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

## WITH TYPESCRIPT

```
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
}
```

# Packages & Repositories

## BALANCING EXTENSIBILITY, MODULARITY WITH PACKAGES



Large number of Packages  
(and Repositories)



npm link



CI Complexity

# Lerna

LERNA MAKES IT EASY



1 Repository,  
Multiple Packages



1 PR,  
Multiple Packages



Simplified CI

## LERNA IN ACTION :: 1 REPOSITORY, 20+ PACKAGES

@loopback/authentication	v0.10.18
@loopback/boot	v0.11.8
@loopback/build	v0.6.11
@loopback/cli	v0.18.1
@loopback/context	v0.11.11
@loopback/core	v0.10.3
@loopback/dist-util	v0.3.3
@loopback/http-caching-proxy	v0.2.7
@loopback/http-server	v0.2.3
@loopback/metadata	v0.8.11
@loopback/openapi-spec-builder	v0.7.11
@loopback/openapi-v3-types	v0.7.11
@loopback/openapi-v3	v0.11.2
@loopback/repository-json-schema	v0.9.15
@loopback/repository	v0.13.2
@loopback/rest	v0.17.1
@loopback/service-proxy	v0.5.14
@loopback/testlab	v0.10.11
@loopback/example-hello-world	v0.8.13
@loopback/example-log-extension	v0.10.13
@loopback/example-rpc-server	v0.9.3
@loopback/example-todo	v0.13.2
@loopback/docs	v0.14.2

# Composing the Building Blocks

@LOOPBACK/CONTEXT



Universal Container



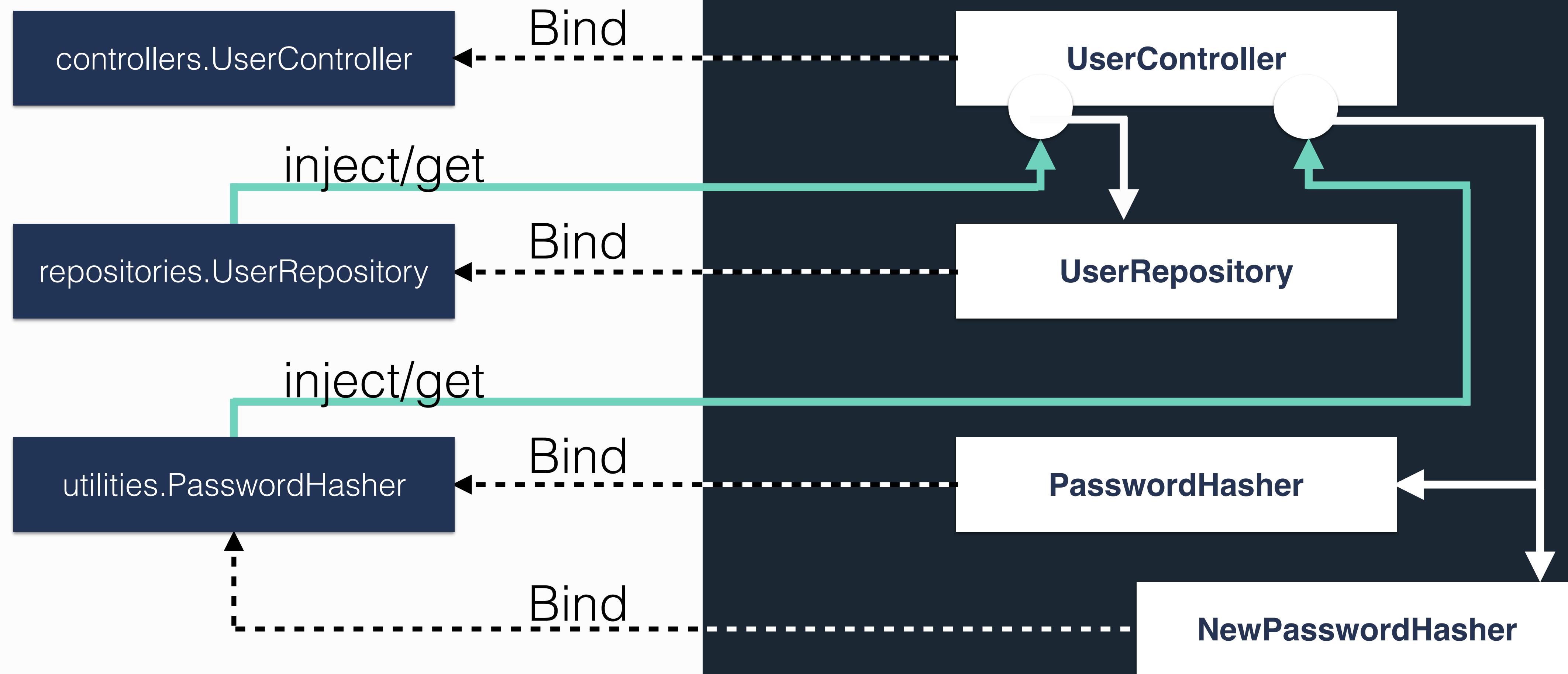
Key Based Binding



Dependency Injection

## CONTEXT

## ARTIFACTS



## DEPENDENCY INJECTION :: CODE EXAMPLE

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

# Extensibility & Extensible Code Patterns

QUESTION

# Why does extensibility matter?

## EXTENSIBILITY CHALLENGES



Lots of Artifacts



Unknown Future Direction



Cannot Know It All

# Extensions

## EXTENSIONS



Contribute Artifacts



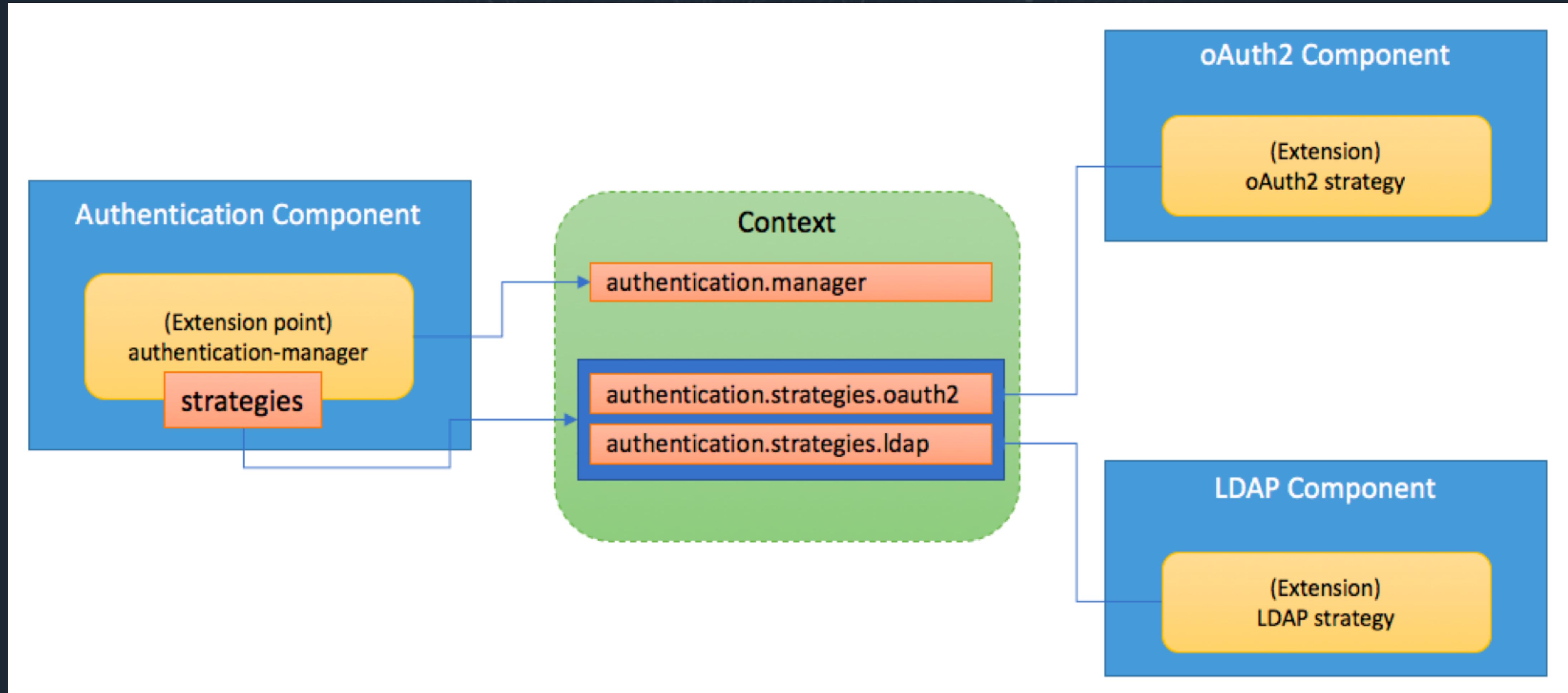
Key Based Binding



Packaged As A  
Component

# Extension Point Pattern

## EXTENSION POINT PATTERN



# Sequence of Actions

## SEQUENCE OF ACTIONS

- ❖ Simple functions composed to create a response
- ❖ Can intercept req / res and modify or build on it

## EXAMPLE

```
async handle(context: RequestContext): Promise<void> {
  try {
    const {request, response} = context;
    const route = this.findRoute(request);
    const args = await this.parseParams(request, route);
    const result = await this.invoke(route, args);

    debug(`%s result -`, route.describe(), result);
    this.send(response, result);
  } catch (error) {
    this.reject(context, error);
  }
}
```

# Convention, Standardization, Flexibility

## @LOOPBACK/CLI

```
[(py27) Taranveers-MBP:~ taranveer$ lb4 --help
Usage:
  lb4 app [<name>] [options]

Options:
  -h,  --help          # Print the generator's options and usage
  --skip-cache        # Do not remember prompt answers
  --skip-install      # Do not automatically install dependencies
  --applicationName  # Application class name
  --description       # Description for the application
  --outdir            # Project root directory for the application
  --tslint             # Enable tslint
  --prettier           # Enable prettier
  --mocha              # Enable mocha
  --loopbackBuild     # Use @loopback/build
  --vscode             # Use preconfigured VSCode settings
  --private            # Mark the project private (excluded from npm publish)
  -c,  --config         # JSON file name or value to configure options
  -y,  --yes             # Skip all confirmation prompts with default or provided value
  --format             # Format generated code using npm run lint:fix

Arguments:
  name   # Project name for the application  Type: String  Required: false

Available commands:
  lb4 app
  lb4 extension
  lb4 controller
  lb4 datasource
  lb4 model
  lb4 example
  lb4 openapi
(py27) Taranveers-MBP:~ taranveer$ █
```

## SUMMARY

- TypeScript
- Lerna
- `@loopback/context`
  - Universal Registry Container
  - Provides Extensibility & Composability
- Extensible Code Patterns
  - Components
  - Extension Point Pattern
  - Sequence of Actions
  - Convention, Standardization, Flexibility
  - CLI

SEE IT IN ACTION



**strongloop/loopback-next**



[v4.loopback.io](https://v4.loopback.io)

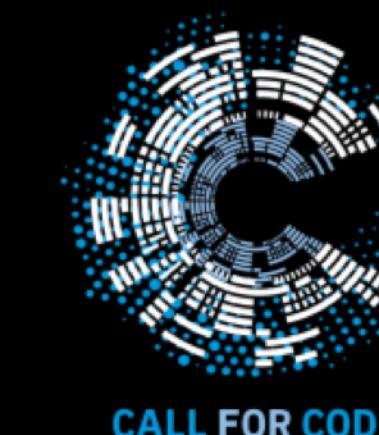


@StrongLoop

NATURAL  
DISASTERS  
ARE AMONG  
THE WORLD'S  
GREATEST  
CHALLENGES

HOW WOULD  
22 MILLION  
DEVELOPERS SOLVE  
THESE GLOBAL  
ISSUES IF GIVEN A  
CHANCE TO ANSWER  
THE CALL?

FIND OUT HOW AT  
[developer.ibm.com/callforcode](http://developer.ibm.com/callforcode)



Call for Code Founding Partner

Iconography by [FontAwesome](#) is  
licensed under [CC BY 4.0](#)

# Thank You



@virkt25